E-ISSN:2540-9719
ISSN:2302-8149

PENERAPAN *WORD* N-GRAM UNTUK *SENTIMENT ANALYSIS REVIEW* MENGGUNAKAN METODE SUPPORT VECTOR MACHINE (STUDI KASUS: APLIKASI SAMBARA)

Fitriyani, Toni Arifin
Sistem Informasi, Teknologi Informasi, Universitas Adhirajasa Reswara Sanjaya,
Jalan Sekolah Internasional No.1-2 Antapani, Bandung.
Email: fitriyani.adja@gmail.com, toni.arifin@ars.ac.id

(Diterima: 15 Agustus 2020, direvisi: 21 Agustus 2020, disetujui: 10 September 2020)

ABSTRACT

Sambara application is an innovation from Bapenda West Java for motor vehicle tax services. The Sambara application expected can be provide efficiency, effectiveness and service improvement. The success of the application can be determined by conducting a sentiment review analysis. Sentiment analysis aims to detect polarity in the text in the form of negative or positive opinions, using text mining. At the text processing stage, the Word N-Gram feature is added as a word identification approach and for classification it uses the Support Vector Machine (SVM) method. This study aims to determine the application of Word N-Gram, the results of the accuracy value using the SVM method, and find out how much influence the application of Word N-Gram on the accuracy value. The highest accuracy value in this research was 89.00% with AUC value of 0.944 (excellent classification) on the amount of data 900, but when uses Bi-gram and Tri-gram results in a decrease in accuracy. The accuracy value with the highest increase is in the application of tri-grams with the amount of 1,200 data. Increase in accuracy value by 0.92% compared to Uni-Gram to 88.59% with AUC value of 0.95.

Keywords: *analysis sentiment, support vector machine (SVM), text mining, word n-gram*

ABSTRAK

Aplikasi Sambara merupakan inovasi dari Bapenda Jabar untuk pelayanan pajak kendaraan bermotor. Aplikasi Sambara diharapkan memberikan efesiensi, efektifitas, dan perbaikan pelayanan. Keberhasilan aplikasi dapat diketahui dengan melakukan *analysis sentiment review*. *Analysis sentiment* bertujuan untuk mendeteksi polaritas di dalam teks berupa opini negatif atau positif., dengan *menggunakan text mining*. Pada tahapan *text processing* ditambahkan fitur *Word* N-Gram sebagai pendekatan identifikasi kata dan untuk klasifikasinya menggunakan metode *Support Vector Machine* (SVM). Penelitian ini bertujuan untuk mengetahui penerapan *Word* N-Gram, hasil nilai akurasi dengan menggunakan metode SVM, dan mengetahui seberapa besar pengaruh penerapan Word N-Gram terhadap nilai akurasi. Hasil nilai akurasi tertinggi pada penelitian ini sebesar 89.00% dengan nilai AUC 0.944 (*excellent classification*) pada jumlah data 900, namun saat dilakukan penerapan *Bi-gram* dan *Tri-gram* menghasilkan penurunan akurasi. Nilai akurasi dengan kenaikan tertinggi yaitu pada penerapan *Tri-gram* dengan jumlah data 1.200. Kenaikan nilai akurasi sebesar 0.92% dibandingkan dengan Uni-Gram menjadi 88.59% dengan nilai AUC 0.954.

Kata Kunci: *analysis sentiment, support vector machine (SVM), text mining, word n-gram*

1 PENDAHULUAN

Aplikasi Sambara merupakan inovasi dari Bapenda Jabar untuk pelayanan pajak kendaraan bermotor. Aplikasi Sambara diharapkan memberikan efesiensi, efektifitas, dan perbaikan pelayanan [1]. Keberhasilan aplikasi dapat diketahui dengan melakukan *analysis sentiment review*. *Sentiment analysis* atau *opinion mining* adalah bidang studi yang menganalisis opini, sentimen, evaluasi, penilaian, sikap, dan emosi orang terhadap entitas seperti produk, layanan, organisasi, individu, masalah, peristiwa, topik, dan atributnya [2] dengan tujuan mendeteksi polaritas didalam teks berupa opini negatif atau positif melalui proses *text mining* [3]. *Text mining* merupakan penerapan konsep

dan teknik dari data mining, hanya saja diperlukan lebih banyak tahapan, karena memiliki karakteristik yang lebih kompleks dibandingkan dengan data biasa [4]. Hal ini dikarenakan *text mining* menggunakan data tidak terstruktur atau minimal semiterstruktur [5].

N-gram merupakan pendekatan identifikasi dan analisis fitur popular yang digunakan dalam pemodelan bahasa dan bidang pemprosesan bahasa alami [6]. Penambahan proses n-gram ini telah dilakukan oleh beberapa penelitian yang menghasilkan penambahan nilai akurasi. Pada penelitian yang dilakukan oleh Indrayuni & Wahyudin [7] mendapatkan peningkatan akurasi 2%, sedangkan pada penelitian yang dilakukan oleh Pramono hasil akurasinya meningkat 6,7% [8].

Pada penelitian yang membahas komparasi metode menyatakan bahwa hasil akhir metode SVM lebih besar dibandingkan dengan metode lain. Seperti penelitian yang telah dilakukan oleh Elly Indrayuni yang mengkomparasikan metode SVM dengan *Naïve Bayes* menghasilkan akurasi lebih tinggi 5.5% pada studi kasus analisa sentimen review Film [9]. Peneliti Utami juga melakukan analisis sentimen opini publik yang mengkomparasi algoritma SVM berbasis PSO dengan *K-Nearest Neighbor* Berbasis PSO, menghasilkan algoritma SVM lebih tinggi 13,05% [10]. Metode SVM juga terbukti lebih tinggi 44% dibandingkan dengan *Lexicon-Based* berdasarkan penelitian yang dilakukan oleh Najib studi kasus Analisis Sentimen Berbasis Ontologi pada Kampanye Pilpres Indonesia Tahun 2019 di Twitter [11].

