Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0893
p-ISSN: 2089-3787

Deteksi *Fake Review* Menggunakan Metode *Support Vector Machine* dan *Naïve Bayes* Di Tokopedia

Habib Alamsyah[1*], Yana Cahyana[2], Adi Rizky Pratama[3]
Program Studi Teknik Informatika, Universitas Buana Perjuangan Karawang,
Karawang, Indonesia
*e-mail *Corresponding Author:* if19.habibalamsyah@mhs.ubpkarawang.ac.id

Abstract

In the world of online business and services, product and service reviews can have a major influence on user trust and purchasing decisions. However, there is a risk of fake reviews that can affect user trust and purchase decisions. Therefore, detecting fake reviews is very important to avoid fraud and increase user trust. The techniques used in detecting fake reviews are Support vector machine (SVM) and Naïve Bayes. SVM and Naïve Bayes are machine learning algorithms used to classify data into positive and negative categories. In the implementation results using SVM on fake review detection, it appears that SVM and Naïve Bayes can classify reviews into two categories with fairly high accuracy. Through the implementation of SVM and Naïve Bayes, it has been identified that the patterns that are often found in fake reviews are excessive use of words and inconsistent with the actual user experience, so that they can help identify fake reviews more effectively. With the results of the implementation of SVM and Naïve Bayes on fake review detection, several stages in this study used the SVM and Naïve Bayes methods, namely preprocessing, word weighting using TF-IDF, which then implemented the SVM and Naïve Bayes methods. The SVM test results can detect reviews with an accuracy of up to 94.38% and Naïve Bayes produces an accuracy of 91.57%.
Keywords: *support vector machine; naïve bayes; fake review; detection; machine learning.*

Abstrak

Dalam dunia bisnis dan layanan *online*, *review* produk dan layanan dapat memberikan pengaruh yang besar terhadap kepercayaan dan keputusan pembelian pengguna. Namun, terdapat risiko *review* palsu atau *fake review* yang dapat mempengaruhi kepercayaan dan keputusan pembelian pengguna. Oleh karena itu, pendeteksian *fake review* sangat penting dilakukan untuk menghindari penipuan dan meningkatkan kepercayaan pengguna. Teknik yang digunakan dalam pendeteksian *fake review* adalah *Support vector machine* (SVM) dan *Naïve bayes*. SVM dan *Naïve bayes* adalah algoritma *machine learning* yang digunakan untuk mengklasifikasikan data ke dalam kategori positif dan negatif. Dalam hasil implementasi menggunakan SVM pada pendeteksian *fake review*, terlihat bahwa SVM dan *Naïve bayes* dapat mengklasifikasikan *review* ke dalam dua kategori dengan akurasi yang cukup tinggi. Melalui implementasi SVM dan *Naïve bayes*, berhasil teridentifikasi bahwa pola-pola yang sering terdapat pada *fake review* adalah penggunaan kata-kata berlebihan dan tidak konsisten dengan pengalaman pengguna sebenarnya, sehingga dapat membantu dalam mengidentifikasi *review* palsu dengan lebih efektif. Dengan adanya hasil implementasi SVM dan *Naïve bayes* pada pendeteksian *fake review*, Adapun beberapa tahapan dalam penelitian ini menggunakan metode SVM dan *Naïve bayes* yaitu *preprocessing*, pembobotan kata menggunakan TF-IDF, yang selanjutnya implementasi metode SVM dan *Naïve Bayes*. Hasil pengujian SVM dapat mendeteksi *review* dengan akurasi yang mencapai 94,38% serta *Naïve Bayes* menghasilkan akurasi sebesar 91,57%.
Kata kunci*: support vector machine; naïve bayes; fake review;* deteksi*; machine learning.*

1. Pendahuluan

Teknologi di Indonesia saat ini menunjukkan perkembangan yang sangat signifikan. Semakin canggihnya teknologi membuat penggunaan *email*, sosial media, pesan instan hingga perdagangan elektronik semakin mudah dilakukan oleh masyarakat. Aplikasi *e-commerce* juga semakin berkembang di Indonesia dan memudahkan masyarakat untuk membeli barang tanpa harus datang langsung ke toko. Namun, hal ini juga menimbulkan berbagai aspek positif dan negatif ketika semua orang menggunakan fasilitas tersebut dalam hal berbelanja secara daring.

Pada laporan Digital 2020 menunjukkan bahwa Indonesia sebagai negara ke-4 (empat) dengan jumlah penduduk tertinggi di dunia yang memiliki 64% pengguna internet dari total populasi di Indonesia pada tahun 2020. Jumlah tersebut bertambah 17% dibanding tahun 2019 [1]. Berdasarkan data dari [2] pembeli digital di Indonesia berada pada 31,6 juta pengguna pada tahun 2018. Banyaknya jumlah pengguna dan penjual pada *e-commerce* di Indonesia membuat persaingan dalam perdagangan elektronik menjadi sangat luas dan semakin ketat.

Banyaknya produk yang beredar di *e-commerce* dengan berbagai tingkat kualitas yang berbeda-beda, membuat pembeli harus lebih teliti saat memilih produk yang akan dibeli. Ulasan ini memiliki peran penting dalam meningkatkan pengalaman belanja pelanggan baru [3]. Oleh karena itu, sebelum memutuskan untuk membeli suatu produk, biasanya calon pembeli akan melihat terlebih dahulu ulasan atau *review* yang ditinggalkan oleh pembeli sebelumnya.

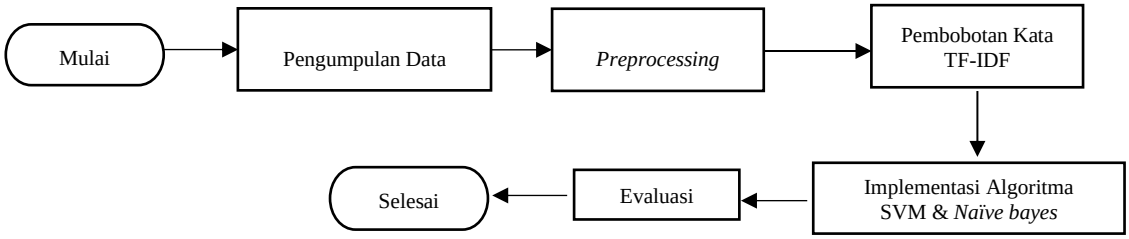
Namun seiring dengan perkembangan *e-commerce* di Indonesia, banyak penjual yang menyalahgunakan dengan menyewa agensi untuk membuat *fake review* atau ulasan palsu pada produk yang mereka jual demi mengembangkan bisnis dan menaikkan reputasi produk mereka. Akibatnya, banyak dari calon pembeli merasa tertipu dengan *review* produk yang tidak sesuai dengan kondisi serta kualitas dari barang tersebut. Ulasan palsu sendiri memberikan hasil sangat efektif dalam jangka pendek bagi penjual karena dalam 2 minggu saja penjual dapat meningkatkan *rating* tokonya dengan rata-rata 0,16 dengan kenaikan berlipat ganda dari 5 sampai 10 ulasan per minggu [4]. Penyebaran informasi yang salah dalam bentuk *fake review* dapat menimbulkan konsekuensi negatif, sangat berbahaya bagi bisnis dan pengguna [5]. Menurut [6] manipulasi ulasan pada awalnya memiliki efek positif pada penjualan, dan kemudian berdampak negatif karena manipulasi menjadi lebih sering dan intensif. Penelitian juga menunjukkan bahwa ketika manipulasi menjadi lebih umum dan konsumen menjadi lebih waspada, ulasan yang mencurigakan dapat memiliki efek buruk yang lebih besar pada persepsi konsumen [7].

