

Terbit online pada laman : <http://teknosi.fti.unand.ac.id/>

Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |



Artikel Penelitian

Analisis Sentimen Ulasan Aplikasi Samsat Digital Nasional Menggunakan Algoritma Naive Bayes Classifier

Ruby Chandra^{a,}, Evasaria Magdalena Sipayung^b*^{a,b} Program Studi Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 15 Juli 2024

Revisi Akhir: 30 Agustus 2024

Diterbitkan Online: 31 Desember 2024

KATA KUNCI

analisis sentimen,

confusion matrix,

naïve bayes,

samsat digital nasional

KORESPONDENSI

E-mail: evasaria.sipayung@gmail.com*

ABSTRACT

Samsat Digital Nasional (SIGNAL) merupakan aplikasi layanan pengesahan Surat Tanda Nomor Kendaraan berbasis mobile. Aplikasi ini membuat perubahan dimana masyarakat yang biasanya harus mendatangi kantor Samsat untuk mengurus pajak kendaraan motornya, sekarang dapat dilakukan dimana saja. Semenjak diluncurkannya aplikasi ini, muncul berbagai tanggapan berupa ulasan dari para pengguna SIGNAL. Ulasan-ulasan ini memiliki informasi yang sangat berguna bagi pihak pemilik aplikasi untuk mengetahui kekurangan dan kelebihan dari aplikasinya. Proses perolehan informasi untuk mengetahui kekurangan dan kelebihan secara manual tentunya memakan waktu lama jika ada puluhan ribu ulasan yang harus dibaca satu persatu. Dengan analisis sentimen, proses tersebut dapat dipersingkat dimana sistem dapat secara otomatis memisahkan ulasan berdasarkan informasi yang terkandung didalamnya dengan cepat dan tepat. Penelitian ini menggunakan algoritma Naive Bayes Classifier untuk melakukan klasifikasi terhadap ulasan menjadi 3 kelas yaitu sentimen positif, negatif, dan netral. Penelitian dilakukan dengan menggunakan 1792 data ulasan yang diambil dari kolom ulasan aplikasi SIGNAL di Google Play Store, dimana 1433 data dialokasikan sebagai data latih dan 359 data dialokasikan sebagai data uji. Data teks selanjutnya terbentuk numerik dengan menggunakan pembobotan TF-IDF. Pengujian dilakukan dengan menggunakan confusion matrix untuk mengetahui accuracy, precision, dan recall sistem. Berdasarkan hasil pengujian dengan 359 data uji menggunakan confusion matrix didapatkan nilai accuracy dari sistem adalah 91.643% dengan nilai rata-rata precision sebesar 90.41%, dan nilai rata-rata recall sebesar 74.973%.

1. PENDAHULUAN

Aplikasi Samsat Digital Nasional (SIGNAL) merupakan pelayanan pengesahan Surat Tanda Nomor Kendaraan (STNK) tahunan, pembayaran Pajak Kendaraan Bermotor (PKB), dan pembayaran Sumbangan Wajib Dana Lalu Lintas Angkutan Jalan (SWDKLLJ) [1]. Aplikasi ini dapat digunakan oleh berbagai pihak yang menginginkan kemudahan dalam membayar pajak kendaraan bermotor mereka tanpa harus mendatangi kantor Samsat. Pengguna dapat mengunduh aplikasi ini baik untuk perangkat mobile berbasis android maupun IOS. Meskipun aplikasi ini dapat diperoleh dengan bebas, namun fasilitas dari aplikasi ini hanya diperuntukkan untuk pemilik sah dari

kendaraan tersebut dengan kata lain tidak dapat diwakilkan oleh orang lain.

Aplikasi SIGNAL pertama kali di-publish pada Google Play Store untuk pengguna android pada 16 Juni 2021 [1]. Sampai saat penelitian ini dilaksanakan, terdapat 35 ribu ulasan pengguna yang dapat dilihat pada kolom ulasan aplikasi SIGNAL. Ulasan-ulasan yang diberikan pengguna memiliki beragam informasi, ada yang berupa komentar positif mengenai kepuasan pengguna, komentar negatif mengenai kekecewaan pengguna, dan komentar netral. Informasi-informasi ini dapat sangat berguna bagi pihak pengelola aplikasi SIGNAL dimana mereka dapat mengetahui kekurangan dari aplikasi yang luput dari perhatian, serta kelebihan dari aplikasi SIGNAL. Ulasan memiliki peranan penting bagi pihak pengembang aplikasi dalam menganalisis,

memperbaiki, mengetahui kebutuhan pengguna demi meningkatkan kualitas aplikasi [2] dan respon dan komentar dari masyarakat [3]. Sistem dapat melakukan kategorisasi sentimen teks dalam kelas positif, negatif dan netral terhadap kondisi Covid-19. Prediksi yang dilakukan dengan metode Multinomial Naïve Bayes menghasilkan akurasi sebesar 74%, *precision* sebesar 74% dan juga *recall* sebesar 74%. Sehingga diperoleh nilai AUC adalah 0,74. Ini menandakan bahwa, algoritma yang diusulkan memiliki tingkatan *fair classification* atau (nilai diagnostik sedang/cukup baik) [4]. Hasil evaluasi dari algoritma Multinomial Naïve Bayes (MNB) yang telah dilakukan dalam mengklasifikasikan sentimen ulasan pengguna aplikasi BRImo menggunakan 10-fold *cross validation* mendapatkan hasil evaluasi *confusion matrix* terbaik pada fold-2 dengan nilai akurasi 98,02%, presisi 97,06%, recall 97,06%, dan f1-score 97,06% [5]. Algoritma MNB adalah algoritma kategorisasi teks yang cepat, mudah diimplementasikan [6]. Dengan adanya sebuah sistem analisis sentimen, proses perolehan informasi ini dapat dipangkas karena sistem dapat melakukan pemisahan informasi secara otomatis dan relatif lebih cepat dibandingkan jika harus dilakukan secara manual. Pada penelitian ini dikembangkan sistem analisis sentimen yang dapat melakukan kategorisasi sentimen teks dalam kelas positif, negatif, dan netral terhadap ulasan aplikasi SIGNAL secara otomatis dengan menggunakan algoritma Multinomial Naïve Bayes dan hasil akurasi yang lebih baik.