Mengacu pada penelitian–penelitain terdahulu maka kami melakukan penelitian penerapan N-Gram dengan menggunakan metode SVM untuk mengklasifikasikanya dengan studi kasus *review* aplikasi Sambara di Google Play. Namun pada penelitian ini kami menggunakan jenis *Word* N-Gram dengan jenis t*erm* Uni-Gram, Bi-Gram, dan Tri-Gram. Kami melakukan pengembangan model dari penelitian - penelitian terdahulu, dengan menambahkan fitur – fitur di tahapan *text processing* yang dapat dilihat pada Gambar 2, dan menyesuaikan fitur dengan dataset yang digunakan yaitu bahasa Indonesia. Sehingga pada penelitian ini akan terlihat hasil dari dimensi dataset setelah dilakukannya setiap fitur *text processing* yang digunakan, lama proses klasifikasi, dan pengaruh penerapan *Word* N-Gram terhadap nilai akurasinya.

2 TINJAUAN PUSTAKA

Penelitian analisis sentimen sudah dilakukan oleh beberapa peneliti sebelumnya seperti pada penelitian [12] yang membahas menganai analisis sentiment Pelanggan Toko JD.ID menggunakan metode *Naïve Bayes Classifier* berbasis konversi ikon emosi menghasilkan nilai akurasi 98%.

Penelitian [13] mengenai komparasi algoritma *Support Vector Machine* dan *Naïve Bayes* dengan algoritma genetika pada analisis sentimen calon gubernur jabar 2018 – 2023 menghasilkan Algoritma SVM berbasis GA menghasilkan rata-rata akurasi 93,03% dengan AUC 0,869 dan Algoritma *Naive Bayes* berbasis GA menghasilkan rata-rata akurasi 92,85% dengan AUC 0,543. Penelitian [14] oleh Mahendra mengenai sentimen pengguna Gopay menggunakan Metode *Lexicon Based* dan *Support Vector Machine* menghasilkan nilai akurasi 84,38% untuk *kernel polynominal*.

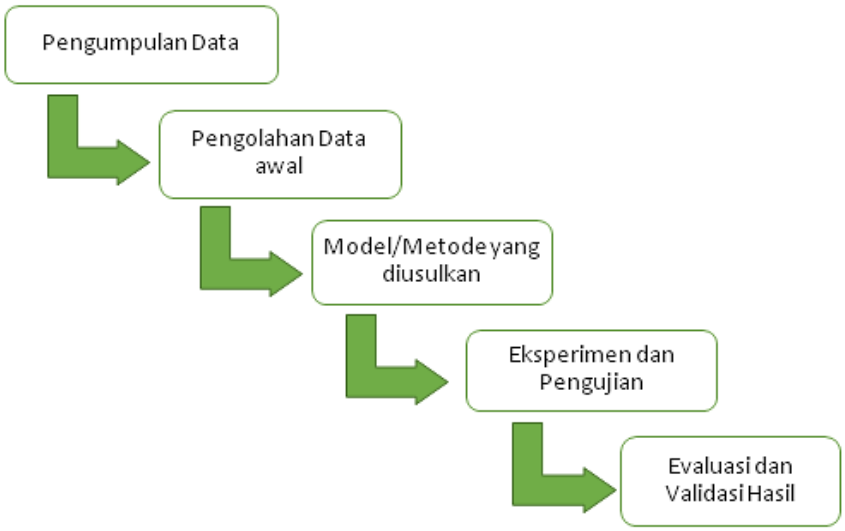
Mengacu penelitian [15] mengenai analisis sentimen pada media sosial twitter menggunakan *naïve bayes classifier* dengan ekstrasi fitur n-gram menghasilkan nilai akurasi 89,67% setelah menggunakan karakter n-gram menjadi 92%. Penelitian [16] klasifikasi teks pengaduan pada sambat online menghasilkan bahwa algoritma KNW-KNN mampu melakukan klasifikasi teks pengaduan dengan nilai tetangga k terdekat yang paling optimal adalah 3, dengan rata-rata hasil presentase percision sebesar 77.85%.

Penelitian [17] mengenai klasifikasi *Movie review of* IMDb dalam bahasa inggris dengan melakukan proses *Stopword*, dan *Numeric and special character removal* pada tahapan *text processing*, dan bereksperimen dengan jumlah data metode dan penerapan N-gram untuk menghitung *vectorizer* supaya mendapatkan nilai akurasi yang lebih.

3 METODE PENELITIAN

Menurut Dawson metode penelitian yang biasa digunakan secara umum ada empat (4) metode yaitu penelitian Tindakan, eksperimen, studi kasus, dan survei [18]. Pada penelitian ini metode yang digunakan yaitu metode eksperimen dengan tahapan yang ada pada Gambar 1.

E-ISSN:2540-9719
ISSN:2302-8149



Gambar 1. Rancangan Penelitian

3.1. Pengumpulan Data

Pengumpulan data pada penelitian ini menggunakan data publik atau disebut dengan data sekunder yang diperoleh dari situs Google Play dengan alamat web *https://play.google.com/store/apps/details?id=id.go.bapenda.sambara&hl=id&showAllReviews=true* Data yang digunakan untuk penelitian ini yaitu *review* aplikasi sambara mulai dari tanggal 22 Januari 2018 sampai dengan tanggal 26 Mei 2020. *Review* yang didapatkan sebanyak 2001 ulasan terdiri dari 772 ulasan sentimen negatif, dan 1.229 ulasan positif.