Penelitian untuk *fake review* telah dilakukan sebelumnya oleh [8]. Diperoleh hasil akurasi sebesar 74,46%, namun pelabelan yang digunakan apabila keseluruhan *review* berisikan kalimat positif atau kalimat negatif maka dapat dicurigai sebagai *review* palsu, menurut penulis pelabelan tersebut dinilai tidak tepat sasaran untuk dijadikan acuan sebagai ulasan palsu karena banyak dari ulasan positif dan negatif yang memang murni diberikan oleh pembeli sebelumnya, hasil penelitiannya pun akan menjadi ambigu, ketika semua *review* positif dan negatif dicurigai sebagai *fake review*. Selanjutnya, penelitian juga telah dilakukan oleh [9], penelitian dilakukan dengan memunculkan dua fitur baru berlandaskan *word affect intensities* berupa fitur kelompok emosi positif dan fitur kelompok emosi negatif.

Pada penelitian ini, *fake review* dipelajari dalam konteks *review* produk. Pada *review* produk, *fake review*er mungkin menulis ulasan palsu untuk mempromosikan produk tersebut atau menjatuhkan produk pesaing. Proses pelabelan data merupakan salah satu faktor penting untuk menaikkan akurasi saat mendeteksi *fake review*.

Dalam penelitian ini, metode yang digunakan yaitu *Support vector machine* dan *Naïve bayes* dengan teknik *preprocessing Natural Language Processing* (NLP). Setelah itu dilakukan *preprocessing* dan proses pelabelan dilakukan secara otomatis menggunakan teknik *Latent Dirichlet Allocation* (LDA). Selanjutnya model yang telah dilatih, diuji terhadap dataset baru yang belum mempunyai label. Untuk mendeteksi *fake review*, metode ini dapat membantu dalam mengidentifikasi seberapa besar pengaruhnya terhadap pendeteksian *fake review*.

2. Metodologi



Gambar 1. Alur Penelitian.

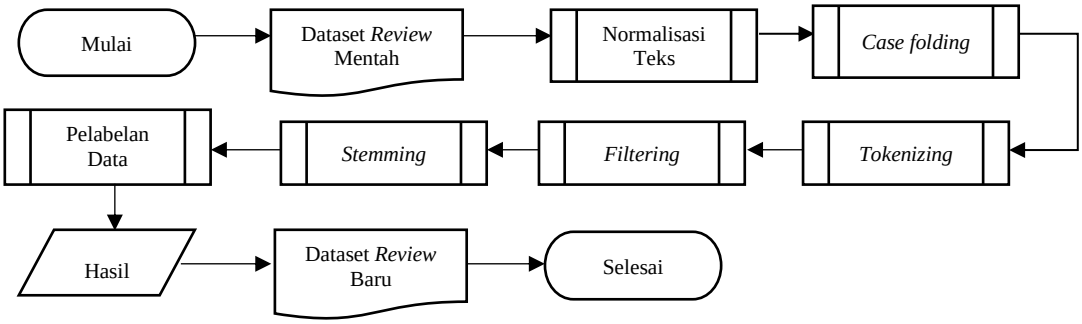
Penelitian ini dilakukan dengan menjalankan beberapa tahapan yaitu pengumpulan data, *preprocessing*, pembobotan kata TF-IDF, implementasi algoritma SVM dan *Naïve bayes*, dan yang terakhir evaluasi. Alur tahapan tersebut dapat ditunjukan pada Gambar 1.

2.1. Pengumpulan Data

Tahap pengumpulan data merupakan tahap yang sangat penting dalam sebuah penelitian, data-data tersebut yang terkumpul merupakan bahan utama yang menjadi inti dari objek penelitian [10]. Data yang telah di *scrape* pada *website* Tokopedia merupakan kumpulan *review* pembeli produk pada toko *online*. Setelah itu data mentah disimpan ke dalam *file* dengan *extension* CSV (*Comma Separated Values*).

2.2. *Preprocessing*

Tahap *preprocessing* merupakan tahapan awal untuk mempersiapkan dokumen agar lebih mudah untuk diproses. Secara umum, tahapan *preprocessing* sebelum proses klasterisasi meliputi *case folding*, *tokenizing*, *filtering*, dan *stemming* [11]. Pada proses pengumpulan data, dataset tersebut *term*asuk ke dalam *unstructured* data (data tidak terstruktur). Sebelum dilakukan permodelan lanjutan, dataset tersebut perlu dilakukan *cleaning* data untuk mengeliminasi *missing value*, menghilangkan *special character*, dan emoji serta mengatasi *noisy* data agar hasil perhitungan optimal. Setelah itu dilakukan *text preprocessing* seperti pada Gambar 2 dan dilakukan pelabelan data menggunakan teknik *Latent Dirichlet Allocation* (LDA). Menurut [12], Text *preprocessing* atau bisa disebut Text Indexing merupakan proses yang digunakan untuk merubah teks menjadi kumpulan kata agar menjadi hal yang bisa diproses oleh komputer. Tahapan text processing ini berguna agar data teks yang memiliki banyak noise atau tidak terstruktur menjadi lebih terstruktur [13]. Menurut [14], ide utama dari algoritma *Latent Dirichlet Allocation* (LDA) adalah dalam sebuah dokumen, terdapat beberapa topik yang saling terkait, dan setiap topik memiliki kata-kata yang berkaitan satu sama lain.



Gambar 2. *Preprocessing*

2.3. Pembobotan Kata TF-IDF

Metode pembobotan kata atau TF-IDF (*Term Frequency-Inverse Document Frequency*) berguna untuk menghitung bobot setiap kata. Cara kerja metode ini dengan menghitung nilai *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) pada setiap token (kata) dalam korpus. Berikut merupakan rumus metode TF-IDF seperti pada persamaan (1).

$$W_{ij} = tf_{ij} + \left(log \frac{D}{df_j}\right) \quad (1)$$

Dimana W_{ij} merupakan bobot *term* terhadap dokumen/*review*. tf_{ij} adalah jumlah kemunculan *term* (tj) dalam dokumen/*review* (di). D adalah jumlah dokumen/*review*. Dan df_j adalah jumlah dokumen/*review* yang mengandung *term* (tj).

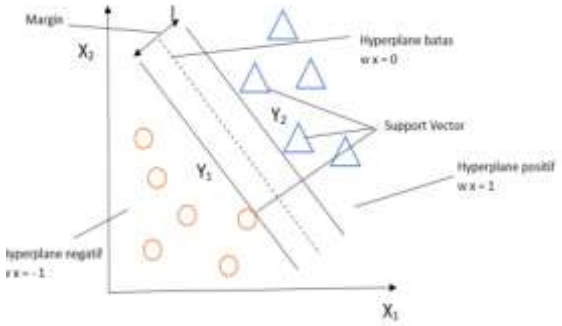
2.4. Implementasi

2.4.1. *Support vector machine*

Support vector machine (SVM) merupakan metode klasifikasi yang diperkenalkan pertama kali oleh Vapnik pada tahun 1998 [15]. *Support vector machine* merupakan algoritma *machine learning* yang menerapkan fungsi *hyperplane* pada data sehingga terbentuk daerah – daerah tiap kelas. *Hyperplane* sendiri merupakan sebuah fungsi yang digunakan sebagai pemisah antar kelas yang ada [16]. Dalam SVM, algoritma memiliki kemampuan untuk mengategorikan pola sebagai positif atau negatif, dan menggunakan batas keputusan yang berbeda untuk membedakannya [17]. Ilustrasi SVM dapat dilihat pada Gambar 3.