2. METODE

Pada penelitian ini akan digunakan algoritma Naïve Bayes untuk melakukan analisis sentimen ulasan aplikasi SIGNAL. Berdasarkan hasil penelitian terdahulu, algoritma ini dapat memperoleh tingkat keakuratan yang cukup baik dan waktu pemrosesan yang singkat jika dibandingkan dengan algoritma klasifikasi lainnya [7][8]. Naïve Bayes pertama kali dikemukakan oleh seorang statistikawan Thomas Bayes pada abad ke-18. Metode ini merupakan salah satu metode pengklasifikasian berbasis probabilitas sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan [9]. Metode ini menggunakan pengetahuan yang sudah terjadi di masa lampau untuk memprediksi peluang kejadian di masa depan. Bentuk umum dari rumus Naïve Bayes dapat dilihat pada persamaan 1.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (1)$$

Dimana:

A = hipotesis data B merupakan suatu kelas spesifik

B = data yang belum diketahui label/kelasnya

$P(A|B)$ = probabilitas posterior, probabilitas hipotesis A berdasarkan kondisi B

$P(B|A)$ = probabilitas B berdasarkan kondisi hipotesis

$P(B)$ = probabilitas B

$P(A)$ = probabilitas hipotesis A atau Prior Probability

Untuk dapat diterapkan pada kasus pengklasifikasian teks, maka diperlukan penyesuaian pada rumus dasar dari metode Naïve Bayes itu sendiri. Salah satu jenis dari metode Naïve Bayes yaitu Multinomial Naïve Bayes menghadirkan solusi untuk melakukan pengklasifikasian teks berdasarkan teori Naïve Bayes. Algoritma Multinomial Naïve Bayes bekerja dengan cara mengambil sejumlah kata yang ada dalam setiap dokumen dengan asumsi bahwa dokumen memiliki beberapa kejadian dalam kata dengan panjang yang tidak tergantung pada dari kelasnya dalam dokumen [10]. Persamaan Multinomial Naïve Bayes yang

digunakan untuk mengklasifikasikan dokumen/data teks dapat dilihat pada persamaan 2.

$$P(c|d) \propto P(c) \prod_{k=1}^n P(t_k|c) \quad (2)$$

$P(t_k|c)$ merupakan *conditional probability* dari kata t_k yang muncul pada dokumen dengan kelas c. $P(c)$ merupakan *prior probability* dokumen d sebagai kelas c. Penentuan kelas didapatkan dengan membandingkan nilai *posterior probability* antar kelas, nilai kelas *posterior probability* terbesar akan dipilih sebagai hasil prediksi. *Prior probability* dapat dihitung dengan menggunakan persamaan 3.

$$P(c) = \frac{N_c}{N} \quad (3)$$

N_c merupakan jumlah dari keseluruhan dokumen dengan kelas c, dan N merupakan jumlah keseluruhan dokumen. Untuk $P(t_k|c)$ atau *conditional probability* dapat dihitung dengan menggunakan persamaan berikut dengan menggunakan *laplace smoothing* persamaan 4.

$$P(t_k|c) = \frac{1+N_k}{|V|+N'} \quad (4)$$

Dimana N_k adalah jumlah kemunculan kata t_k dalam dokumen pelatihan di c. N' merupakan jumlah seluruh kata pada dokumen pelatihan di kelas c, dan $|V|$ merupakan jumlah kata unik pada dokumen pelatihan.

Persamaan untuk *Multinomial Naïve Bayes* dengan menggunakan pembobotan TF-IDF dapat lihat pada persamaan 5 [11].

$$P(t_k|c) = \frac{1+W_{ct}}{(\sum W' \epsilon V W' ct) + B'} \quad (5)$$

Dimana W_{ct} adalah nilai pembobotan W (TF-IDF) dari kata t pada kelas c. $\sum W' \epsilon V W' ct$ adalah jumlah total W dari keseluruhan kata yang berada di kelas c. B' merupakan jumlah W kata unik pada seluruh dokumen (nilai IDF tidak dikalikan dengan TF).

2.1. Pengumpulan Data

Pada penelitian ini pengumpulan data dilakukan dengan mengambil komentar ulasan pada kolom review aplikasi SIGNAL di Google Play Store. Pengumpulan komentar akan menggunakan library Python yang bernama google-play-scraper. Proses pengumpulan akan menghasilkan file dengan ekstensi csv. Data yang diambil sebanyak 1792 komentar terbaru yang dilakukan pada 2 April 2022. Data yang diperoleh merupakan data yang diunggah oleh pengguna mulai dari 10 Oktober 2021 sampai 2 April 2022.

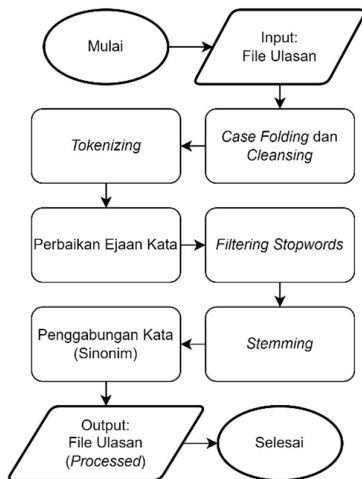
2.2. Pengolahan Data

Terdapat 2 proses yaitu proses pelabelan sentimen secara manual dan proses *text preprocessing*. Proses pelabelan dilakukan untuk memberikan label yang mewakili isi dari suatu komentar. Pelabelan sentimen pada penelitian ini dilakukan secara manual oleh peneliti, dimana ulasan dibagi menjadi tiga kelas yaitu positif, negatif, dan netral. Setelah proses pelabelan manual proses selanjutnya adalah *preprocessing data*. *Preprocessing*

data merupakan serangkaian proses yang digunakan untuk mempersiapkan data mentah (*raw*) menjadi bentuk yang lebih siap untuk digunakan dalam proses *data mining* [12]. Proses ini berguna untuk menghilangkan *noise* atau derau pada dataset dengan harapan hasil yang didapatkan pada proses klasifikasi lebih tepat.

