"BIMASAKTI"
Jurnal Riset Mahasiswa Bidang Teknologi Informasi

Volume 8 Nomor 2
P ISSN : 2355-4401 – EISSN 3109-1652
Available online at https://ejournal.unikama.ac.id/index.php/JFTI

ANALISIS OPINI FILM PADA NETFLIX DENGAN ALGORITMA NAÏVE BAYES DAN SUPPORT VECTOR MACHINE MENGGUNAKAN SELEKSI FITUR CHI-SQUARE

Rahman Riady[1)], Alexius Endy Budianto[2)], Moh Ahsan[3)]

Universitas PGRI Kanjuruhan Malang[1,2,3)]
email : riadyrahman46@gmail.com

Abstrak

Studi ini bermaksud menganalisis opini pengguna terhadap film di platform Netflix menggunakan algorita Naïve Bayes serta Support Vector Machine. Fokus studi adalah meningkatkan akurasi klasifikasi melalui penerapan seleksi fitur mengunakan metode Chi-square. Data yang digunakan diperoleh melalui proses web scraping pada ulasan user di Google Play Store. Labeling secara otomatis dengan bantuan library transformers, dengan hasil label positif 131 serta negatif 869 dari 1000 ulasan. Tahapan penelitian meliputi crawling data, pelabelan otomatis mengunakan library transformers, pre-processing (case folding, tokensiasi, stopword removal, normalisasi, serta stemming), pembobotan dengan metode TF-IDF, serta pengujian akurasi model mengunakan rasio pembagian data 90:10, 80:20 serta 70:30. Hasil penelitian menunjukan bahwa model terbaik yaitu pada metode Support Vector Machine mendapat akurasi 92,5% pada dataset 80:20, sedangkan Support Vector Machine berbasis Chi-square mendapat akurasi 91,5% pada dataset 80:20, Naïve Bayes mendapat akurasi 82% pada dataset 80:20, serta akurasi Naïve Bayes berbasis Chi-square 79% pada dataset 80:20. Maka bosa disimpulkan bahwasanya Chi-square tidak bosa meningkatkan performa dari metode Naïve Bayes serta Support Vector Machine di studi ini.

Kata Kunci : *Analisis Sentimen; Naïve Bayes; Support Vector Machine;Chi-square; Netflix*

Abstract

This research aims to analyse user opinions on films on the Netflix platform using the Naïve Bayes algorithm and Support Vector Machine. The focus of the research is to increase classification accuracy through feature selection using the Chi-square method. The data used is obtained through a web scraping process of user reviews on Google Play Store. Automatic labeling is supported by the Transformers library, resulting in 131 positive labels and 869 negative labels from 1000 reviews. The research stages include data crawling, automatic labeling using the Transformers library, pre-processing (case folding, tokenisation, stopword removal, normalisation, and stemming), weighting with the TF-IDF method, and testing model accuracy using data split ratios of 90:10, 80:20, and 70:30. The findings of the study indicate that the Support Vector Machine algorithm reached an accuracy rate of 92.5% using the 80:20 data split, whereas its Chi-square enhanced variant attained 91.5% accuracy on the same dataset. Meanwhile, the Naïve Bayes classifier recorded an accuracy of 82%, and its Chi-square integrated version yielded 79%. These results suggest that incorporating Chi-square did not enhance the predictive performance of either the Naïve Bayes or Support Vector Machine approaches in this research.

Keywords **:** *Sentiment Analysis; Naïve Bayes; Support Vector Machine; Chi-square; Netflix*

1. PENDAHULUAN

Perkembangan teknologi, khususnya internet, mendorong masyarakat untuk membagikan opini di media sosial seperti Netflix. Industri film kini memanfaatkan aplikasi digital agar lebih mudah diakses. Film, yang sangat digemari, dinilai dari berbagai aspek seperti alur cerita, akting, serta efek visual, serta mengandung pesan moral serta nilai edukatif. Dengan kemajuan teknologi, akses menonton film kini semakin luas melalui berbagai platform seperti bioskop, televisi, aplikasi, serta website. Netflix sebagai layanan streaming digital memungkinkan

pengguna menonton tanpa iklan kapan saja serta di berbagai perangkat, memberikan kebebasan dalam memilih tontonan tanpa harus mengikuti jadwal siaran TV.