3.2. Pengolahan Data Awal

Pengolahan data awal ini yaitu melakukan *text processing* dengan tahapan–tahapan sebagai berikut :

1. *Select Attribute* yaitu melakukan pemilihan *attribute* yang akan digunakan.
2. *Transform Cases* (*lowercase*) yaitu mengubah setiap huruf pada kalimat menjadi non kapital.
3. *Tokenization* merupakan pemecahan kalimat menjadi beberapa token dan sekaligus menghilangkan tanda baca.
4. *Stopword* merupakan proses menghilangkan kata yang sering bermunculan dan tidak memiliki makna terhadap kalimat tersebut.
5. *Stemming* yaitu proses untuk mengubah semua kata menjadi kata dasar.
6. *Tokens (by length)* merupakan filter untuk memilih kata sesuai dengan syarat panjang karakter dari kata yang ditentukan.
7. *Generate N-Gram (Word)* Pada tahapan ini, setiap kalimat akan dilakukan pemenggalan per kata. Pemodelan N-gram adalah pendekatan identifikasi dan analisis fitur yang populer digunakan dalam pemodelan bahasa dan bidang pemrosesan bahasa alami. N-gram adalah urutan item yang berdekatan dengan panjang n. Itu bisa berupa urutan kata, *byte*, suku kata, atau karakter. Model N-gram yang paling banyak digunakan dalam kategorisasi teks adalah n-gram berbasis *Word* dan *Characters* [6]. Model N-gram dalam analisis sentimen membantu menganalisis sentimen teks atau dokumen [17]. Karakteristik pada N-gram sebagai berikut [19]:
   a. Dapat berfungsi dengan baik walaupun terdapat kesalahan tekstual
   b. Dapat berjalan secara efisien, membutuhkan penyimpanan yang sederhana
   c. Dan waktu proses yang cepat.

   Jenis N-Gram berdasarkan dari jumlah potongan gram substring yaitu uni-gram, bi-gram, tri-gram, quad-gram dan seterusnya sesuai dengan jumlah n dalam n-gram [20]. Pada penelitia ini akan menggunakan samapi dengan jenis *Tri-Gram* Contoh dari proses pemenggalan kalimat menggunakan N-Gram *Word* dengan jumlah pemotongan pada gram, jika hasil inputan “bagus dan memudahkan dalam mengecek jumlah pajak dan tunggakan” maka :
   - Uni-gram :{“bagus”,”dan”,”memudahkan”,”dalam”,…,“tunggakan”}

- Bi-gram : {“bagus”,”bagus_dan”,…,”dan_tunggakan”,”tunggakan”}
- Tri-gram : {“bagus”,”bagus_dan”, “bagus_dan_memudahkan”…”tunggakan”}

8. *Remove Duplicate* merupakan tahapan menghapus record yang memiliki *text* yang sama pada attribute ulasan yang sudah dilakukan proses pada tahapan–tahapan sebelumnya.

Proses pengolahan data awal ini mengahsilkan nilai perkata atau disebut dengan *vector* dengan menggunakan *Term Frequency Inverse Document Frequency* (TF-IDF) yaitu merupakan salah satu cara membobotan kata hasil dari ide Salton yang menggabungkan dengan ide IDF hasil Jones KS pada tahun 1973. Pada tahun 1988 beliau mulai memasukan kata–kata fitur dan bobot, dan membahas eksperimen dengan hasil jika frekuensi suatu kata atau frasa tinggi, dan diartikel lain jarang muncul, maka dianggap memiliki kemampuan yang baik untuk membedakan dan cocok untuk klasifikasi, semakin luas cakupan kata yang muncul dalam dokumen, atribut yang lebih rendah yang membedakan konten dokumen itu rendah (IDF) [21].

Untuk menghitung bobot (W) pada algoritma TF-IDF terhadap masing–masing dokumen menggunakan rumus yaitu :

$$\boldsymbol{W_{ij} = tf_{ij} \times IDF_j}$$

atau

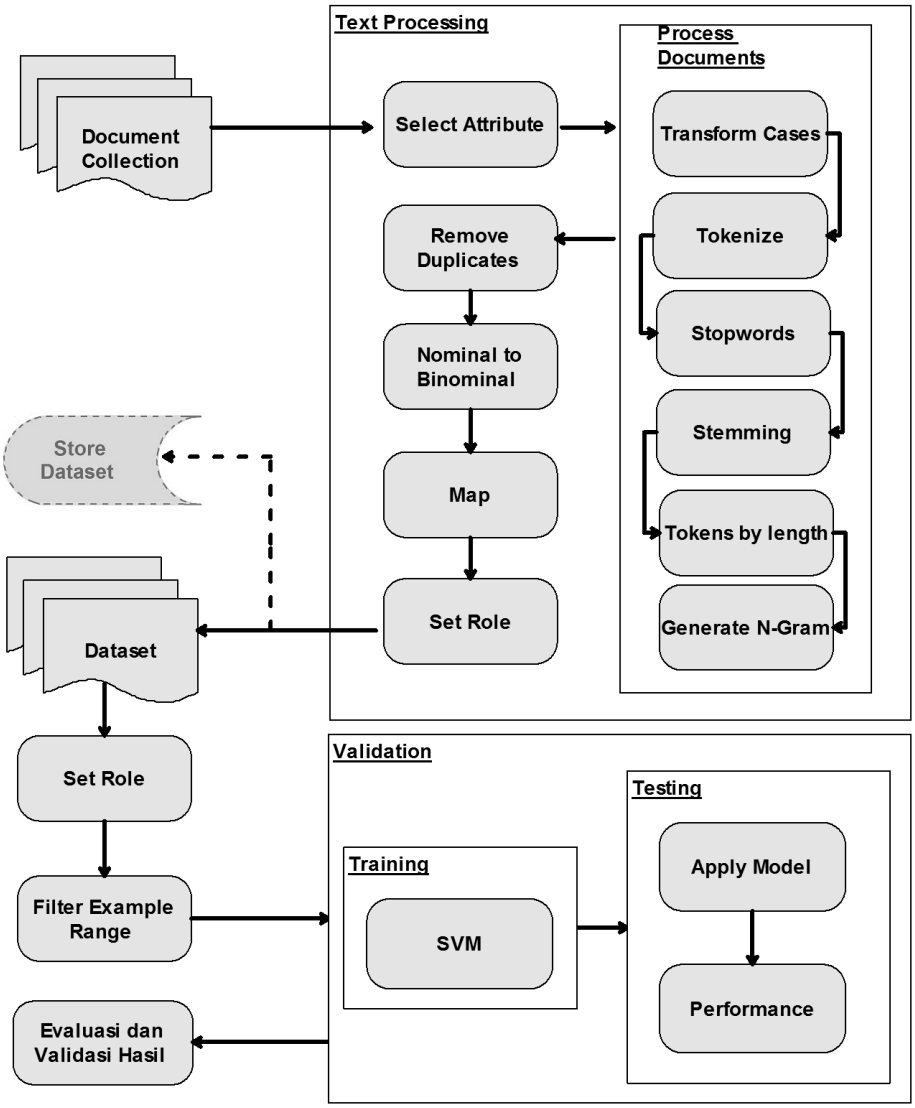
$$\boldsymbol{W_{ij} = tf_{ij} \times \log(D/df_j)}$$