Pada prinsipnya algoritma *support vector machine* bekerja sebagai *binary classifier* atau mengklasifikasikan data menjadi 2 kelas (*binary*). Pengklasifikasi biner SVM yang diperoleh dilatih untuk memutuskan apakah kelas tersebut berasal dari kelompok pertama atau milik

kelompok kelas lainnya. Proses ini diulangi untuk grup kedua yang berisi lebih dari dua kelas hingga hanya memiliki satu kelas untuk setiap grup [18].



Gambar 3. Ilustrasi *Hyperplane* SVM.

SVM menggunakan decision *boundary* (batas keputusan) yang akan menentukan klasifikasi dari data-data pelatihan, sehingga dapat dibentuk sebuah model linier atau *hyperplane* yang paling optimal untuk mengkasifikasi data. *Support vector machine* berusaha menemukan pemisah terbaik untuk memisahkan ke dalam dua kelas dan memaksimalkan margin antara dua kelas [19]. Pada banyak kasus, data tidak bisa diklasifikasi menggunakan metode linier *Support vector machine*, sehingga dikembangkanlah fungsi kernel untuk mengklasifikasikan data dalam bentuk *non-linier* [20].

2.4.2 *Naïve Bayes*

Naïve bayes merupakan salah satu metode yang dapat digunakan untuk mengklasifikasikan data. Bayesian classification merupakan pengklasifikasian statistik yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class [21]. Naive Bayes adalah sebuah metode klasifikasi yang didasarkan pada teorema Bayes. Cara kerja *Naive bayes* dengan menghitung sekumpulan probabilitas dengan kombinasi nilai dalam kumpulan data tertentu, biasa digunakan dalam klasifikasi teks, misalnya klasifikasi dokumen dan penyaringan spam. *classifier naive bayes* memiliki proses pengambilan keputusan yang cepat dibandingkan dengan *classifier* lainnya, dan *naive bayes* sering bekerja dengan baik bahkan pada sejumlah kecil data pelatihan [22]. Dalam prosesnya, *Naïve bayes* mengasumsikan bahwa ada atau tidaknya suatu fitur pada suatu kelas tidak berhubungan dengan ada atau tidaknya fitur lain di kelas yang sama [23]. Berikut adalah persamaan *Naïve bayes* seperti pada persamaan (3).

$$P(H|X) = \frac{P(H|X)P(H)}{P(X)} \quad (3)$$

Dimana P(H│X) merupakan peluang dari data sampel X bila diasumsikan bahwa hipotesa benar. P(H) adalah peluang dari hipotesa H. P(X) adalah peluang dari data sampel yang diamati. X adalah sampel data yang memiliki kelas (label) yang tidak diketahui. Dan H merupakan hipotesa bahwa X adalah data kelas (label).

2.5. Evaluasi

Tahap evaluasi dilakukan untuk mengetahui keakuratan dari pemodelan yang telah diterapkan pada data latih. Kemudian membandingkan hasil dari dua dataset yang berbeda dengan menerapkan *confusion matrix* untuk menghitung *precision*, *recall*, *f1-score*, dan *accuracy* [24].

3. Hasil dan Pembahasan

Bab ini merupakan hasil serta pembahasan dari tahapan yang telah dipaparkan pada metode penelitian.

3.1. Hasil Pengumpulan Data

Penelitian ini dilakukan menggunakan dataset sekunder yang telah di *scrape* pada *website* Tokopedia menggunakan *library Selenium* dan *Beautiful Soup* menggunakan bahasa pemograman Python. Proses *scraping* ini dilakukan pada tanggal 4 November 2022 dengan mengambil data sebanyak 887 *review*. Data yang telah di *scrape* merupakan kumpulan *review* pembeli produk pelangsing badan pada toko *online* dengan *username* Herbilogy. Setelah itu data

mentah disimpan ke dalam *file* dengan *extension* CSV (*Comma Separated Values*). Dataset ini terdiri dari beberapa atribut yaitu nama, *rating*, waktu, dan *review*. Sampel hasil dari tahap ini dapat dilihat pada Tabel 1.

Tabel 1. Sampel hasil pengumpulan data

No.	Nama	***Rating***	Waktu	***Review***
1	Eva	4	1 hari lalu	krg cocok.. tidak ada hasil ..bukan jelek krn yg beli byk ..bahan ok alami..produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ad...
2	Dina	5	1 hari lalu	layanan seller ramah, responsif, komunikatif. semoga cocok dan repeat order
3	R***d	5	2 hari lalu	barang bagus, pengiriman cepat.
4	R***i	5	3 hari lalu	Barang sdh sampai dengan selamat, tq ya gan
…	**…**	…	…	…
887	R***d	5	4 hari lalu	biarkan bintang yg berbicara ya gan.

3.2. *Preprocessing*

Setelah tahap pengumpulan data *review*, terlebih dahulu dilakukan *preprocessing*. Hal ini perlu dilakukan karena dataset yang digunakan *term*asuk dalam data yang tidak terstruktur. Berikut adalah beberapa tahapan *preprocessing* yang dilakukan pada penelitian ini:

3.2.1. Normalisasi Teks

Normalisasi teks merupakan tahap untuk merubah singkatan kata menjadi bentuk kata dasarnya. Pada penelitian ini, karena datasetnya merupakan *review* produk dari banyak pengguna, banyak ditemukan singkatan kata yang perlu diubah menjadi bentuk kata dasar. Tahap ini diperlukan untuk tahap *stemming* karena terbatasnya *library* Sastrawi yang tidak bisa merubah singkatan kata menjadi bentuk kata dasar. Hasil dari tahap ini dapat dilihat pada Tabel 2.

Tabel 2. Normalisasi Teks

No.	***Review***	Normalisasi Teks
1	krg cocok.. tidak ada hasil ..bukan jelek krn yg beli byk ..bahan ok alami..produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ad...	kurang cocok.. tidak ada hasil ..bukan jelek karena yang beli banyak ..bahan ok alami..produk lain ditoko ini aku beli bagus cocok cuma yang ini kurang . ternyata tiap orang beda2 ya ada yang cocok ad...
2	layanan seller ramah, responsif, komunikatif. semoga cocok dan repeat order	layanan penjual ramah,responsif,komunikatif. sesemoga cocok dan repeat order
3	barang bagus, pengiriman cepat.	barang bagus, pengiriman cepat.
4	Barang sdh sampai dengan selamat, tq ya gan	Barang sudah sampai dengan selamat, terima kasih ya gan
…	…	…
887	biarkan bintang yg berbicara ya gan.	biarkan bintang yang bicara ya gan..

3.2.2. *Case folding*

Case folding adalah tahap pengubahan semua huruf campuran baik itu huruf kapital atau huruf kecil, menjadi *lowercase* (huruf kecil) semua. Sampel hasil pada tahap ini dapat dilihat pada Tabel 3.

Tabel 3. Sampel Hasil *Case folding*.

No.	Normalisasi Teks	***Case folding***
1	kurang cocok.. tidak ada hasil ..bukan jelek karena yang beli banyak ..bahan ok alami..produk lain ditoko ini aku beli bagus cocok cuma yang ini kurang . ternyata tiap orang beda2 ya ada yang cocok ad...	kurang cocok.. tidak ada hasil ..bukan jelek karena yang beli banyak ..bahan ok alami..produk lain ditoko ini aku beli bagus cocok cuma yang ini kurang . ternyata tiap orang beda2 ya ada yang cocok ad...
2	layanan penjual ramah,responsif,komunikatif. sesemoga cocok dan repeat order	layanan penjual ramah,responsif,komunikatif. sesemoga cocok dan repeat order
3	barang bagus, pengiriman cepat.	barang bagus, pengiriman cepat.
4	Barang sudah sampai dengan selamat, terima kasih ya gan	barang sudah sampai dengan selamat, terima kasih ya gan
…	…	…
887	biarkan bintang yang bicara ya gan..	biarkan bintang yang bicara ya gan..