Dalam analisis sentimen, ada sejumlah pendekatan yang kerap dimanfaatkan, salah satunya ialah pendekatan Naïve Bayes. Teknik Naïve Bayes merupakan strategi pengelompokan yang lazim diterapkan dalam ranah Penambangan Data serta Penambangan Teks. Strategi Naïve Bayes berlandaskan pada prinsip Bayes yang menyatakan bahwasanya semua fitur punya peran yang setara ataupun tidak saling bergantung dalam menetapkan kategori tertentu. Dalam Penambangan Teks, salah satu strategi pengelompokan yang digunakan untuk mengenali pandangan publik adalah pendekatan Naïve Bayes yang dikenal sebagai Naïve Bayes Classifier [1].

Selain algoritma Naïve Bayes, ada juga algoritma Mesin Vektor Pendukung merupakan metode pembelajaran yang memanfaatkan dugaan berupa persamaan linier pada ruang fitur dimensi yang tinggi serta disusun melalui pendekatan pelatihan yang berlandaskan teori pengoptimalan. Teknik Mesin Vektor Pendukung awal mula diperkenalkan Vapnik di 1992 sebagai integrasi sistematis dari prinsip-prinsip utama dalam bidang pengenalan pola. Tingkat ketepatan dari model yang dihasilkan melalui proses pemrosesan pada metode ini sangat dipengaruhi oleh jenis kernel serta konfigurasi parameter yang diterapkan [2].

Proses data mining, performa suatu algoritma bisa ditingkatkan dengan menggunakan seleksi fitur guna mendapatkan akurasi yang lebih baik serta pengambilan keputusan yang baik pula. Salah satu metode pemilahan atribut yang umum diterapkan ialah chi-square. Chi-square yakni teknik pemilahan atribut yang bertujuan untuk menyingkirkan variabel yang tidak signifikan serta mempercepat analisis.

Pada penelitian yang dilakukan oleh [3] membandingkan kinerja algoritma Naïve Bayes serta Support Vector Machine menggunakan seleksi fiturchi-square. Hasil penelitian menunjukan Naïve Bayes setelah dikombinasikan dengan chi-square berhasil meningkat akurasi sebanyak 3,12%, sedangkan SVM tidak meningkat maupun tidak turun akurasinya setelah dikombinasikan dengan chi-square.

Dari pernyataan diatas maka chi-square dipilih sebegai seleksi fitur dari algoritma Naïve Bayes serta SVM di kajian ini, untuk dilihat apakah mampu meningkatkan akurasi kedua algoritma.

2. METODE / ALGORITMA

Landasan teori dipakai pada studi ini ialah *crawling data*, data mining, analisis sentimen, *text mining, transformer, pre processing, tf-idf, naïve bayes,chi-square, serta confusion matriks.*

2.1. Crawling Data

Crawling ialah suatu teknik menghimpun informasi yang ada di *web*. *Crawling*, juga dikenal sebagai *web crawling* atau *web scraping,* adalah proses otomatis di mana program komputer, yang dikatakan *web crawler* atau *spider,* dengan sistematis menjelajahi halaman *web* untuk menghimpun data. *Web crawler* biasanya dimulai dengan *URL* awal atau daftar *URL* yang ditentukan. Kemudian, mereka mengikuti tautan di halaman tersebut untuk mencapai halaman-halaman lain yang terhubung, serta seterusnya. Proses ini berlanjut secara terus-menerus hingga semua halaman yang relevan telah dikunjungi [4].

2.2. Data Mining

Data mining ialah rangkaian pengekstrakan informasi yang berguna atau wawasan yang tersembunyi dari kumpulan data besar. Metode serta teknik data mining dipakai untuk mengidentifikasi pola-pola yang tersembunyikan, hubungan, serta tren yang relevan dalam data yang ada. Tujuannya adalah untuk merubah data yang mentah jadi informasi yang bermakna serta berguna bagi pengguna. Proses data mining melibatkan beberapa tahap, termasuk pemahaman terhadap tujuan bisnis, pemilihan data yang relevan, *pre-processing* data untuk membersihkan serta mengintegrasikan data, serta penerapan algoritma serta teknik data mining untuk mengungkap pola serta tren dalam data tersebut [5].

2.3. Analisis Sentimen

Yakni rangkaian yang melibatkan pemahaman serta pemrosesan data tekstual agar mendapatkan informasi mengenai opini atau sentimen yang ada di dalamnya. Tujuan inti analisis ini ialah mendeteksi serta memahami opini ataupun pandangan orang kepada sebuah subjek atau pun obyek tertentu, semacam individu, komunitas, ataupun produk. [6].