Dimana :
W*ij* = bobot term (tj) terhadap dokumen (di)
tfij = jumlah kemunculan term (tj) dalam dokumen (di)
D = jumlah semua dokumen yang ada dalam data
tf = Banyaknya kata yang dicari pada sebuah dokumen
IDF = *Inverse document frequency*

3.3. Model/ Metode yang Diusulkan

Model penelitian yang diusulkan untuk penerapan *Word* N-gram menggunakan metode SVM pada *review* aplikasi Sambara dalam bahasa Indonesia dapat dilihat pada Gambar 2. Pada Gambar 2 akan terlihat secara keseluruhan tahapan eksperimen mulai dari *Document Collection* atau pengumpulan dokumen *review* sampai dengan tahapan akhir yaitu evaluasi dan validasi dengan pemaparan tahapan sebagai berikut:

1. *Document Collection* merupakan tahapan *import* data yang terkumpul ke dalam aplikasi Rapidminer untuk dilakukan proses sesuai dengan model yang diusulkan. Tahapan ini kami menggunakan *Read* XLS, kemudian *Loop File* dan *Retrieve File* sehingga data dalam format XLS masuk kedalam database di Aplikasi.
2. Masuk ke tahapan pengolahan data awal atau *text processing* yang sudah dilakukan yang telah dipaparkan sebelumnya. Fitur *Store* Dataset ditambahkan untuk menyimpan hasil dari *text processing* berupa dataset kedalam *directory* Rapidminer.
3. Dataset yang sudah dihasilkan kemudian masuk kedalam fitur *set role* dengan fungsi menentukan *attribute* yang menjadi label untuk diklasifikasikan. Klasifikasi yaitu salah satu model analatik prediktif dalam data mining. Proses klasifikasi dapat dicontohkan secara sederhana saat pemain golf menikmati bermain jika memenuhi kriteria seperti kondisi cuaca, lembab, langit, dan suhu, maka akan diprediksi seseorang lebih menyukai bermain atau tidak sesuai dengan history yang dimiliki Penjelasan diatas dapat disimpulkan bahwa klasifikasi merupakan membuat prediksi berdasarkan history yang dijadikan sampel pembelajaran yang diurutkan kedalam dua atau lebih kelas yang berbeda. Klasifikasi memiliki kemiripin dengan prediksi, namun perbedaannya yaitu prediksi dibangun untuk memprediksi model yang bernilai continue [22].
4. Dalam penelitian ini akan dilakukan pengujian dalam 5 kategori jumlah data, maka setelah *set role* ditambahkan fitur *filter example* untuk menentukan jumlah data yang akan diuji.



Gambar 2. Model yang Diusulkan untuk Klasifikasi *Analysis Review* Aplikasi Sambara

5. Untuk melakukan klasifikasi maka diperlukannya proses validasi. Seperti terlihat pada Gambar 2 ada tahapan validasi, dengan jenis yang digunakan yaitu *cross validation* (K-10). Penggunaan *cross validation* menempatkan *Training* untuk melatih model dan *Testing* untuk pengujian dan melakukan pengukuran didalam validasi. *Cross Validation* (K-10) akan membagi dataset secara terpisah dengan ukuran yang sama, kemudian model dilatih oleh subset data latih dan divalidasi oleh subset data uji sebanyak 10 kali. Untuk metode yang diusulkan pada proses training yaitu metode *Support Vector Machine* (SVM). Metode SVM merupakan sebuah konsep pengklasifikasian dengan menggunakan sebuah garis yang didefinisikan sebagai garis batas antara dua buah kelas [23]. SVM dikembangkan melalui *hyperplane* pemisah yang optimal kondisi linear yang dapat dipisahkan seperti ilustrasi pada gambar 1 [24]. Menurut Minghe Sun SVM memiliki kerangka kerja dengan sifat–sifat berikut [25].
   a. SVM *is a sparse technique*
   b. SVM *is a kernel technique*
   c. SVM *is a maximum margin separator*

   Untuk memisahkan dua klasifikasi melalui *hyperplane* yang optimal menggunakan persamaan berikut :

$$\boldsymbol{f(x) = w.x + b}$$

atau

$$\boldsymbol{f(x) = \sum_{i=1}^{n} a_i\, y_i K(x, x_i) + b}$$

Harus berada diantara duakelas sampel *hyperplane* pemisah yang optimal

$$\boldsymbol{y_i(wx_i + b) \geq 1\ (i = 1, 2, \ldots, n)}$$

dimana :

w : beban *vector* (garis tegak lurus)
x : titik data masukan SVM
b : bias
a_i : nilai bobot setiap titik data
$K(x, x_i)$: fungsi karnel

Setelah data dilatih menggunakan metode SVM, kemudian data masuk kedalam data proses pengujian dengan menggunakan fitur *Apply Model* untuk menerapkan model yang sama pada tahapan *training* ke proses *testing* dengan menguji dan menghasilkan nilai *performance* dari model yang diusulkan berupa nilai akurasi dan AUC.

3.4. Eksperimen dan Pengujian Model

Eksperimen pada penilitian ini mengusulkan penerapan N-gram saat *text processing*. N-Gram yang diusulkan yaitu *Word* N-Gram dengan jenis *Unigram*, *Bigram*, dan *Trigram*. Eksperimen ini juga mengusulkan metode *Support Vector Mechine* yang merupakan metode paling baik berdasarkan hasil penelitian – penelitian sebelumnya, untuk digunakan dalam proses klasifikasi pada *sentiment review* aplikasi Sambara.