Tahap *preprocessing data* tersusun dari proses-proses yang ditunjukkan pada Gambar 1 terdiri dari:

- Case Folding* dan *Cleansing*, proses mengubah karakter huruf *uppercase* menjadi karakter huruf *lowercase* dan menghapus seluruh karakter selain karakter huruf abjad [13].
Contoh: KEREN BanGET
Menjadi: keren banget
- Tokenizing*, proses pemotongan atau pemenggalan kalimat atau dokumen menjadi potongan-potongan kata atau token [13].
Contoh: keren banget
Menjadi: “keren”, “banget”
- Perbaikan Ejaan Kata, proses untuk memperbaiki kata yang penulisannya tidak sesuai dengan kata baku.
Contoh: mdah dlam dignakan
Menjadi: mudah dalam digunakan
- Filtering*, proses untuk mendapatkan elemen kata yang penting dengan menghilangkan elemen kata yang tidak bermakna [14].
Contoh kata yang tidak bermakna: di, ke, dan, dst.
- Stemming*, proses untuk mengembalikan kata ke bentuk dasar dengan cara menghilangkan semua imbuhan dan akhiran pada kata [14].
Contoh: Pelayanan, bekerja
Menjadi: layan, kerja
- Penggabungan Kata (Sinonim), proses untuk menggabungkan kata yang memiliki makna atau arti yang serupa, dengan tujuan untuk meminimalkan variasi kata yang harus dipelajari oleh sistem. Bekerja dengan cara merubah bentuk bahasa lain ke salah satu bentuk saja.
Contoh: jelek, busuk, cacat, buruk.
Di ubah menjadi: buruk

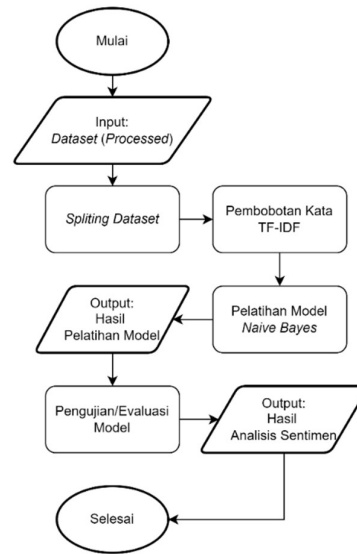


Gambar 1. Tahapan *Preprocessing Data*

2.3. Analisis Sentimen

Dataset komentar yang telah berhasil melalui proses *preprocessing* selanjutnya dibagi menjadi 2 bagian. Pembagian data dengan perbandingan 80% sebagai data latih dan 20% uji. Proses analisis sentimen ditunjukkan pada Gambar 2.

Setelah proses *splitting* selesai, dilanjutkan dengan melakukan perhitungan bobot setiap kata pada dataset. Adapun metode pembobotan yang digunakan adalah TF-IDF atau *Term Frequency–Inverse Document Frequency*. TF-IDF sebuah metode yang digunakan untuk menghitung bobot dari setiap kata dan biasanya diaplikasikan pada kasus *information retrieval*. Metode ini juga adalah metode yang biasa digunakan untuk menentukan seberapa jauh keterhubungan sebuah kata (*term*) dengan dokumen dengan memberikan bobot pada setiap kata [15].



Gambar 2. Proses Analisis Sentimen

Metode ini merupakan hasil dari perkalian metode *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). TF merupakan proses untuk mendapatkan bobot dengan menghitung frekuensi kemunculan sebuah kata pada dokumen tertentu [16]. Rumus TF dapat dilihat pada persamaan 6.

$$tf = tf_{i,d} \tag{6}$$

IDF merupakan metode yang digunakan untuk mengurangi bobot pada term yang memiliki tingkat kemunculan tinggi dan meningkatkan bobot pada term yang memiliki jumlah kemunculan yang rendah [17]. Persamaan IDF dapat dilihat pada persamaan 7.

$$idf_t = \ln \left(\frac{N+1}{df_{t+1}} \right) + 1 \tag{7}$$

Maka untuk mendapatkan bobot TF-IDF suatu kata dapat ditentukan dengan persamaan 8.

$$W_t = tf_{i,d} * idf_t \tag{8}$$

Dimana:

- W_t = Bobot suatu kata
- $tf_{i,d}$ = Nilai *Term Frequency* kata pada sebuah dokumen
- idf_t = Nilai *Inverse Document Frequency* suatu kata

N = Jumlah Dokumen
 df_t = Jumlah Dokumen yang mengandung suatu kata

Setelah proses pembobotan kata, selanjutnya proses pelatihan model berdasarkan teori probabilitas Naïve Bayes. Proses pelatihan ini akan menggunakan data latih, pada data terdapat dua kolom yaitu kolom komentar dan kolom sentimen. Model pembelajaran yang dihasilkan akan dilakukan pengujian dengan menggunakan *confusion matrix* terhadap data uji untuk mengetahui tingkat keakuratan, *precision*, dan *recall* dari sistem analisis sentimen penelitian ini. *Confusion matrix* adalah sebuah matrix dengan ukuran $n \times n$, dimana n merupakan banyaknya target kelas yang diterapkan [18]. Penggunaan *confusion matrix* seringkali digunakan untuk evaluasi sistem yang berisi informasi yang aktual dengan yang diprediksi oleh sistem [19]. Contoh *confusion matrix* untuk *binary class* dapat dilihat pada Tabel 1.

Tabel 1. Confusion Matrix 2 Kelas

Confusion Matrix Binary Class		Actual Class	
		Positif	Negatif
Predicted Class	Positif	TP	FN
	Negatif	FP	TN

TP (True Positif) melambangkan banyaknya data yang berhasil dengan tepat diprediksi sebagai kelas positif, FP (False Positif) melambangkan banyaknya data yang salah diprediksi sebagai data positif, FN (False Negatif) melambangkan banyaknya data yang salah diprediksi sebagai data negatif, dan TN (True Negatif) melambangkan banyaknya data yang berhasil dengan tepat diprediksi sebagai kelas negatif.

Berdasarkan data pada *confusion matrix*, dapat ditentukan tingkat keakuratan, *precision*, dan *recall* dari sistem analisis sentimen. *Accuracy* digunakan untuk menghitung rasio tingkat kedekatan antara nilai yang didapat dengan nilai yang sebenarnya, Precision digunakan untuk menghitung tingkat ketepatan sistem antara bagian data yang yang diambil dengan informasi yang dibutuhkan. *Recall* digunakan untuk menghitung tingkat keberhasilan sistem dalam memperoleh kembali informasi yang relevan [20]. Persamaan *accuracy*, *precision*, dan *recall* dapat dilihat pada persamaan 9, 10, dan 11.