2.4. Text Mining

Text Mining ialah tahap rangkaian dalam analisis data di mana sumber data bersumber dari dokumen teks semacam kalimat, kata-kata, atau teks panjang lainnya. Teknik *text mining* digunakan untuk menggali informasi berharga dari teks yang besar serta kompleks. Salah satu aplikasi umum *text mining* adalah klasifikasi dokumen tekstual. Pada tahap ini, dokumen-dokumen tersebut diklasifikasikan ke dalam kategori atau topik tertentu berdasarkan isinya. Misalnya, dalam sebuah dataset dokumen berita, *text mining* dapat digunakan untuk mengklasifikasikan setiap berita ke dalam topik-topik seperti politik, olahraga, bisnis, hiburan, serta lain sebagainya. Dalam proses *text mining*, teknik seperti pemrosesan bahasa yang alami *(natural language processing/NLP)*, pemodelan topik, analisis sentimen, serta metode *machine learning* dapat digunakan untuk mengolah serta menganalisis data teks. Tujuannya adalah untuk mendapatkan wawasan yang berharga, mengidentifikasi pola atau tren, serta memahami konteks dari dokumen teks tersebut [1].

2.5. Transformer

Arsitektur Transformer adalah pendekatan yang menggeser fokus serta relasi menyeluruh antara masukan serta keluaran, menggantikan struktur yang lazim dipakai dalam pengkode serta pengurai konvensional. Model Transformer mempersembahkan inovasi baru dalam kualitas terjemahan, sambil memungkinkan pelatihan yang jauh lebih efisien dibandingkan dengan arsitektur yang bergantung pada penggunaan lapisan berulang atau konvolusi.Pelabelan *Transformer* adalah proses pemberian label atau penanda pada setiap elemen dalam sebuah rangkaian atau urutan menggunakan model *Transformer*. Model *Transformer* adalah salah Suatu bentuk struktur komputasi cerdas yang amat efisien dalam menangani rangkaian informasi, khususnya dalam aktivitas pengolahan bahasa manusia (natural language processing/NLP). Dalam konteks pelabelan *Transformer*, rangkaian atau urutan data dapat berupa kata-kata dalam sebuah kalimat, token pada teks, ataupun segmen dalam dokumen. Maksud dari pelabelan *Transformer* ialah untuk menghasilkan label yang tepat untuk setiap elemen dalam urutan tersebut [7].

2.6. Pre-processing

Merupakan proses pada analisis data yang terdiri dari rangkaian tahap membersihkan, mengubah, serta menyiapkan data mentah sebelum dipakai pada analisis lanjutan ataupun pembuatan model. Makusd dari *pre-processing* ialah untuk melakukan pemastian data pada format yang sesuai, memperbaiki ataupun melakukan penghilangan data yang tidak valid ataupun tidak sesuai, serta membuat data siap untuk diproses oleh algoritma analisis atau pemodelan yang akan digunakan. Beberapa tahapan pada *pre-processing* data adalah *case*

folding, tokenizingserta frekuensi token, stopword removal, word normalization, serta stemming [8]. Berikut penjelasan dari tahapan-tahapan dalam *pre-processing.*

Tabel 1. Tahapan pre-processing

No	Tahapan	Tujuan
1	*Case Folding*	Mengubah semua kata jadi *lower case*, menghapus url, tanda baca, serta emoji.
2	*Tokenizing,* *Frekuensi Token*	Memisahkan kata menjadi pertoken. Menelusuri frekuensi istilah dalam berkas tertulis.
3	*Stopword Removal*	Membuang istilah yang kurang bernilai semantik
4	*Word Normalization*	Menyusun ulang bentuk ringkas serta gaul ke versi formal
5	*Stemming*	Menyesuaikan istilah ke bentuk leksikal utama.

2.7. TF-IDF

Tf-Idf (Frekuensi Istilah–Kebalikan Frekuensi Dokumen) merupakan pendekatan yang dimanfaatkan pada pengolahan bahasa manusia serta penelusuran informasi untuk mengukur tingkat signifikansi suatu istilah atau kata dalam satu naskah di antara kumpulan naskah yang lebih luas. Tf (Frekuensi Istilah) adalah penilaian terhadap seberapa kerap suatu kata muncul dalam satu naskah. Nilai tf meningkat apabila kata tersebut muncul lebih banyak dalam naskah tersebut. Tujuannya adalah untuk menetapkan bobot yang lebih besar bagi kata yang sering muncul karena kemungkinan besar kata-kata tersebut memiliki keterkaitan yang lebih kuat dengan naskah tersebut [9].