Dataset hasil dari pengolahan data awal yang telah diterapkan tiga jenis *Word* N-Gram, kemudian diklasifikasikan menggunakan metode yang diusulkan dengan jumlah data yang berbeda–beda. Dataset tersebut dibagi menjadi 5 jumlah data yang berbeda yaitu 300 data, 600 data, 900 data, 1200 data, dan 1.332 data. Masing–Masing diberlakukan hal yang sama dalam eksperimen dan pengujiannya.

3.5. Evaluasi dan Validasi Hasil

Validasi model memungkinkan anda untuk memprediksi kinerja model anda pada set data yang tidak digunakan dalam pelatihan [26]. Salah satunya yaitu *Cross Validation*. *Cross validation* merupakan salah satu metode pemilihan model yang paling popular dalam statistik dan *machine learning* karena kesederhanaan konsep dan penerapan yang luas [27]. Cara standar yang digunakan yaitu *cross validation* kelipatan 10 (K-10). Data dibagi secara acak menjadi 10 bagian dengan perkiraan proporsi sama seperti dataset lengkap. Pada giliran satu bagian menjadi pengujian dan sembilan dari sepuluh menjadi data pelatihan [28].

a. Confusion Matrix

Menurut Mariette dan Awad mengemukakan bahwa "*confusion matrix* (aka error matrix) matriks kebingungan (alias *error matrix*). Matriks yang memvisualisasikan kinerja algoritma klasifikasi menggunakan data dalam matriks" [29] yang terlihat pada Tabel 1.

Tabel 1. *Confusion Matrix*

Confusion Matrix		**Predicted**	
		Positive	**Negative**
Actual	Positive	TP	FN
	Negative	FP	TN

Dengan :
TP (*True Positive*) : Hasil yang didentifikasi dengan benar sebagai positif
TN (*True Negative*) : Hasil yang didentifikasi dengan benar sebagai negatif
FP (*False Positive*) : Hasil yang didentifikasi dengan salah sebagai positif

FN (*False Negative*) : Hasil yang didentifikasi dengan salah sebagai negatif

Performa yang akan dibandingkan dalam penelitian ini yaitu menjadi 3 kategori dengan 5 jumlah data yang berbeda sehingga ada 15 performa yang akan diuji seperti pada Tabel 2.

Tabel 2. Daftar Performa yang akan Dibandingkan

Jenis Gram	Jumlah Data				
	300	600	900	1200	1332
Uni	Uni_300	Uni_600	Uni_900	Uni_1200	Uni_1332
Bi	Bi_300	Bi_600	Bi_900	Bi_1200	Bi_1332
Tri	Tri_300	Tri_600	Tri_900	Tri_1200	Tri_1332

4 HASIL DAN PEMBAHASAN

Penelitian ini dilakukan menggunakan Aplikasi RapidMiner Studio versi 9.7 dengan menggunakan hardware Laptop processor Intel Core i5 dengan RAM 8GB dan harddisk 1TB.

4.1. Pengumpulan Data

Pengumpulan data didapatkan dari https://play.google.com/store/apps/details?id=id.go.bapenda.sambara&hl=id&showAllReviews=true yang kemudian dirapihkan menjadi dataset seperti contoh pada Tabel 3.

Tabel 3. Data *Review* Aplikasi Sambara

ID	Nama Pengguna	Tanggal Ulasan	Bintang	Ulasan	Sentimen
D1	Ifni Anasta .k	17-12-19	1	PUNYA SAYA KENAPA MUNCUL TULISAN TIDAK DI TEMUKAN SETELAH MEMASUKAN NO KTP DAN NO RANGKA?	Negatif
D2	Tita Arianti	21-09-19	2	Mau dapetin kode bayar ko gagal terus,padahal no ktp sm no rangka nya udah bener,di keterangan,data tidak ditemukan??	Negatif
D3	Boo Young	03-09-19	5	Aplikasi mantul, waktu pengesahan stnk di samsat. datang,print dan cap. Pulang deh Orang" yang ngantri pada bengong wkwkwk	Positif
D4	Imam Arif santosa	16-10-19	5	Bagus sangat membantu sekali, karena apk sambara jadi tau berapa yang harus d bayar dan tau berapa denda yang harus d bayar	Positif
D5	Bambang Sutiadi	13-11-19	5	Mempermudah pembayaran pajak, dan mempermudah data kendaraan yg sdh bayar dan blm byr	Positif
D6	Pengguna Google	30-11-18	1	Gak bisa cek info PKB sering error	Negatif

4.2. Pengolahan Data Awal

Tahapan pertama dalam eksperimen ini yaitu melakukan tahapan *text processing* yang selanjutnya akan dilakukan penerapan N-Gram.

a. *Tokenization*

Tahapan ini akan memecahkan kalimat menjadi token - token kata dan sekaligus menghilangkan tanda baca, dengan menghsilkan jumlah frekuesi kata yang muncul.

b. *Transform Case*

Tahapan ini akan mengubah semua huruf kapital menjadi huruf kecil (*lower case*) oleh Rapidminer, yang akan menghindari dobel kata. Seperti contoh kata “RANGKA” pada D1 dengan kata “rangka” pada D3 akan menjadi 2 *attribute* jika tidak dilakukan *transform case*.

c. *Stopword*
Tahapan ini akan menghilangkan kata yang sering digunakan walaupun mencerminkan makna pada review tersebut maka kata akan dihilangkan. Daftar kata yang akan dihilangkan didapatkan dari website https://github.com/masdevid/ID-Stopwords. Kata - kata yang dihilangkan seperti kata "mau", "dan", dan "di".

d. *Steamming*
Pada tahapan *stemming*, kata yang berimbuhan akan diubah menjadi kata dasar seperti beberapa kata contoh dibawah ini:
"Memasukan" → "Masuk" dan "Ditemukan" → "temu"

e. *Filter Tokens (by length)*
Pada tahapan *tokens by length* kata akan disaring berdarkan panjang huruf. Kata yang kurang dari 3 karakter atau huruf atau lebih dari 25 huruf maka akan dihilangkan. Sebagai contoh pada D1 kata yang akan dihilangkan yaitu "di" dan "no" karena kurang dari 3 karakter.

f. *TF-IDF*
Tahapan ini akan menghitung nilai dari setiap *attribute* dalam setiap *record* yang ada didataset dengan rumus TF-IDF. Salah satu contoh perhitungannya seperti:
Kata "Bayar" terdapat 4 kali kata dalam jumlah 6 dokumen maka rumusnya

$$IDF_{ij} = \log(D/df_j)$$

$$IDF_{(bayar)} = \log(6/4)$$

$$IDF_{(bayar)} = \log(1.5)$$

$$IDF_{(bayar)} = 0.176$$

Setelah mendapatkan nilai IDF selanjutnya dihitung kembali menggunakan rumus *Term Frequenxy* yang dapat dilihat perhitungannya pada Tabel 4.