3.2.3. *Tokenizing*

Tokenizing merupakan proses pemotongan teks menjadi bagian-bagian yang lebih kecil. Pada proses ini juga dilakukan penghilangan angka, tanda baca, dan karakter lain yang dianggap tidak memiliki pengaruh terhadap pemrosesan data. Berikut merupakan sampel hasil dari *tokenizing* pada Tabel 4.

Tabel 4. *Tokenizing*

No.	***Case folding***	***Tokenizing***
1	kurang cocok.. tidak ada hasil ..bukan jelek karena yang beli banyak ..bahan ok alami..produk lain ditoko ini aku beli bagus cocok cuma yang ini kurang . ternyata tiap orang beda2 ya ada yang cocok ad...	['kurang', 'cocok', 'tidak', 'ada', 'hasil', 'bukan', 'jelek', 'karena', 'yang', 'beli', 'banyak', 'bahan', 'ok', 'alami', 'produk', 'lain', 'ditoko', 'ini', 'aku', 'beli', 'bagus', 'cocok', 'cuma', 'yang', 'ini', 'kurang', 'ternyata', 'tiap', 'orang', 'ya', 'ada', 'yang', 'cocok', 'ad']
2	layanan penjual ramah,responsif,komunikatif. sesemoga cocok dan repeat order	['layanan', 'penjual', 'ramah', 'responsif', 'komunikatif', 'sesemoga', 'cocok', 'dan', 'repeat', 'order']
3	barang bagus, pengiriman cepat.	['barang', 'bagus', 'pengiriman', 'cepat']
4	Barang sudah sampai dengan selamat, terima kasih ya gan	['barang', 'sudah', 'sampai', 'dengan', 'selamat', 'terima', 'kasih', 'ya', 'gan']
…	…	…
887	biarkan bintang yang bicara ya gan..	['biarkan', 'bintang', 'yang', 'bicara', 'ya', 'gan']

3.2.4. *Filtering*

Tahapan *filtering* digunakan untuk mengambil kata-kata penting dan menghilangkan kata yang tidak memiliki makna atau biasa disebut *stopword* seperti kata penghubung dan, yang, serta, setelah, dan lainnya. Hasil dari tahap ini dapat dilihat pada Tabel 5.

Tabel 5. *Filtering*.

No.	***Tokenizing***	***Filtering***
1	['kurang', 'cocok', 'tidak', 'ada', 'hasil', 'bukan', 'jelek', 'karena', 'yang', 'beli', 'banyak', 'bahan', 'ok', 'alami', 'produk', 'lain', 'ditoko', 'ini', 'aku', 'beli', 'bagus', 'cocok', 'cuma', 'yang', 'ini', 'kurang', 'ternyata', 'tiap', 'orang', 'ya', 'ada', 'yang', 'cocok', 'ad']	['cocok', 'hasil', 'jelek', 'beli', 'bahan', 'ok', 'alami', 'produk', 'ditoko', 'beli', 'bagus', 'cocok', 'orang', 'ya', 'cocok', 'ad']

2	['layanan', 'penjual', 'ramah', 'responsif', 'komunikatif', 'sesemoga', 'cocok', 'dan', 'repeat', 'order']	['layanan', 'penjual', 'ramah', 'responsif', 'komunikatif', 'sesemoga', 'cocok', 'repeat', 'order']
3	['barang', 'bagus', 'pengiriman', 'cepat']	['barang', 'bagus', 'pengiriman', 'cepat']
4	['barang', 'sudah', 'sampai', 'dengan', 'selamat', 'terima', 'kasih', 'ya', 'gan']	['barang', 'selamat', 'terima', 'kasih', 'ya', 'gan']
…	…	…
887	['biarkan', 'bintang', 'yang', 'bicara', 'ya', 'gan']	['biarkan', 'bintang', 'bicara', 'ya', 'gan']

3.2.5. *Stemming*

Stemming adalah tahapan untuk mengganti kata ke bentuk dasarnya. Contohnya kata memakan, dimakan, makanan akan ditransformasi menjadi kata makan. Pada tahap ini menggunakan algoritma *stemming* Nazief dan Adriani untuk kata berbahasa Indonesia. Berikut merupakan sampel hasil dari *stemming* pada Tabel 6.

Tabel 6. Stemming

No.	*Filtering*	*Stemming*
1	['cocok', 'hasil', 'jelek', 'beli', 'bahan', 'ok', 'alami', 'produk', 'ditoko', 'beli', 'bagus', 'cocok', 'orang', 'ya', 'cocok', 'ad']	['cocok', 'hasil', 'jelek', 'beli', 'bahan', 'ok', 'alami', 'produk', 'toko', 'beli', 'bagus', 'cocok', 'orang', 'ya', 'cocok', 'ad']
2	['layanan', 'penjual', 'ramah', 'responsif', 'komunikatif', 'sesemoga', 'cocok', 'repeat', 'order']	['layan', 'jual', 'ramah', 'responsif', 'komunikatif', 'moga', 'cocok', 'repeat', 'order']
3	['barang', 'bagus', 'pengiriman', 'cepat']	['barang', 'bagus', 'kirim', 'cepat']
4	['barang', 'selamat', 'terima', 'kasih', 'ya', 'gan']	['barang', 'selamat', 'terima', 'kasih', 'ya', 'gan']
…	…	…
887	['biarkan', 'bintang', 'bicara', 'ya', 'gan']	['biar', 'bintang', 'bicara', 'ya', 'gan']

3.2.6. Pelabelan Data

Tahapan pelabelan data merupakan tahap yang sangat penting untuk membuat model, semakin baik proses pelabelan yang dilakukan maka semakin baik juga model yang dihasilkan. Pada penelitian ini pelabelan data dilakukan secara otomatis menggunakan teknik *Latent Dirichlet Allocation* (LDA). LDA memungkinkan identifikasi topik yang tersembunyi dalam dokumen dan memodelkan bagaimana kata-kata dalam dokumen tersebut terkait dengan topik-topik tersebut. Dalam LDA, setiap dokumen dianggap sebagai campuran dari beberapa topik, dan setiap kata dalam dokumen dianggap berasal dari salah satu topik yang dipilih secara acak. Model ini didasarkan pada asumsi bahwa dokumen yang berbeda dapat memiliki pola topik yang berbeda pula. LDA menggunakan pendekatan probabilitas untuk menentukan distribusi topik dalam suatu dokumen dan distribusi kata dalam suatu topik. Hasil dari proses ini terdapat dua kelas yaitu OR (*Original*) sebanyak 547 data dan CG (*Computer generated*) sebanyak 340 data yang terlampir pada Gambar 4.

	length								target							
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
Label																
CG	332.0	89.433735	62.138563	3.0	32.75	72.0	170.0	182.0	332.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0
OR	555.0	56.711712	43.305894	1.0	27.00	44.0	72.0	214.0	555.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Gambar 4. Data Yang Telah Diberi Label Menggunakan Teknik LDA.

Berikut merupakan sampel data yang telah diberi label, terlampir pada Tabel 7.