2.8. Naïve Bayes

Naïve Bayes ialah metode pengklasifikasi yang dilandaskan kepada Teorema Bayes serta memakai probabilitas sederhana. Metode ini memiliki asumsi bahwasanya setiap fitur ataupun kondisi pada data yang diamati saling independen satu dengan lain. Asumsi ini sering disebut sebagai "naïve" karena dalam dunia nyata, fitur-fitur tidak selalu independen. Meski asumsi independensi ini seringkali tidak benar secara absolut, *Naïve Bayes* tetap menjadi metode yang terkenal serta efektif dalam banyak kasus. Keuntungan utama dari *Naïve Bayes* ialah kecepatan serta kinerja yang bagys untuk melakukan pengklasifikasian data.

Dalam pengklasifikasi menggunakan *Naïve Bayes*, probabilitas posterior diperoleh dengan menkalikan probabilitas prior serta likelihood. Probabilitas prior adalah probabilitas sebelum melihat data, sedangkan likelihood adalah probabilitas data mengingat kelas atau label yang ada. Kemudian, Kategori dengan kemungkinan akhir terbesar akan ditetapkan sebagai hasil tebakan. Walaupun asumsi independensi yang tinggi sering tak terpenuhi dalam data fakta, *Naïve Bayes* masih bisa memberi hasil yang baik pada banyak kasus, terutama ketika jumlah fitur cukup besar serta memiliki kompleksitas yang rendah. Dalam banyak aplikasi, *Naïve Bayes* sering dipakai pada pemrosesan teks, pengenalan pola, serta sistem rekomendasi [6].

Persamaan *teorema bayes* sebagai berikut :

$$P\ (H \mid X) = \frac{P(H \mid X).P(H)}{P(H)} \ldots \qquad (1)$$

Dimana :

X = Kategori dengan informasi yang belum dikenali
H = Hipotesis informasi pada suatu kategori tertentu
P(H|X)= Probabilitas pada hipotesis H menurut keadaan pada X
P(H) = Probabilitas pada hipotesis H
P(X|H)= Probabilitas pada X menurut keadaan dugaan H
P(X) = Probabilitas pada X

2.9. Support Vector Machine

SVM yaitu algoritma klasifikasi yang ialah bagian dari jenis pembelajaran terawasi. Mencari garis batasan ataupun *hyperplane* terbaik buat memisahkan 2 kelas merupakan bawah dari konsep SVM Gambar ini menggambarkan gimana SVM beroperasi. Perhitungan *margin hyperplane* serta penentuan titik maksimum digunakan buat memastikan *hyperplane* yang sangat maksimal buat membagi informasi jadi 2 kelas. Suatu persamaan bisa digunakan buat mendapatkan *hyperplane* dalam SVM. Pendekatan SVM membagi dataset menjadi dua kelas, dengan variabel xi serta yi masing-masing ditugaskan pada data serta kelas. Nilai kelas yang dipisahkan oleh *hyperplane* adalah 1, serta nilai kelas lainnya adalah -1.

$(w.x_i) + b = 0$ (2)
$(w.x_i + b) \leq 1, y_i = -1$
$(w.x_i + b) \geq 1, y_i = 1$
Ket :
x_i =yaitu data -i
w =yaitu nilai bobot suport victor tegak lurus dengan hyperplane
b = yaitu nilai biasa
y_i =yaitu kelas data ke -i

2.10. Chi-Square

Chi-Square ialah sebuah statistik uji yang dipakai dalam analisis statistik untuk menentukan apa ada korelasi antara 2 variable dengan skala nominal ataupun ordinal. Statistik ini didasarkan di perhitungan perbedaan antar frekuensi yang diobservasi serta frekuensi yang dikehendaki dalam tabel kontingensi. Secara umum, Metode Chi-Square dimanfaatkan untuk menilai apa ada selisih yang berarti antara sebaran teramati (observasi aktual) serta sebaran yang diperkirakan (dalam kondisi tanpa keterkaitan antar parameter). Teknik Chi-Square mampu menyajikan indikasi mengenai ada tidaknya relasi antara dua parameter tersebut adalah kebetulan atau terdapat hubungan yang nyata [10].