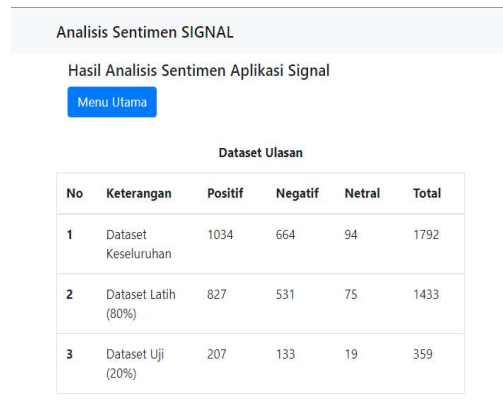
$$Accuracy = \frac{TPos+TNeg}{Total\ data\ uji} * 100 \tag{9}$$

$$Precision = \frac{TPos}{TPos+FPpos} \tag{10}$$

$$Recall = \frac{TPos}{TPos+FNeg} \tag{11}$$

3. HASIL

Komposisi jumlah data pada proses splitting data dengan perbandingan 80:20 dengan total data set sebanyak 1792 data ditunjukkan pada Gambar 3.



Analisis Sentimen SIGNAL					
Hasil Analisis Sentimen Aplikasi Signal					
Dataset Ulasan					
No	Keterangan	Positif	Negatif	Netral	Total
1	Dataset Keseluruhan	1034	664	94	1792
2	Dataset Latih (80%)	827	531	75	1433
3	Dataset Uji (20%)	207	133	19	359

Gambar 3. Komposisi Dataset Penelitian

sebanyak 1433 data untuk melatih model Naïve Bayes dan 359 data digunakan untuk menguji kembali model Naïve Bayes yang dihasilkan.

Pada Tabel 2 merupakan sampel data ulasan dari dataset penelitian ini yang akan digunakan untuk mengetahui cara kerja dari algoritma Multinomial Naïve Bayes. Terdapat 5 data untuk pembelajaran algoritma dan 1 data untuk melakukan prediksi. Data-data tersebut akan dilakukan penghitungan bobot menggunakan TF-IDF untuk mendapatkan bobot masing-masing kata.

Tabel 2. Contoh Data Ulasan

NO	ULASAN	LABEL
1	mantap kali verifikasi tahap salah tolong ribet guna	Negatif
2	verifikasi wajah susah banget	Negatif
3	mantap aplikasi bantu mudah bayar pajak motor depan panjang sim	Positif
4	oke proses lancar hasil cepat terimakasih	Positif
5	bayar pajak via digital hemat biaya transportasi terobos bagus mantap polri	Positif
6.	mantap aplikasi mudah banget bayar pajak kendaraan	?

Pada Tabel 2 merupakan sampel data ulasan dari dataset penelitian ini yang akan digunakan untuk mengetahui cara kerja dari algoritma Multinomial Naïve Bayes. Terdapat 5 data untuk pembelajaran algoritma dan 1 data untuk melakukan prediksi. Data-data tersebut akan dilakukan penghitungan bobot menggunakan TF-IDF untuk mendapatkan bobot masing-masing kata.

Tabel 3. Hasil TF-IDF pada Ulasan 1

KATA	TF	DF	IDF	TF-IDF
mantap	1	3	1.4054651081	1.4054651081
kali	1	1	2.0986122886	2.0986122886
verifikasi	1	2	1.6931471806	1.6931471806
tahap	1	1	2.0986122886	2.0986122886
salah	1	1	2.0986122886	2.0986122886
tolong	1	1	2.0986122886	2.0986122886
ribet	1	1	2.0986122886	2.0986122886
guna	1	1	2.0986122886	2.0986122886

Pada Tabel 3 dapat dilihat untuk bobot setiap kata pada ulasan 1. Untuk mengetahui bobot kata “mantap” pertama kita harus mengetahui nilai TF dan IDF terlebih dahulu. Kata “mantap” muncul sekali pada ulasan 1 maka nilai TFnya bernilai 1. Kata “mantap” muncul pada 3 ulasan (DF). Perhitungan nilai IDF untuk kata “mantap”: menggunakan persamaan 7.

$$idf = \ln\left(\frac{5+1}{3+1}\right) + 1$$

$$idf = 0.40546 + 1 = 1.4056$$

Bobot kata “mantap” dengan menggunakan persamaan 8.

$$W_t = t_{f,d} * idf_t$$

$$W_t = 1 * 1.4056 = 1.4056$$

Proses pembobotan ini juga dilakukan kepada empat ulasan lainnya dan hasil untuk TF-IDF pada ulasan 1 sampai 5 dapat dilihat pada Gambar 4.

KATA	BOBOT (TF-IDF)					Σ BOBOT		IDF
	u1	u2	u3	u4	u5	POSITIF	NEGATIF	
aplikasi	0	0	2,0986	0	0	2,0986	0	2,0986
bagus	0	0	0	0	2,0986	2,0986	0	2,0986
banget	0	2,0986	0	0	0	0	2,0986	2,0986
banu	0	0	2,0986	0	0	2,0986	0	2,0986
bayar	0	0	1,6931	0	1,6931	3,3863	0	1,6931
biaya	0	0	0	0	2,0986	2,0986	0	2,0986
cepat	0	0	0	2,0986	0	2,0986	0	2,0986
depan	0	0	2,0986	0	0	2,0986	0	2,0986
digital	0	0	0	0	0	0	0	2,0986
guna	2,0986	0	0	0	0	0	2,0986	2,0986
hasil	0	0	0	2,0986	0	2,0986	0	2,0986
hemat	0	0	0	0	2,0986	2,0986	0	2,0986
kali	2,0986	0	0	0	0	0	2,0986	2,0986
lancar	0	0	0	2,0986	0	2,0986	0	2,0986
mantap	1,4055	0	1,4055	0	1,4055	2,8109	1,4055	1,4055
motor	0	0	2,0986	0	0	2,0986	0	2,0986
mudah	0	0	2,0986	0	0	2,0986	0	2,0986
oke	0	0	0	2,0986	0	2,0986	0	2,0986
pajak	0	0	1,6931	0	1,6931	3,3863	0	1,6931
panas	0	0	2,0986	0	0	2,0986	0	2,0986
polri	0	0	0	0	2,0986	2,0986	0	2,0986
proses	0	0	0	2,0986	0	2,0986	0	2,0986
ribet	2,0986	0	0	0	0	0	2,0986	2,0986
salah	2,0986	0	0	0	0	0	2,0986	2,0986
sim	0	0	2,0986	0	0	2,0986	0	2,0986
siyah	0	2,0986	0	0	0	0	2,0986	2,0986
tahap	2,0986	0	0	0	0	0	2,0986	2,0986
terimakasih	0	0	0	2,0986	0	2,0986	0	2,0986
terobos	0	0	0	0	2,0986	2,0986	0	2,0986
tolong	2,0986	0	0	0	0	0	2,0986	2,0986
transparansi	0	0	0	0	2,0986	2,0986	0	2,0986
verifikasi	1,6931	1,6931	0	0	0	0	3,3863	1,6931
via	0	0	0	0	2,0986	2,0986	0	2,0986
wajah	0	2,0986	0	0	0	0	2,0986	2,0986
						51,5558	23,6793	69,4433