Tabel 4. Daftar Performa yang akan Dibandingkan

Hitungan	D1	D2	D3	D4	D5	D6
Frequency	0	**1**	0	**1**	**2**	0
TF-IDF	0.176*0	**0.176*1**	0.176*0	**0.176*1**	**0.176*2**	0.176*0

Hasil IDF kemudian dikalikan dengan *Frequency* nya, kemudian menghasilkan nilai TF-IDF.

g. *Remove Duplicate*
Tahapan ini akan menghapus semua data yang sama, hanya akan dimasukan satu data saja yang berbeda. Jumlah data yang semula 2.001 untuk dilakukan *text processing*, menjadi berkurang hanya ada 1.332 data.

h. *N-Gram*
Penerapan *Word* N-Gram dengan jumlah data 1.332 menghasilkan jumlah attribute sebanyak 1.696 pada *uni-gram*, 7.251 attribute pada *Bi-Gram*, dan 13.238 *attribute* pada *Tri-Gram.*

Tabel 4. Hasil Pengolahan Data Awal

Proses	Jumlah Data	Jumlah Attribute
Tokenisasi	1.493	2.919
Transform	1.477	2.474
Stopword	1.380	2.164
Stemming	1.361	1.799
Token by Length	1.332	1.696
Bi Gram	1.332	7.261
Uni Gram	1.332	13.238

E-ISSN:2540-9719
ISSN:2302-8149

Hasil dari proses pengolahan awal atau *text processing* menghasilkan jumlah data dan *attribute* yang terlihat pada Tabel 4.

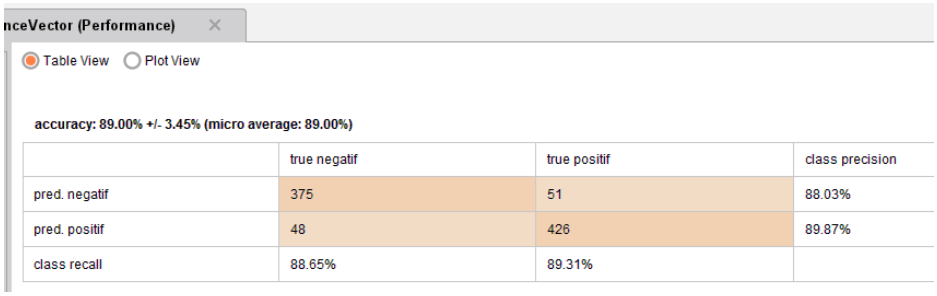
4.3. Evaluasi dan Validasi

Hasil dataset dari *text processing* selanjutnya dilakukan klasifikasi menggunakan menggunakan metode SVM dengan validasi menggunakan *cross validation* K-10. Pada penelitian ini dilakukan pemprosesan dengan jumlah data yang berbeda untuk mengetahui jumlah data yang tepat untuk digunakan.

Jumlah Data	Jumlah Data	Data		Uni-Gram			Bi-Gram			Tri-Gram		
		Positif	Negatif	Waktu	Accuracy	AUC	Waktu	Accuracy	AUC	Waktu	Accuracy	AUC
Uni-Gram	300	170	130	00.01.08	84.00%	0.93	00.21.32	82.67%	0.934	00.20.52	77,67%	0.936
	600	275	325	00.03.58	87.08%	0.937	00.57.54	87.83%	0.939	01.37.13	86,83%	0.942
	900	477	423	00.09.48	89.00%	0.944	01.40.25	88.89%	0.948	04.16.50	88,00%	0.945
	1200	599	601	00.16.14	87.58%	0.951	04.09.19	87.50%	0.951	07.35.53	88,50%	0.954
	1332	576	756	00.18.33	88.21%	0.958	03.44.41	88.59%	0.957	09.15.28	88,44%	0.961

Gambar 4. Perbandingan Hasil Penelitian

Hasil evaluasi dari penelitian ini dapat dilihat dari Gambar 4 bahwa nilai akurasi tertinggi terdapat pada *Uni-Gram* dengan jumlah data 900 yaitu 89.00%. Nilai akurasi Bi-Gram dengan jumlah data 1.332 menghasilkan nilai akurasi lebih tinggi dibandingan dengan nilai akurasi yang dihasilkan Uni-gram. Nilai akurasi ¬Uni-gram sebesar 88.21% sedangankan Bi-gram 88.59%. Penerapan tri-gram menghasilkan nilai akurasi terbesar dari jumlah data 1.200 dengan nilai akurasi 88.50% dan AUC 0.954. Nilai akurasi tertinggi dibandingkan dengan penerapan uni-gram dan bi-gram.



	true negatif	true positif	class precision
pred. negatif	375	51	88.03%
pred. positif	48	426	89.87%
class recall	88.65%	89.31%	

Gambar 5. *Confusion Matrix Uni-Gram* 900 Data

Pada Gambar 5 menampilkan *Confusion* Matrix untuk menghitung nilai akurasi didapatkan dengan rumus :