Rumus pengujian pada *Chi-Square* sebagai berikut :

$$x^2 = \left[\frac{\Sigma(f_0 - f_e)^2}{f_e}\right] \quad (3)$$

dimana :
x^2 = Nilai dari *chi-square*
f_o = Frekuensi yang diinginkan
f_e = Frekuensi yang didapatkan

2.11. Confusion Matriks

Adalah sebuah tabel yang mencatatkan hasil dari proses klasifikasi. Untuk menilai kinerja model yang telah dibuat, evaluasi dilakukan dengan mengukur performanya, serta salah satu

parameter evaluasi yang digunakan adalah akurasi, yang dihitung berdasarkan informasi yang terdapat dalam *confusion matriks*. [11].

Berikut persamaannya*:*

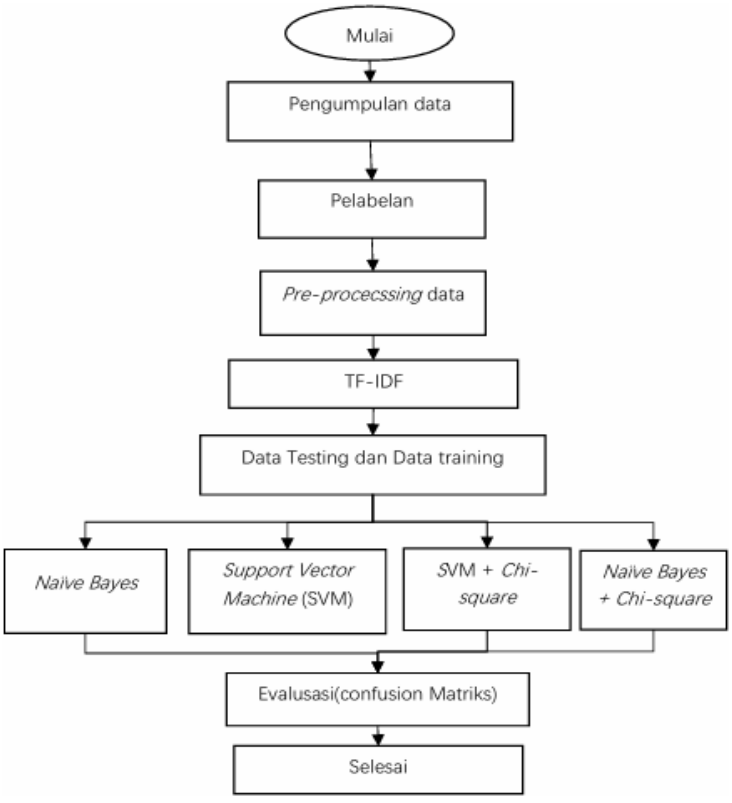
Tabel 2. Persamaan confusion matriks

		Prediksi	
		Positif	Negatif
Aktual	Positif	PB	NB
	Negatif	PS	NS

Keterangan :
PB : Prediksi benar aktual benar
NB : Prediksi salah aktual benar
PS : Prediksi benar aktual salah
NS : Prediksi salah aktual salah

$$Confusion\ Matriks : \frac{PB+NS}{PB+NB+PS+NS}\ X\ 100\% \qquad (4)$$

Berikut rancangan yang dipakai dalam penelitian ini.



Gambar 1. Rancangan Penelitian

Dari gambar sebelumnya bisa diketahuo rancangan pada kajian terbagi atas penghimpunan data/*crawling data*, labeling, *pre-processing, tf-idf*, *data testing serta testing,* klasifikasi, serta *evaluasi.*

3. HASIL DAN PEMBAHASAN

3.1. Crawling Data

Dalam studi ini, data dipakai adalah data yang ddidapay dari ulasan/komentar user aplikasi Netflix. Pengumpulan data menggunakan teknik *scrapping* di *Google Play Store*. Adapun tahapan yang dilaksanakan adalah *install google play scraper* pada *python*, lalu memasukan id aplikasi yang dituju, dalam hal ini *'com.netflix.mediaclient'*. Data yang diperoleh berjumlah 1000 data yang diambil pada tanggal 8 juli 2024 dengan hanya mengambil data pada atribut *content* serta *score*.