Tabel 7. Pelabelan Data

No.	Nama	***Rating***	Waktu	Label	***Review***
1	Eva	4	1 hari lalu	CG	krg cocok.. tidak ada hasil ..bukan jelek krn yg beli byk ..bahan ok alami..produk lain ditoko ini aku beli bagus cck cm yg ini krg. tyt tiap org beda2 ya ada yg cck ad...
2	Dina	5	1 hari lalu	CG	layanan seller ramah, responsif, komunikatif. semoga cocok dan repeat order
3	R***d	5	2 hari lalu	CG	barang bagus, pengiriman cepat.
4	R***i	5	3 hari lalu	CG	Barang sdh sampai dengan selamat, tq ya gan
…	…	…	…	…	…
887	R***d	5	4 hari lalu	CG	biarkan bintang yg berbicara ya gan.

3.3. Pembobotan Kata TF-IDF

Pembobotan kata berguna untuk menghitung bobot pada setiap kata dalam *review* agar dapat diolah dan dicari polanya. Proses ini menggunakan metode pembobotan kata atau TF-IDF dengan menganalisa hubungan antara sebuah kalimat dengan sekumpuluan dokumen kemudian melakukan perhitungan berdasarkan persamaan (1). Berikut merupakan hasil perhitungan sampel data *review* menggunakan metode TF-IDF yang terlampir pada Tabel 8.

Tabel 8. Sampel Data Yang di Vektorisasi.

Review	**TF-IDF**					
	Produk	Kirim	Moga	Bantu	Turun	Bb
produk kirim moga bantu turun bb	0.3759	0.3201	0.2264	0.5250	0.3909	0.4793

TF (*Term Frequency*) menghitung seberapa sering sebuah kata muncul dalam dokumen. Dalam penghitungan TF, semakin sering kata muncul dalam dokumen, semakin besar nilai TF-nya. IDF (*Inverse Document Frequency*) menghitung seberapa sering sebuah kata muncul dalam seluruh dokumen yang ada. Dalam penghitungan IDF, semakin jarang kata muncul di seluruh dokumen, semakin besar nilai IDF-nya.

Kombinasi nilai TF dan IDF kemudian digunakan untuk menghitung nilai TF-IDF. Nilai TF-IDF memberikan bobot pada kata dalam suatu dokumen atau korpus teks, yang menunjukkan seberapa penting kata tersebut dalam dokumen atau k

Tabel 9. (a) (b) 10 Kata Dengan Nilai TF-IDF Tertinggi.

Kata	**TF-IDF**
moga	67.6879
cocok	62.2276
coba	55.7684
cepat	46.1213
kirim	44.4928

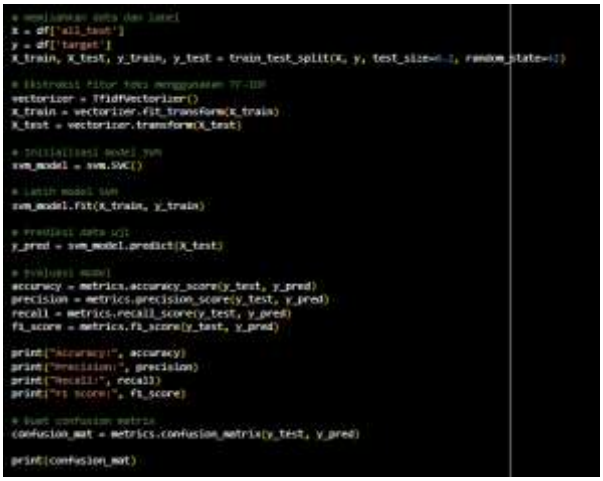
(a)

Kata	**TF-IDF**
puas	37.6573
terima	36.9984
bagus	35.6819
kasih	35.2220
barang	33.9432

(b)

3.4 Hasil Implementasi SVM

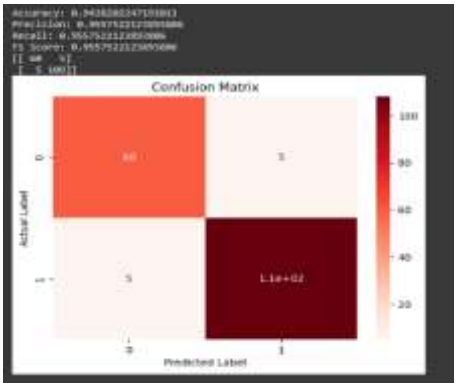
Pada implementasi ini, dataset dipecah menjadi data *training* dan data *testing* dengan perbandingan 80:20 menggunakan fungsi *train_test_split* dari *library* sklearn. Vektor fitur dibuat menggunakan TfidfVectorizer dari *library* sklearn, kemudian model SVM dibuat dengan menggunakan kernel linear dan C=1.0. Model SVM di *training* menggunakan data *training*, lalu dilakukan prediksi nilai label menggunakan data *testing*. Akurasi dan *confusion matrix* dihitung menggunakan fungsi *accuracy*_score dan confusion_matrix dari *library* sklearn, dan hasilnya di print ke layar.



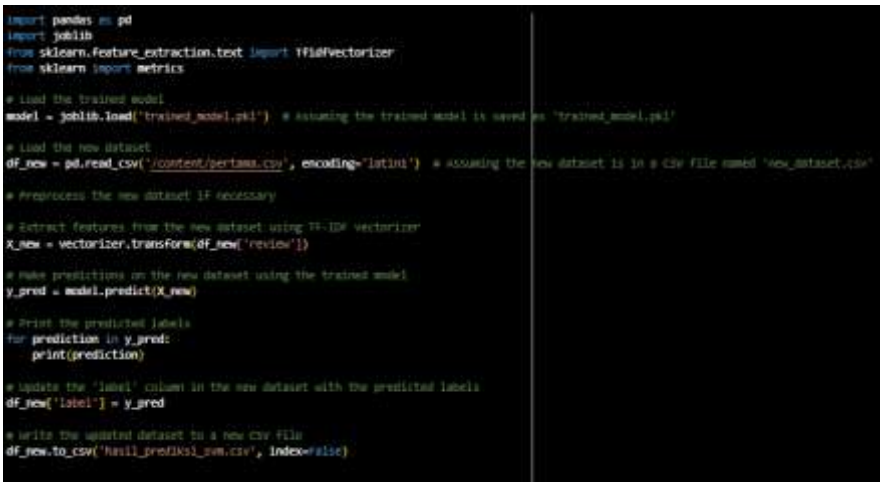
```
# menjalankan data dan label
X = df['all_text']
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# ekstraksi fitur teks menggunakan TF-IDF
vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform(X_train)
X_test = vectorizer.transform(X_test)

# inisialisasi model svm
svm_model = svm.SVC()

# latih model SVM
svm_model.fit(X_train, y_train)

# prediksi data uji
y_pred = svm_model.predict(X_test)

# evaluasi model
accuracy = metrics.accuracy_score(y_test, y_pred)
precision = metrics.precision_score(y_test, y_pred)
recall = metrics.recall_score(y_test, y_pred)
f1_score = metrics.f1_score(y_test, y_pred)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1_score)

# buat confusion matrix
confusion_mat = metrics.confusion_matrix(y_test, y_pred)

print(confusion_mat)
```

Gambar 5. Model SVM

Model SVM pada Gambar 5 digunakan untuk memperoleh hasil akurasi dan nilai *confusion matrix*. Didapatkan hasil akurasi pada metode SVM sebesar 94,38%, lalu *confusion matrix* ditampilkan dalam bentuk tabel dan *heatmap* untuk memudahkan visualisasi dan pemahaman tentang hasil prediksi model. Tabel dan *heatmap* ini dapat membantu untuk mengevaluasi performa model dengan lebih baik, terutama untuk memperoleh informasi tentang jumlah data yang diklasifikasikan secara benar dan salah. Dari Gambar 6 menunjukkan bahwa terdapat 108 data *true positive*, 5 data *false positive*, 60 data *true negative*, dan 5 data *false negative*.