Gambar 4. Hasil TF-IDF pada Ulasan 1-5

Setelah proses pembobotan sudah dilakukan selanjutnya masuk keproses penghitungan probabilitas *Multinomial Naïve Bayes*. Adapun tahap-tahap dari algoritma ini dalam menentukan label dari ulasan ke-6 ini sebagai berikut:

- Menghitung nilai *prior probability* menggunakan persamaan 3.
 $P(\text{positif}) = 3/5 = 0.6$
 $P(\text{negatif}) = 2/5 = 0.4$
- Menghitung nilai *conditional probability* dari setiap kata pada ulasan 6 menggunakan persamaan 5.
 - $P(\text{mantap}|\text{positif}) = \frac{1+W_{\text{pos,mantap}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+2,8109}{51,5558+69,4433} = 0.03149$
 - $P(\text{mudah}|\text{positif}) = \frac{1+W_{\text{pos,mudah}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+2,0986}{51,5558+69,4433} = 0.02560$
 - $P(\text{banget}|\text{positif}) = \frac{1+W_{\text{pos,banget}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+0}{51,5558+69,4433} = 0.00826$
 - $P(\text{bayar}|\text{positif}) = \frac{1+W_{\text{pos,bayar}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+3,3863}{51,5558+69,4433} = 0.03625$
 - $P(\text{kendara}|\text{positif}) = \frac{1+W_{\text{pos,kendara}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+0}{51,5558+69,4433} = 0.00826$
 - $P(\text{mantap}|\text{negatif}) = \frac{1+W_{\text{neg,mantap}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+1,4055}{23,6792+69,4433} = 0.02583$
 - $P(\text{mudah}|\text{negatif}) = \frac{1+W_{\text{neg,mudah}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+0}{23,6792+69,4433} = 0.01073$
 - $P(\text{banget}|\text{negatif}) = \frac{1+W_{\text{neg,banget}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+2,0986}{23,6792+69,4433} = 0.03327$
 - $P(\text{bayar}|\text{negatif}) = \frac{1+W_{\text{neg,bayar}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+0}{23,6792+69,4433} = 0.01073$
 - $P(\text{kendara}|\text{negatif}) = \frac{1+W_{\text{neg,kendara}}}{(\sum W' \epsilon V W' ct)+B'} = \frac{1+0}{23,6792+69,4433} = 0.01073$
- Menghitung nilai *posterior probability*
Posterior probability didapatkan dengan mengalikan nilai probabilitas *prior* dan *conditional* seperti yang ditunjukkan pada persamaan 2.
 - $P(\text{positif} | \text{ulasan-6}) = P(\text{positif}) * P(\text{mantap} | \text{positif}) * P(\text{mudah} | \text{positif}) * P(\text{banget} | \text{positif}) * P(\text{bayar} | \text{positif}) * P(\text{kendara} | \text{positif})$
 $= 0.6 * 0.03149 * 0.02560 * 0.00826 * 0.03625 * 0.00826$
 $= 1.1963 * 10^{-9}$
 - $P(\text{negatif} | \text{ulasan-6}) = P(\text{negatif}) * P(\text{mantap} | \text{negatif}) * P(\text{mudah} | \text{negatif}) * P(\text{banget} | \text{negatif}) * P(\text{bayar} | \text{negatif}) * P(\text{kendara} | \text{negatif})$
 $= 0.4 * 0.02583 * 0.01073 * 0.03327 * 0.01073 * 0.01073$
 $= 4.2465 * 10^{-10}$
- Membandingkan nilai probabilitas *posterior* untuk menentukan label dari ulasan ke-6.
 Berdasarkan nilai *posterior*, ulasan ke-6 memiliki peluang lebih besar sebagai kelas positif dibandingkan sebagai kelas negatif. Maka ulasan ke-6 akan ditetapkan sebagai ulasan dengan sentimen positif.

		Predicted		
		Positif	Netral	Negatif
Actual	Positif	201	1	5
	Netral	7	7	5
	Negatif	12	0	121

Gambar 5. Confusion Matrix Data Uji

Confusion matrix dari pengujian 359 data ulasan ditunjukkan pada Gambar 4. Sebanyak 201 dari 207 ulasan positif berhasil diprediksi dengan benar sebagai kelas positif. Ada 7 dari 19 ulasan netral berhasil diprediksi dengan benar sebagai kelas netral. 122 dari 133 ulasan negatif berhasil diprediksi dengan benar sebagai kelas negatif.

	Accuracy	Precision	Recall
Positif		91.364%	97.101%
Negatif	91.643%	92.366%	90.977%
Netral		87.500%	36.842%

Gambar 6. Accuracy, precision, dan recall

Setelah mengetahui data pada confusion matrix maka selanjutnya dapat dilakukan penghitungan accuracy, precision, dan recall. Nilai accuracy yang diperoleh sistem sebesar 91.463%, nilai rata-rata precision sebesar 90,41%, dan nilai rata-rata recall sebesar 74.973% ditunjukkan pada Gambar 5.