Accuracy (AC) $= \frac{426+375}{426+375+48+51} =$ **89.00%**

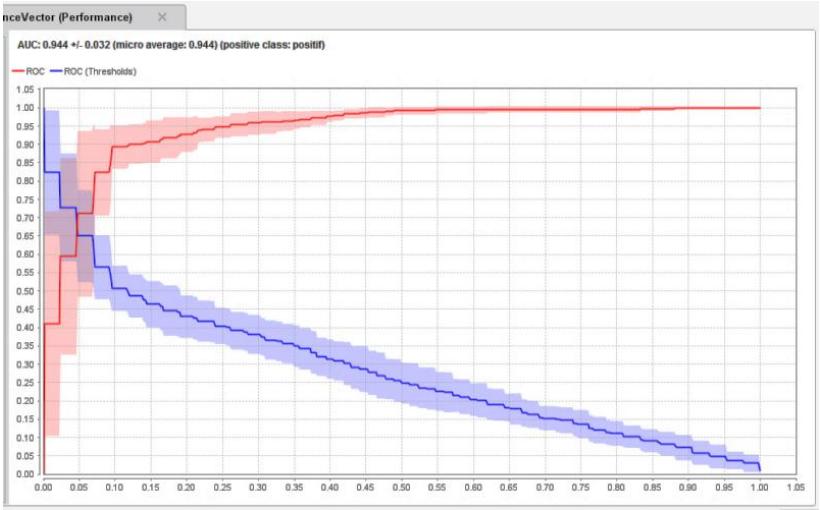
Selain nilai akurasi terdapat nilai *precision positive* yang merupakan rasio predikitif dari semua kelas positif yang didapatkan dengan rumus :

Precision (P) $= \frac{426}{426+48} =$ **89.87%**

Sebagai gambaran keberhasilan model dalam mengklasifikasikan kembali maka dicari nilai *recall* dengan rumus :

Recall(R, True positive rate) $= \frac{426}{426+51} =$ **89.31%**

E-ISSN:2540-9719
ISSN:2302-8149



Gambar 6. ROC *Uni-Gram* 900 Data

Selain menggunakan *confusion matrix* untuk mengevaluasi eksperimen ini, evaluasi eksperimen juga melihat nilai AUC yang dapat dilihat pada Gambar5 yaitu sebesar 0.944 termasuk kedalam *excellent classification.*

5 KESIMPULAN

Text processing sebelum dilakukan penerapan *Word* N-gram, berhasil mengurangi dimensi data yang semula berjumlah 2.919 *attribute* menjadi 1.696 attribute. Setelah dilakukan penerapan *Bi-Gram* dan *Tri*-Gram pada tahapan *text processing* menambah dimensi data menjadi 7.261 *attribute* untuk *Bi-Gram* dan 13.238 *attribute* untuk *Tri*-Gram. Penggunaan fitur *remove duplicate* mengurangi jumlah data, karena fitur tersebut akan menghapus data dengan isian yang sama. Data yang dikumpulkan sebanyak 2.001 data, setelah dilakukan *text processing* menjadi 1.332 data. Waktu proses klasifikasi dipengaruhi oleh jumlah *attribute* dan jumlah data, maka dari itu waktu proses terlama terjadi saat melakukan proses klasifikasi data penerapan *Tri-Gram* dengan jumlah data 1.332 yaitu 9 jam 15 menit 28 detik. Hasil akurasi tertinggi pada penelitian ini sebesar 89.00% dengan nilai AUC 0.944 (excellent classification) pada jumlah data 900, namun saat dilakukan penerapan Bi-gram dan Tri-gram menghasilkan penurunan akurasi. Nilai akurasi dengan kenaikan tertinggi yaitu pada penerapan tri-gram dengan jumlah data 1.200. Kenaikan nilai akurasi sebesar 0.92% dibandingkan dengan Uni-Gram menjadi 88.59% dengan nilai AUC 0.954. Untuk itu dapat disimpulkan bahwa penerapan Word N-Gram akan meningkatkan nilai akurasi jika data yang digunakan dalam jumlah banyak.

Penelitian ini dapat digunakan sebagai referensi baru untuk penelitian berikutnya, dan bisa dikembangkan menjadi sebuah aplikasi *sentiment analysis*. Pembuatan aplikasi tersebut bertujuan untuk mendapatkan informasi yang harus diperbaiki dan dipertahankan dalam sebuah aplikasi secara cepat dan akurat. Untuk penelitian selanjutnya disarankan membuat daftar *stopword* dari *wordlist* penerapan N-gram sebelum dilakukan proses klasifikasi. Hal ini mungkin bisa mengurangi dimensi yang akan membuat *wordlist* yang digunakan lebih unik dan memberikan nilai *weight* setiap katanya lebih besar. Melakukan pengembangan model penelitian yang lebih kompleks dengan menambahkan fitur – fitur yang ada, dan juga mencoba melakukan dengan metode yang lainnya untuk menemukan model yang terbaik.

REFERENSI

[1] B. Jabar, “Cek Pajak Kendaran Melalui Aplikasi Sambara.” https://bapenda.jabarprov.go.id/2018/08/14/cek-pajak-kendaraan-melalui-aplikasi-sambara/.

[2] B. Liu, “Sentiment Analysis and Opinion Mining,” *Synth. Lect. Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, May 2012, doi: 10.2200/S00416ED1V01Y201204HLT016.

[3] S. Gupta, “Sentiment Analysis: Concept, Analysis and Applications,” *Toward Data Science*, 2018. https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17.

[4] B. Susanto, "Text dan Web Mining," 2020. http://lecturer.ukdw.ac.id/budsus/pdf/textwebmining/TextMining_Kuliah.pdf (accessed Apr. 10, 2019).

[5] E. Junianto and D. Riana, "Penerapan PSO Untuk Seleksi Fitur Pada Klasifikasi Dokumen Berita Menggunakan NBC," *J. Inform.*, vol. 4, no. 1, pp. 38–45, 2017, [Online]. Available: https://ejournal.bsi.ac.id/ejurnal/index.php/ji/article/view/1810.

[6] H. Ahmed, I. Traore, and S. Saad, "Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques," vol. 10618, I. Traore, I. Woungang, and A. Awad, Eds. Cham: Springer International Publishing, 2017, pp. 127–138.