Gambar 6. *Heatmap Confusion matrix* SVM



```
import pandas as pd
import joblib
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn import metrics

# Load the trained model
model = joblib.load('trained_model.pkl')  # Assuming the trained model is saved as 'trained_model.pkl'

# Load the new dataset
df_new = pd.read_csv('/content/pertama.csv', encoding='latin1')  # Assuming the new dataset is in a CSV file named 'new_dataset.csv'

# Preprocess the new dataset if necessary

# Extract features from the new dataset using TF-IDF vectorizer
X_new = vectorizer.transform(df_new['review'])

# Make predictions on the new dataset using the trained model
y_pred = model.predict(X_new)

# Print the predicted labels
for prediction in y_pred:
    print(prediction)

# Update the 'label' column in the new dataset with the predicted labels
df_new['label'] = y_pred

# Write the updated dataset to a new CSV file
df_new.to_csv('hasil_prediksi_svm.csv', index=False)
```

Gambar 7. Model Prediksi Dataset Baru Menggunakan SVM

Setelah dilakukan pemodelan SVM, dilakukan pengujian prediksi model SVM terhadap dataset baru berjumlah 29 data yang tidak mempunyai label. Pertama model yang telah dilatih disimpan

ke dalam file 'trained_model.pkl' menggunakan *library* joblib. Selanjutnya dilakukan pengujian prediksi pada model yang telah dilatih terhadap dataset baru seperti yang terlampir pada Gambar 7.

Pada Gambar 8, memperlihatkan hasil prediksi dengan model SVM yang telah dilatih terhadap dataset baru yang tidak mempunyai label, didapatkan hasil prediksi yaitu 8 data berlabel OR (*Original*) dan 21 data berlabel CG (*Computer Generated*).

label	count	mean	std	min	25%	50%	75%	max
CG	21.0	4.952381	0.218218	4.0	5.0	5.0	5.0	5.0
OR	8.0	4.625000	0.517549	4.0	4.0	5.0	5.0	5.0

Gambar 8. Hasil Prediksi Menggunakan SVM

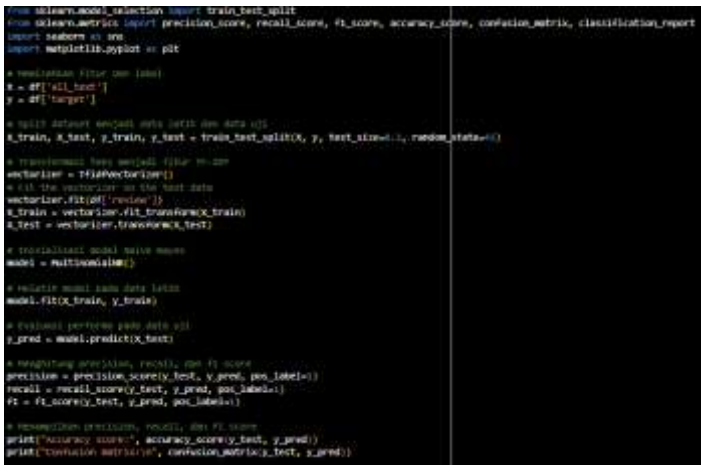
Sampel hasil prediksi dilampirkan pada Tabel 10.

Tabel 10. Hasil Prediksi Dataset Baru

No.	Nama	*Rating*	Waktu	*Review*	Label
1	Liza	5	6 hari lalu	Bagus, hasilnya tdk langsung, tapi bertahap, setelah diminum 5hari x 2 tablet baru BAB lebih lancar tanpa mules, artinya produk ini aman, tdk bermasalah pada maag	CG
2	R***i	5	2 minggu lalu	barang sdh sampai dengan selamat, tq ya gan	OR
3	Asmy	5	3 minggu lalu	puas	OR
4	nitha	4	3 minggu lalu	kurang puas krn tdk ada perubahan dan berat badan saya tidak turun juga	CG
…	…	…	…	…	
29	i***i	5	3 bulan lalu	Sudah pembelian kesekian kalinya. Terima kasih bonusnya.	OR

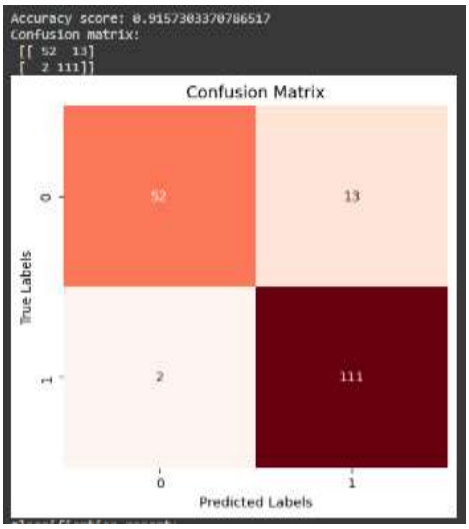
3.6 Hasil Implementasi *Naïve bayes*

Pada implementasi ini, dataset dipecah menjadi data *training* dan data *testing* dengan perbandingan 80:20 menggunakan fungsi *train_test_split* dari *library* sklearn. Vektor fitur dibuat menggunakan *TfidVectorizer* dari *library* sklearn, kemudian model *Naïve bayes* dilatih pada data *training* dan akurasi model dievaluasi pada data *testing.* Selanjutnya nilai akurasi, *recall, precission, f1-score*, dan *confusion matrix* dihitung menggunakan *library* sklearn dan hasilnya di *print* ke layar.



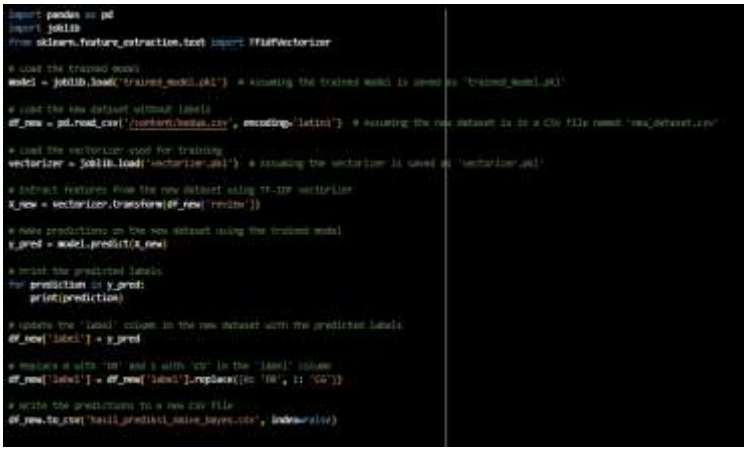
Gambar 9. Model *Naïve bayes*

Model *Naïve bayes* pada Gambar 9 digunakan untuk memperoleh hasil akurasi, *precission, recall, f1-score*, dan *confusion matrix*. Didapatkan hasil akurasi pada metode *Naïve bayes* sebesar 91,57%, lalu *confusion matrix* ditampilkan dalam bentuk tabel dan *heatmap* untuk memudahkan visualisasi dan pemahaman tentang hasil prediksi model. Tabel dan *heatmap* ini dapat membantu untuk mengevaluasi performa model dengan lebih baik, terutama untuk memperoleh informasi tentang jumlah data yang diklasifikasikan secara benar dan salah. Dari Gambar 10 menunjukkan bahwa terdapat 111 data *true positive*, 13 data *false positive*, 52 data *true negative*, dan 2 data *false negative*.