Setelah mengetahui data pada confusion matrix maka selanjutnya dapat dilakukan penghitungan accuracy, precision, dan recall. Nilai accuracy yang diperoleh sistem sebesar 91.463%, nilai rata-rata precision sebesar 90,41%, dan nilai rata-rata recall sebesar 74.973% ditunjukkan pada Gambar 6.

Pada penelitian ini, penerapan algoritma Multinomial Naïve Bayes menggunakan library scikit-learn. Pelatihan model klasifikasi dilakukan dengan melakukan menggunakan fungsi fit(), sedangkan pengujian model terhadap data uji dilakukan dengan menggunakan fungsi predict(). Fungsi fit() dan predict() pada Scikit-Learn ditunjukkan pada Gambar 7.

Pada MultinomialNB terdapat 2 parameter yang dapat diatur, parameter alpha untuk mengatur nilai smoothing (laplace/lidstone) dimana secara default bernilai 1, dan class_prior yang dapat digunakan untuk menentukan nilai prior probability secara manual. Fungsi fit() akan menghasilkan sebuah model klasifikasi berdasarkan input X,y. Fungsi predict() akan mengklasifikasikan data X menggunakan model yang telah dibentuk pada fungsi fit().

```
class sklearn.naive_bayes.MultinomialNB(*, alpha=1.0, class_prior=None)
classifier = MultinomialNB()
classifier.fit(X,y)
classifier.predict(X)
```

Gambar 7. Fungsi fit() dan predict() pada Scikit-Learn

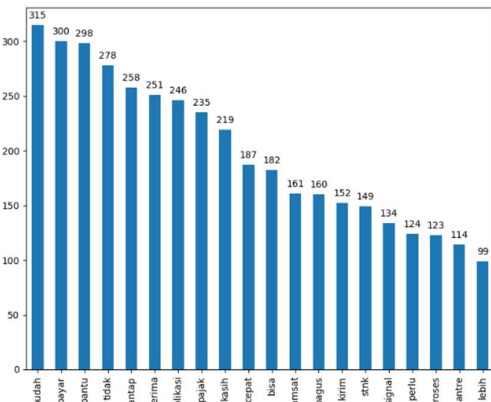
Word cloud digunakan untuk mengetahui kata yang dominan pada sekumpulan dokumen. Dalam penelitian ini, word cloud dibentuk untuk setiap kelas sentimen pada dataset. Dataset ulasan yang sudah melewati tahap preprocessing akan dipisahkan berdasarkan label sentimen. Pada word cloud ditampilkan 50 kata yang memiliki jumlah kemunculan tertinggi.

Word cloud ulasan positif ditunjukkan pada Gambar 8 berdasarkan kata yang sering muncul dapat diketahui informasi yang terdapat pada ulasan dengan sentimen positif adalah aplikasi mudah untuk digunakan dan dapat membantu masyarakat dalam melaksanakan kewajibannya tanpa perlu datang mengantre di Kantor Samsat.



Gambar 8. Word Cloud Ulasan Positif

Jumlah kemunculan dari setiap kata-kata ditampilkan dalam grafik bar ditunjukkan pada Gambar 9 menampilkan 20 kata beserta jumlah kemunculan dari setiap kata-kata tersebut.



Gambar 9. Frequency Kemunculan Kata Pada Ulasan Positif

Kata yang sering muncul pada ulasan pengguna dengan kelas negatif ditunjukkan pada Gambar 10. Berdasarkan kata yang sering muncul tersebut dapat diketahui informasi yang terdapat pada ulasan dengan sentimen negatif adalah beberapa pengguna aplikasi mengalami kendala yaitu tidak bisa mendaftar sebagai pengguna karena proses verifikasi foto ktp yang sulit, proses verifikasi email yang sulit, verifikasi foto wajah yang selalu gagal, dan beberapa dari pengguna masih kedatangan gagal dalam melakukan proses pembayaran



Gambar 10. Word Cloud Ulasan Negatif

Kata yang sering muncul pada ulasan pengguna dengan kelas netral ditunjukkan pada Gambar 11. Berdasarkan kata yang sering muncul tersebut dapat diketahui informasi yang terdapat pada ulasan dengan sentimen netral adalah pengguna cenderung mempertanyakan mengenai hal-hal yang belum diketahui seperti apakah bisa melakukan pembayaran pajak motor bekas yang belum balik nama atau atas nama orang lain, mengenai cara proses pembayaran. ada pula yang menyatakan pada ulasannya dengan ingin mencoba terlebih dahulu.



Gambar 11. Word Cloud Ulasan Netral

4. PEMBAHASAN

Proses *scraping* data dilakukan dengan menggunakan library *google-play-scraper*. Library ini digunakan untuk mengambil data ulasan pengguna aplikasi SIGNAL pada *Google Play Store*. Proses *scraping* data menghasilkan kumpulan data yang terdiri dari nama pengguna, tanggal ulasan, rating bintang, dan ulasan pengguna. Proses *scraping* yang dilakukan pada sistem adalah mengambil ulasan terbaru tanpa filter score rating sebanyak inputan yang diberikan user kepada sistem. *Source code* untuk *scraping* data ulasan ditunjukkan pada Gambar 12.

```

1 from google_play_scraper import reviews, Sort
2 from pandas import DataFrame
3 def scraper_reviews(jumlah):
4     crawled_data, continuation_token = reviews(
5         'app.signal.id',
6         lang = 'id',
7         country = 'ID',
8         sort=Sort.NEWEST,
9         count = jumlah
10    )
11     temp = []
12     for i in range(len(crawled_data)):
13         temp.append([
14             crawled_data[i]['user_name'],
15             crawled_data[i]['at'],
16             crawled_data[i]['score'],
17             crawled_data[i]['content']
18         ])
19     dataframe_ulasan = DataFrame(temp, columns = ['nama_pengguna', 'tanggal', 'skor', 'ulasan'])
20     dataframe_ulasan['label'] = ''
21     return dataframe_ulasan

```

Gambar 12. Source Code Scraping Data Ulasan

Data ulasan yang diperoleh *google-play-scraper* dikirim kembali ke sistem untuk diubah kedalam bentuk file berekstensi csv, contoh file ulasan hasil *scraping* ditunjukkan pada Gambar 13.

Setelah file sudah siap, maka file ulasan diberikan kepada *user* untuk nantinya dapat diberikan label pada dataset data.