[7] E. Indrayuni and M. Wahyudi, "PENERAPAN CHARACTER N-GRAM UNTUK SENTIMENT ANALYSIS REVIEW HOTEL MENGGUNAKAN ALGORITMA NAIVE BAYES," *Konfrensi Nas. Ilmu Pengetah. dan Teknol.*, 2015.

[8] F. Pramono, Didi Rosiyadi, and Windu Gata, "Integrasi N-gram, Information Gain, Particle Swarm Optimation di Naïve Bayes untuk Optimasi Sentimen Google Classroom," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 3, pp. 383–388, 2019, doi: 10.29207/resti.v3i3.1119.

[9] E. Indrayuni, "Komparasi Algoritma Naive Bayes Dan Support Vector Machine Untuk Analisa Sentimen Review Film," *J. Pilar Nusa Mandiri*, vol. 14, no. 2, p. 175, 2018, doi: 10.33480/pilar.v14i2.918.

[10] L. A. Utami, "Analisis Sentimen Opini Publik Berita Kebakaran Hutan Melalui Komparasi Algoritma Support Vector Machine Dan K-Nearest Neighbor Berbasis Particle Swarm Optimization," *Pilar Nusa Mandiri*, vol. 13, no. 1, pp. 103–112, 2017.

[11] A. C. Najib, A. Irsyad, G. A. Qandi, and N. A. Rakhmawati, "Perbandingan Metode Lexicon-based dan SVM untuk Analisis Sentimen Berbasis Ontologi pada Kampanye Pilpres Indonesia Tahun 2019 di Twitter," *Fountain Informatics J.*, vol. 4, no. 2, p. 41, Nov. 2019, doi: 10.21111/fij.v4i2.3573.

[12] F. V. Sari and A. Wibowo, "Analisis Sentimen Pelanggan Toko Online Jd. Id Menggunakan Metode Naïve Bayes Classifier Berbasis Konversi Ikon Emosi," *Simetris J. Tek. Mesin, Elektro dan Ilmu Komput.*, vol. 2, no. 2, pp. 681–686, 2019.

[13] D. Gunawan, D. Ardiansyah, F. Akbar, and A. Salman, "Komparasi Algoritma Support Vector Machine Dan Naïve Bayes Dengan Algoritma Genetika Pada Analisis Sentimen Calon Gubernur Jabar 2018-2023," *Komparasi Algoritm. Support Vector Mach. Dan Naïve Bayes Dengan Algoritm. Genet. Pada Anal. Sentimen Calon Gubernur Jabar 2018-2023*, vol. VI, 2020, doi: 10.31294/jtk.v4i2.

[14] R. Mahendrajaya, G. A. Buntoro, and M. B. Setyawan, "ANALISIS SENTIMEN PENGGUNA GOPAY MENGGUNAKAN METODE LEXICON BASED DAN SUPPORT VECTOR MACHINE," *Komputek*, pp. 52–63, 2019, [Online]. Available: http://studentjournal.umpo.ac.id/index.php/komputek%0AANALISIS.

[15] A. Nugroho, "Analisis Sentimen Pada Media Sosial Twitter Menggunakan Naive Bayes Classifier Dengan Ekstrasi Fitur N-Gram," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 2, no. 2, p. 200, 2018, doi: 10.30645/j-sakti.v2i2.83.

[16] A. A. Prasanti, M. A. Fauzi, and M. T. Furqon, "Klasifikasi Teks Pengaduan Pada Sambat Online Menggunakan Metode N- Gram dan Neighbor Weighted K-Nearest Neighbor (NW-KNN)," vol. 2, no. 2, pp. 594–601, 2018.

[17] A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," *Expert Syst. Appl.*, vol. 57, pp. 117–126, 2016, doi: 10.1016/j.eswa.2016.03.028.

[18] C. W. Dawson, *Project in Computing and Information Systems A Student's Guide*, 2nd ed. Inggris: ADDISON-WESLEY, 2009.

[19] A. Nurfalah and A. A. Suryani, "Analisis Sentimen Berbahasa Indonesia dengan Pendekatan Lexicon-Based Pada Media Sosial," *J. Masy. Inform. Indones.*, 2017.

[20] Z. Pratama, E. Utami, and M. R. Arief, "Analisa Perbandingan Jenis N-GRAM Dalam Penentuan Similarity Pada Deteksi Plagiat," *Creat. Inf. Technol. J.*, vol. 4, no. 4, p. 254, 2019, doi: 10.24076/citec.2017v4i4.118.

[21] A. Guo and T. Yang, "Research and improvement of feature words weight based on TFIDF

algorithm," *Proc. 2016 IEEE Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2016*, pp. 415–419, 2016, doi: 10.1109/ITNEC.2016.7560393.

[22] Tutorlalspoint, "Data Mining - Classification & Prediction," *Tutorlalspoint*, 2020. https://www.tutorialspoint.com/data_mining/dm_classification_prediction.htm (accessed Jun. 07, 2020).

[23] J. Sahertian and A. Sanjaya, "Deteksi Buah Pada Pohon Menggunakan Metode SVM dan Fitur Tekstur," *Semnas Teknomedia*, pp. 19–24, 2017.

[24] G. Shi, "Support Vector Machines," in *Data Mining and Knowledge Discovery for Geoscientists*, Elsevier, 2014, pp. 87–110.

[25] M. Sun, "Support Vector Machine Models for Classification," *Encycl. Bus. Anal. Optim.*, pp. 2395–2409, 2014, doi: 10.4018/978-1-4666-5202-6.ch215.

[26] C. Neale, D. Workman, and A. Dommalapati, "Cross Validation: A Beginner's Guide," *Toward Data Science*, 2019. https://towardsdatascience.com/cross-validation-a-beginners-guide-5b8ca04962cd (accessed May 03, 2020).

[27] J. Lei, "Cross-Validation With Confidence," in *Journal of the American Statistical Association*, 2019, pp. 1–35.

[28] H. Witten, Ian, E. Frank, and M. A. Hall, *Data Mining*. 2008.

[29] M. Awad and R. Khanna, "Machine Learning," in *Efficient Learning Machines*, vol. 6, no. 1, Berkeley, CA: Apress, 2015, pp. 1–18.