Gambar 10. *Heatmap Confusion matrix Naïve bayes*

Setelah dilakukan pemodelan *Naïve Bayes*, dilakukan pengujian prediksi model *Naïve Bayes* terhadap dataset baru berjumlah 29 data yang tidak mempunyai label. Pertama model yang telah dilatih disimpan ke dalam file '*trained_model.pkl*' menggunakan *library* joblib. Selanjutnya dilakukan pengujian prediksi pada model yang telah dilatih terhadap dataset baru seperti yang terlampir pada Gambar 11.



Gambar 11. Model Prediksi Menggunakan *Naïve Bayes*.

Pada Gambar 12, memperlihatkan hasil prediksi dengan model *Naïve Bayes* yang telah dilatih terhadap dataset baru yang tidak mempunyai label didapatkan hasil prediksi yaitu 22 data berlabel OR (*Original*) dan 7 data berlabel CG (*Computer Generated*).

label	count	mean	std	min	25%	50%	75%	max
CG	22.0	4.954545	0.213201	4.0	5.0	5.0	5.0	5.0
OR	7.0	4.571429	0.534522	4.0	4.0	5.0	5.0	5.0

Gambar 12. Hasil Prediksi Menggunakan Naïve Bayes.

Sampel hasil prediksi dilampirkan pada Tabel 11.

Tabel 11. Hasil Prediksi Dataset Baru

No.	Nama	*Rating*	Waktu	*Review*	Label
1	Liza	5	6 hari lalu	Bagus, hasilnya tdk langsung, tapi bertahap, setelah diminum 5hari x 2 tablet baru BAB lebih lancar tanpa mules, artinya produk ini aman, tdk bermasalah pada maag	CG
2	R***i	5	2 minggu lalu	barang sdh sampai dengan selamat, tq ya gan	OR
3	Asmy	5	3 minggu lalu	puas	OR
4	nitha	4	3 minggu lalu	kurang puas krn tdk ada perubahan dan berat badan saya tidak turun juga	CG
…	…	…	…	…	
29	i***i	5	3 bulan lalu	Sudah pembelian kesekian kalinya. Terima kasih bonusnya.	OR

3.7 Evaluasi

Tahap ini dilakukan untuk mengetahui keakuratan dari pemodelan yang telah diterapkan pada data latih. Kemudian membandingkan hasil dari dua dataset yang berbeda dengan menerapkan *confusion matrix* untuk menghitung *precision*, *recall*, *f1-score*, dan *accuracy*. Hasil implementasi menggunakan algoritma SVM dengan penggunaan data sebanyak 887 data yang terbagi atas 621 data *training* dan 266 data *testing* di evaluasi dengan dihitung nilai *accuracy*, *precision*, *recall*, dan f1-score nya seperti yang terlampir pada Gambar 13. Berikut merupakan persamaan (3) untuk mencari hasil akurasi.

$$Accuracy = \frac{108+60}{108+5+60+5} x\ 100\% = 94{,}38\% \quad (3)$$

Berikut persamaan (4) untuk mencari nilai *precision*.

$$Precission = \frac{108}{108+5} x\ 100\% = 95{,}57\% \quad (4)$$

Berikut persamaan (5) untuk mencari nilai *recall*.

$$Recall = \frac{108}{108+5} x\ 100\% = 95{,}77\% \quad (5)$$

Berikut persamaan (6) untuk mencari nilai *f1-score*.

$$f1 - Score = 2\ x \frac{95{,}57\%\ x\ 95{,}57\%}{95{,}57\% + 95{,}57\%} = 95{,}57\% \quad (6)$$

```
Precision: 0.9557522123893806
Recall: 0.9557522123893806
F1 Score: 0.9557522123893806
```

Gambar 13. Hasil *Precision*, *Recall*, dan *f1-score* SVM.

Sedangkan untuk hasil implemetasi menggunakan algoritma *Naïve bayes* dengan penggunaan data sebanyak 887 data yang terbagi atas 621 data *training* dan 266 data *testing* di evaluasi dengan dihitung nilai *accuracy*, *precision*, *recall*, dan *f1-score* seperti yang terlampir pada Gambar 14. Berikut merupakan persamaan (7) untuk mendapatkan nilai *accuracy*.

$$Accuracy = \frac{111+52}{111+13+52+2} x\ 100\% = 91{,}57\% \quad (7)$$

Berikut persamaan (8) untuk mencari nilai *precision*.

$$Precission = \frac{111}{111+13} x\ 100\% = 89{,}51\% \quad (8)$$

Berikut persamaan (9) untuk mencari nilai *recall*.

$$Recall = \frac{111}{111+2} x\ 100\% = 98{,}23\% \quad (9)$$

Berikut persamaan (10) untuk mencari nilai *f1-score*.

$$f1 - Score = 2\ x\ \frac{89{,}51\%\ x\ 98{,}23\%}{89{,}51\%+98{,}23\%} = 93{,}67\% \quad (10)$$

```
Precision: 0.8951612903225806
Recall: 0.9823008849557522
F1 Score: 0.9367088607594938
```

Gambar 14. Hasil *Precision*, *Recall*, dan *F1-score Naïve bayes*.

Hasil *accuracy, precision, recall,* dan *f1 score* pada kedua metode tersebut memiliki nilai yang hampir sama, ini menunjukkan bahwa model sudah cukup baik secara keseluruhan dapat memberikan performa seimbang antara prediksi yang benar positif (*True Positive*) dan prediksi yang benar negatif (*True Negative*). Hal ini juga menunjukkan bahwa model memiliki tingkat kesalahan yang relatif rendah dan mampu dengan baik mengidentifikasi kedua kelas yang ada (kelas positif dan kelas negatif) pada dataset.

5. Simpulan

Penelitian ini menggunakan metode SVM dan *Naïve bayes* pada *website* Tokopedia. Pada kedua metode tersebut, dataset dilakukan proses *preprocessing* yang selanjutnya data dibagi menjadi dua bagian yaitu 80% data *training* berjumlah 621 data dan 20% data *testing* berjumlah 266 data. Selanjutnya dilakukan pengujian prediksi menggunakan Model SVM dan Naïve Bayes yang telah dilatih terhadap dataset baru yang tidak mempunyai label. Dataset baru tersebut berjumlah 29 data. Terakhir, kedua model dievaluasi pada tahap pengujian menggunakan *confusion matrix*.

Dalam proses pengujian memprediksi dataset baru menggunakan SVM yang telah dilatih, memperoleh hasil prediksi yaitu 8 data berlabel OR (*Original*) dan 21 data berlabel CG (*Computer Generated*). Sedangkan untuk metode Naïve Bayes, memperoleh hasil prediksi yaitu 7 data berlabel OR (*Original*) dan 22 data berlabel CG (*Computer Generated*).

Selanjutnya pada proses evaluasi, metode SVM menghasilkan akurasi sebesar 94,38% dengan nilai precission, nilai *recall*, serta nilai F1 score sebesar 95,57%. Lalu metode *Naïve bayes* menghasilkan akurasi sebesar 91,57% dengan nilai precission 89,51%, nilai *recall* sebesar 98,23%, serta nilai f1 score sebesar 93,67%. Jika dilihat dari hasil akurasi kedua metode tersebut, dapat disimpulkan bahwa metode SVM menghasilkan akurasi yang lebih baik dibandingkan dengan metode *Naïve bayes*.

Pada penelitian ini memiliki keterbatasan dalam jumlah dataset yang hanya berjumlah 887 data, setelah dilakukan *preprocessing*, data yang sudah diberi label berjumlah 547 data berlabel CG (*Computer generated*) dan 340 data dengan label OR (*Original*), ini menjadikan dataset kurang seimbang antara data dengan label CG dan label OR.