Gambar 13. Isi File Ulasan Hasil Scraping

Setelah data ulasan dikumpulkan, tahap selanjutnya adalah memberikan label pada setiap ulasan. Untuk ulasan dengan sentimen positif diberi label 1, sentimen negatif diberi label -1 dan sentimen netral diberi label 0.

Preprocessing data ulasan dimulai dengan memanggil fungsi *caseandclean()* untuk proses *case folding* dan *cleansing*, fungsi *split()* untuk proses *tokenizing*, fungsi *normalisasikata()* untuk proses perbaikan ejaan kata, fungsi *removestopword()* untuk proses *filtering stopwords*, dan fungsi *stemmer.stem()* untuk proses *stemming* kata. *Source code* untuk *scraping* data ulasan ditunjukkan pada Gambar 14

```

1 # Preprocessing
2 for i in data_frame_labeled.index:
3     data_frame_labeled.at[i, 'Case_Folding_dan_Cleansing'] = prepro.caseandclean(
4         data_frame_labeled.at[i, 'ulasan'])
5     data_frame_labeled.at[i, 'Tokenizing'] = data_frame_labeled.at[i, 'Case_Folding_dan_Cleansing'].split()
6     data_frame_labeled.at[i, 'Perbaikan_kata'] = prepro.normalisasikata(data_frame_labeled.at[i, 'Case_Folding_dan_Cleansing'])
7     data_frame_labeled.at[i, 'Filtering'] = prepro.removestopword(data_frame_labeled.at[i, 'Perbaikan_kata'])
8     data_frame_labeled.at[i, 'Stemming'] = prepro.stemmer.stem(data_frame_labeled.at[i, 'Filtering'])
9     data_frame_labeled.at[i, 'Stemming'] = prepro.normalisasikata(data_frame_labeled.at[i, 'Stemming'])
10    data_frame_labeled.at[i, 'Stemming'] = prepro.removestopword(data_frame_labeled.at[i, 'Stemming'])

```

Gambar 14. Source Code Preprocessing Dataset

Dataset yang sudah melalui tahap *preprocessing* selanjutnya akan dilakukan proses *splitting*. Proses *splitting dataset* digunakan untuk membagi dataset menjadi 2 bagian yaitu data uji dan data latih. Proses ini dilakukan dengan memanfaatkan fungsi pada *library sklearn* yaitu *train_test_split*. Fungsi *train_test_split* akan membagi dataset *X* dan label *y* berdasarkan nilai parameter *test_size* yaitu 0.2 atau 20% akan dialokasikan menjadi dataset uji dan sisanya sebagai dataset latih.

Proses pembobotan TF-IDF dilakukan untuk mengubah bentuk data ulasan menjadi bentuk numerik yang bisa diolah oleh sistem. Proses ini dilakukan dengan memanfaatkan class ekstraksi teks pada *library sklearn* yaitu *TfidfVectorizer()*.

Setelah pembobotan dilakukan pada dataset latih dan dataset uji maka variable classifier digunakan untuk menampung model dari algoritma Multinomial Naive Bayes dimana alpha atau nilai *laplace smoothing* yang digunakan adalah 0,05. Dalam proses melatih model, akan digunakan fungsi *fit(X_latih_tfidf, y_latih)* untuk mendapatkan nilai-nilai probabilitas yang dibutuhkan seperti *prior probability* dan *conditional probability*. Fungsi *predict(X_uji_tfidf)* digunakan untuk menguji model klasifikasi yang sudah terbentuk oleh fungsi *fit()* dimana akan mengembalikan *array list* label prediksi.

Dataset ulasan yang sudah dipreprocessing menjadi dataset positif, negatif, dan netral. masing-masing dataset akan dilanjutkan ke proses penghitungan jumlah term dengan menggunakan fungsi `countFrequency()`. Fungsi `countFrequency()` menerima 2 parameter yaitu parameter dataset sentimen dan nama file untuk *word cloud* dan grafik *bar term frequency*.

Pada penelitian sebelumnya kategorisasi sentimen teks dalam kelas positif dan negatif pada 1011 data ulasan aplikasi BRImo dengan nilai akurasi 98,02%. Pada penelitian klasifikasi sentimen pemerintah terhadap penanganan Covid-19 menggunakan 2000 dataset *twitter* melakukan kategorisasi sentimen teks dalam kelas positif, negatif, dan netral dengan Multinomial Naïve Bayes menghasilkan akurasi sebesar 74%.

Analisis sentimen pada penelitian ini dapat melakukan kategorisasi sentimen teks dalam kelas positif dan negatif dan netral terhadap 1792 data ulasan aplikasi SIGNAL menggunakan algoritma Multinomial Naïve Bayes dengan akurasi 91.643%. Kategorisasi sentimen teks dalam kelas positif, negatif dan netral pada ulasan aplikasi SIGNAL menghasilkan akurasi yang lebih baik.

5. KESIMPULAN

Tingkat keakuratan sistem yang didapatkan dari rumus *accuracy* berdasarkan *confusion matrix*, yaitu dengan membandingkan hasil prediksi benar keseluruhan dengan seluruh jumlah data uji diperoleh nilai akurasi sebesar 91.643%. Sistem berhasil memprediksi secara benar 329 data dari 359 data uji.

Hasil *precision* sistem yang didapatkan dari rumus *precision* berdasarkan *confusion matrix*, yaitu dengan membandingkan hasil prediksi benar positif dengan jumlah hasil klasifikasi positif diperoleh nilai untuk *precision* pada kelas positif sebesar 91.364%, *precision* pada kelas positif sebesar 92.366%, *precision* pada kelas netral sebesar 87.5% dan hasil rata-rata *precision* sebesar 90.41%.

Hasil *recall* sistem yang didapatkan dari rumus *recall* berdasarkan *confusion matrix*, yaitu dengan membandingkan hasil prediksi benar positif dengan jumlah keseluruhan data pada kelas positif diperoleh nilai untuk *recall* pada kelas positif sebesar 97.101%, *recall* pada kelas negatif sebesar 90.977%, *recall* pada kelas netral sebesar 36.842% dan rata-rata nilai *recall* sebesar 74.973%. Berdasarkan nilai *recall*, diketahui bahwa sistem kurang dapat memprediksi dengan baik kelas netral. Hal ini dapat disebabkan karena jumlah data netral yang tidak seimbang dengan jumlah data pada kelas lain sehingga kemampuan sistem dalam mengenali ulasan netral buruk dibandingkan dengan ulasan negatif dan ulasan positif.

DAFTAR PUSTAKA

- [1] GooglePlayStore, "SIGNAL – SAMSAT DIGITAL NASIONAL," 2021. <https://play.google.com/store/apps/details?id=app.signal.id&hl=en&gl=US> (accessed Mar. 20, 2022).
- [2] E. P. Nirwandani, Indriati, and R. C. Wihandika, "Analisis Sentimen Pada Ulasan Pengguna Aplikasi Mandiri Online Menggunakan Metode Modified Term Frequency Scheme Dan Naïve Bayes," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 3. pp. 1039–1047, 2021.
- [3] Evasaria Magdalena Sipayung, Bhustomy Hakim, "Analisis Sentimen Kebijakan Pemerintah terhadap Kepercayaan Masyarakat Menggunakan Metode NLP Rule Based", *Jurnal Sistem dan Teknologi Informasi*, Vol. 12, No. 1, Januari 2024.
- [4] Yuyun, Nurul Hidayah, Supriadi Sahibu, "Algoritma Multinomial Naïve Bayes Untuk Klasifikasi Sentimen Pemerintah Terhadap Penanganan Covid-19 Menggunakan Data Twitter", *Jurnal Resti*, Vol 5 No 4 (2021): Agustus 2021.
- [5] Nanda Fibriyanti Arminda, Nina Sulistiyowati, Tesa Nur Padilah, "Implementasi Algoritma Multinomial Naive Bayes Pada Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Brimo", *Jurnal Mahasiswa Teknik Informatika*, Vol. 7 No. 3, Juni 2023.
- [6] Muhammad Abbas, Kamran Ali Memon, Abdul Aleem Jamali, Saleemullah Memon, Anees Ahmed, "Multinomial Naive Bayes Classification Model for Sentiment Analysis", *IJCSNS International Journal of Computer Science and Network Security*, VOL.19 No.3, March 2019.
- [7] prediksi menentukan karyawan tetap pada pt. ysp industries indonesia," Sekolah Tinggi Teknologi Pelita Bangsa, 2018.
- [8] M. Irfan, W. Uriawan, O. T. Kurahman, M. A. Ramdhani, and I. A. Dahlia, "Comparison of Naive Bayes and K-Nearest Neighbor methods to predict divorce issues," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 434, no. 1, 2018, doi: [10.1088/1757-899X/434/1/012047](https://doi.org/10.1088/1757-899X/434/1/012047).
- [9] I. A. Mastan and Y. Toni, "Menggunakan Metode Naive Bayes Classifier Sentiment Analysis Of Chicken Culinary Place From Visitor Comments Uing Naive Bayes Classifier Method," vol. 3, no. 1, pp. 42–50, 2020.
- [10] Y. Rachmanto, "Uji Akurasi Pengkategorian Otomatis Berita Berbahasa Indonesia Dengan Algoritma Multinomial Naive Bayes Classifier (NBC)," pp. 32–38, 2018.
- [11] A. Rahman and A. Doewes, "Online News Classification Using Multinomial Naive Bayes," *ITSMART J. Ilm. Teknol. dan Inf.*, vol. 6, no. 1, pp. 32–38, 2017.
- [12] J. F. Andry, H. Hartono, and A. Chakir, "Data Set Analysis Using Rapid Miner to Predict Cost Insurance Forecast with Data Mining Methods," 2022.
- [13] L. Hermawan and M. Bellanier Ismiati, "Pembelajaran Text Preprocessing berbasis Simulator Untuk Mata Kuliah Information Retrieval," *J. Transform.*, vol. 17, no. 2, p. 188, 2020, doi: [10.26623/transformatika.v17i2.1705](https://doi.org/10.26623/transformatika.v17i2.1705).
- [14] A. P. Thenata, "Text Mining Literature Review on Indonesian Social Media," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 2, p. 226, 2021, doi: [10.26418/jp.v7i2.47975](https://doi.org/10.26418/jp.v7i2.47975).
- [15] Nofenky and D. B. Rarasati, "Recommendation for Classification of News Categories Using Support Vector Machine Algorithm with SVD," *Ultim. J. Tek. Inform.*, vol. 13, no. 2, pp. 72–80, 2021, doi: [10.31937/ti.v13i2.1854](https://doi.org/10.31937/ti.v13i2.1854).
- [16] Wiyanto, W. Priatna, and J. S. Hidayat, "Implementasi Term Frequency – Inverse Document Frequency (TF-IDF) Dan Vector Space Model (VSM) Untuk Pencarian Berita Bahasa Indonesia," *Pelita Teknol. J. Ilm. Inform. Arsit. dan Lingkung.*, vol. 14, pp. 119–133, 2019.
- [17] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Int. J. Comput. Appl.*, vol. 181, no. 1, pp. 25–29, 2018, doi: [10.5120/ijca2018917395](https://doi.org/10.5120/ijca2018917395).
- [18] Z. Karimi, "Confusion Matrix," *Confusion Matrix*, no. October, pp. 260–260, 2021, doi: [10.1007/978-1-4899-7687-1_50](https://doi.org/10.1007/978-1-4899-7687-1_50).
- [19] A. Ridwan, "Penerapan Teknik Bagging Pada Algoritma Naive Bayes Dan Algoritma C4.5 Untuk Mengatasi

- Ketidakeimbangan Kelas,” J. Bisnis Digit. dan Sist. Inf., vol. 1, pp. 63–70, 2020.
- [20] V. Armando, “Sistem Rekomendasi Pembelian Telepon Genggam dengan Metode Content-Based Filtering,” Universitas Atma Jaya Yogyakarta, 2017.

BIODATA PENULIS

Ruby Chandra

Menerima gelar Sarjana Komputer dari Program Studi Informatika Universitas Bunda Mulia pada tahun 2023.

Evasaria Magdalena Sipayung

Menerima gelar Sarjana Teknik dari Sekolah Tinggi Teknologi Telkom Bandung Jurusan Teknik Informasi pada tahun 2003, dan gelar Magister Teknik dari Institut Teknologi Bandung, Sekolah Tinggi Elektro Indonesia (STEI) pilihan Teknologi Informasi pada tahun 2007.