Dalam penelitian selanjutnya dapat dilakukan penambahan jumlah dataset agar seimbang, berdasarkan pengujian yang telah dilakukan peneliti pada penelitian ini, memperbanyak jumlah dataset dan menyeimbangkan data dapat meningkatkan hasil akurasi dari pengujian.

Daftar Referensi

[1] V. Stefanny and B. Tiara, "Overview Perbandingan Jumlah User Fintech (Peer-To-Peer Lending) Dengan Jumlah Pengguna Internet Di Indonesia Pada Masa Pandemi Covid-19," *Insa. Pembang. Sist. Inf. dan Komput.*, vol. 9, no. 1, pp. 134–141, 2021, doi: 10.58217/ipsikom.v9i1.194.

[2] Katadata, "Proyeksi Pembeli dan Penetrasi Pembeli Digital Indonesia (2016-2022E)," *katadata.co.id*, 2018. https://databoks.katadata.co.id/datapublish/2018/03/27/berapa-pembeli-digital-indonesia

[3] S. N. Alsubari *et al.*, "Data analytics for the identification of fake reviews using supervised learning," *Comput. Mater. Contin.*, vol. 70, no. 2, pp. 3189–3204, 2022, doi: 10.32604/cmc.2022.019625.

[4] A. Awalina and F. A. Bachtiar, "Klasifikasi Ulasan Palsu Menggunakan Borderline Over-Sampling (Bos) Dan Support Vector Machine (Svm) (Studi Kasus : Ulasan Tempat Makan) Spam Review Classification Using Borderline Over-Sampling and," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 2, pp. 419–426, 2022, doi: 10.25126/jtiik.202295692.

[5] J. Fontanarava, G. Pasi, and M. Viviani, "Feature analysis for fake review detection through supervised classification," *Proc. - 2017 Int. Conf. Data Sci. Adv. Anal. DSAA 2017*, vol. 2018-Janua, pp. 658–666, 2017, doi: 10.1109/DSAA.2017.51.

[6] M. Zhuang, G. Cui, and L. Peng, “Manufactured opinions: The effect of manipulating online product reviews,” *J. Bus. Res.*, vol. 87, no. February 2017, pp. 24–35, 2018, doi: 10.1016/j.jbusres.2018.02.016.
[7] T. Reimer and M. Benkenstein, “When good WOM hurts and bad WOM gains: The effect of untrustworthy online reviews,” *J. Bus. Res.*, vol. 69, no. 12, pp. 5993–6001, 2016, doi: 10.1016/j.jbusres.2016.05.014.
[8] B. E. Pasaribu, A. Herdiani, and W. Astuti, “Deteksi Fake Reviews Menggunakan Support Vector Machine,” vol. 6, no. 2, pp. 8788–8797, 2019.
[9] R. S. H. Istanto, F. A. Bachtiar, and A. Ridok, “Pengaruh Word Affect Intensities Terhadap Deteksi Ulasan Palsu,” *J. Teknol. Inf. dan Ilmu …*, vol. 9, no. 2, pp. 427–434, 2022, doi: 10.25126/jtiik.202295652.
[10] I. Handayani, I. J. Dewanto, and D. Andriani, “Pemanfaatan RinfoForm Sebagai Media Pengumpulan Data Kinerja Dosen,” *Technomedia J.*, vol. 2, no. 2, pp. 14–28, 2018, doi: 10.33050/tmj.v2i2.321.
[11] A. Amalia, M. S. Lydia, S. D. Fadilla, and M. Huda, “Perbandingan Metode Klaster dan Preprocessing Untuk Dokumen Berbahasa Indonesia,” *J. Rekayasa Elektr.*, vol. 14, no. 1, pp. 35–42, 2018, doi: 10.17529/jre.v14i1.9027.
[12] T. Jo, *Text Categorization: Approaches*, vol. 45. 2019. doi: 10.1007/978-3-319-91815-0_6.
[13] A. Syihabudin, A. Juwita, and A. Pratama, “Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Terhadap Produk Motor Matic Honda Beat dan Scoopy,” vol. IV, pp. 95–101, 2023.
[14] P. A. Telnoni, “Pelabelan Data Dengan Latent Dirichlet Allocation dan K-Means Clustering pada Data Twitter Menggunakan Bahasa Indonesia,” *J. Elektro dan Telekomun. Terap.*, vol. 7, no. 2, p. 885, 2021, [Online]. Available: http://journals.telkomuniversity.ac.id/jett/article/view/3442
[15] R. A. Rizal, I. S. Girsang, and S. A. Prasetiyo, “Klasifikasi Wajah Menggunakan Support Vector Machine (SVM),” *REMIK (Riset dan E-Jurnal Manaj. Inform. Komputer)*, vol. 3, no. 2, p. 1, 2019, doi: 10.33395/remik.v3i2.10080.
[16] M. I. Fikri, T. S. Sabrila, and Y. Azhar, “Perbandingan Metode Naïve Bayes dan Support Vector Machine pada Analisis Sentimen Twitter,” *Smatika J.*, vol. 10, no. 02, pp. 71–76, 2020, doi: 10.32664/smatika.v10i02.455.
[17] D. Ismafillah, T. Rohana, and Y. Cahyana, “Implementasi Model Support Vector Machine dan Logistic Regression Untuk Memprediksi Penyakit Stroke,” vol. 10, no. 1, pp. 248–256, 2023, doi: 10.30865/jurikom.v10i1.5478.
[18] H. N. Irmanda and Ria Astriratma, “Klasifikasi Jenis Pantun Dengan Metode Support Vector Machines (SVM),” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 5, pp. 915–922, 2020, doi: 10.29207/resti.v4i5.2313.
[19] L. Mutawalli, M. T. A. Zaen, and W. Bagye, “KLASIFIKASI TEKS SOSIAL MEDIA TWITTER MENGGUNAKAN SUPPORT VECTOR MACHINE (Studi Kasus Penusukan Wiranto),” *J. Inform. dan Rekayasa Elektron.*, vol. 2, no. 2, p. 43, 2019, doi: 10.36595/jire.v2i2.117.
[20] A. Setiyono and H. F. Pardede, “Klasifikasi Sms Spam Menggunakan Support Vector Machine,” *J. Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 275–280, 2019, doi: 10.33480/pilar.v15i2.693.
[21] N. M. Putry, “Komparasi Algoritma Knn Dan Naïve Bayes Untuk Klasifikasi Diagnosis Penyakit Diabetes Mellitus,” *EVOLUSI J. Sains dan Manaj.*, vol. 10, no. 1, 2022, doi: 10.31294/evolusi.v10i1.12514.
[22] A. Sihombing and A. C. . Fong, “Fake Review Detection on Yelp Dataset Using Classification Techniques in Machine Learning,” *IEEE*, pp. 64–68, 2019, doi: https://doi.org/10.1109/IC3I46837.2019.9055644.
[23] Y. Yuliana, P. Paradise, and K. Kusrini, “Sistem Pakar Diagnosa Penyakit Ispa Menggunakan Metode Naive Bayes Classifier Berbasis Web,” *CSRID (Computer Sci. Res. Its Dev. Journal)*, vol. 10, no. 3, p. 127, 2021, doi: 10.22303/csrid.10.3.2018.127-138.
[24] R. Tineges, A. Triayudi, and I. D. Sholihati, “Analisis Sentimen Terhadap Layanan Indihome Berdasarkan Twitter Dengan Metode Klasifikasi Support Vector Machine (SVM),” *J. Media Inform. Budidarma*, vol. 4, no. 3, p. 650, 2020, doi: 10.30865/mib.v4i3.2181.