Adapun data yang berhasil diperoleh dari tahapan diatas terlihat pada tabel berikut. Data yang diambil pada penelitian ini menggunakan rating dari rentang 1-5. Rating 1-2 bisa menunjukan ketidakpuasan, sedangkan rating 3-5 menunjukan kepuasan, akan tetapi dalam pemberian label tetap merujuk pada hasil dari library transformers.

Tabel 3. Data Hasil Scrapping

score	*Content*
5	Selama ini ok
1	Susah banget loginnya
1	Baru aja mau login ke aplikasi nya malah di suruh bayar
5	apk nonton yang bgusss

3.2. Labeling

Dalam penelitian ini dalam melakukan pelabelan secara otomatis menggunakan bantuan dari *library transformers*, dengan hasil ulasan positif serta ulasan negatif. Dalam tahapannya ulasan akan dibaca oleh sistem, lalu diterjemahkan kedalam bahasa inggris agar dikenali oleh sistem, barulah *library transformers* akan memberikan label kepada ulasan. Setelah berhasil memberikan label pada ulasan, semua ulasan yang berlabel positif akan di replace dengan angka 0 serta berlabel negatif di replace dengan angka 1. Berikut hasil labeling secara otomatis menggunakan bantuan *library transformers.*

Tabel 4. Data Hasil Labeling

Content	***Label***
Selama ini ok	0
Susah banget loginnya	1
Baru aja mau login ke aplikasi nya malah di suruh bayar	1
apk nonton yang bgusss	0

Dari hasil labeling diatas menggunakan bantuan *library transformers* mendapatkan ulasan berlabel positif berjumlah 131 serta ulasan berlabel negatif berjumlah 869.

3.3. Pre-processing Data

Ulasan pengguna pada aplikasi netflix sudah melalui beberapa tahapan yakni *labeling* otomatis menggunakan bantuan *library transformers* serta *replace values* pada atribut label. Berikutnya masuk ketahap *pre-processing* data yang tujuannya agar dapat data yang lebih bersih serta baik supaya masuk ketahapan berikutnya. Adapun tahapan ini terdiri dari *"case folding, tokenisasi serta frekuensi token, stopword removel*, normalisasi kata*,* serta *stemming".*

Tabel 5. Data Hasil Pre-processing

Tahapan	Data
	Baru aja mau login ke aplikasi nya malah di suruh bayar
Case Folding	baru aja mau login ke aplikasi nya malah di suruh bayar
Tokenizing, Frekuensi Token	['baru', 'aja', 'mau', 'login', 'ke', 'aplikasi', 'nya', 'malah', 'di', 'suruh', 'bayar'] {'baru': 1, 'aja': 1, 'mau': 1, 'login': 1, 'k...
Stopword Removal	['login', 'aplikasi', 'suruh', 'bayar']
Word Normalization	['login', 'aplikasi', 'suruh', 'bayar']
Stemming	['login', 'aplikasi', 'suruh', 'bayar']

3.4. TF-IDF

Sesudah lewat tahapan *pre- procesing* berikutnya dicoba sesi membobotkan kata, tahapan ini bertujuan buat menghitung bobot nilai dari frekuensi kata didalam dokumen serta pula bobot nilai dari frekuensi kata dalam banyak. Pembobotan ini buat memandang sepanjang mana tingkatan relevan suatu kata didalam dokumen.

Setelah melalui tahapan pembobotan *tf-idf*, tercantum dibawah ini.

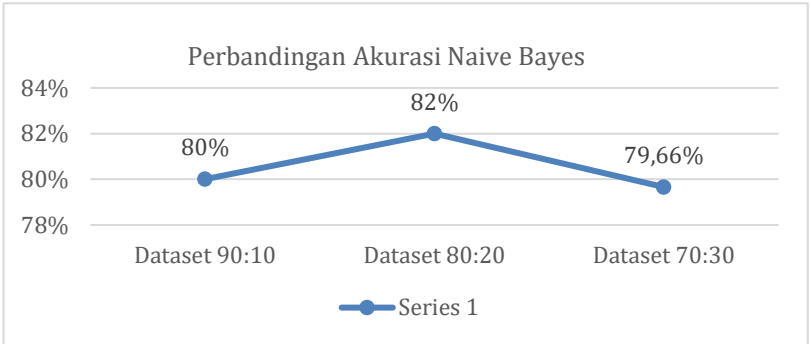
Tabel 6. Hasil tf-idf

	abdi	abis	abissitu	about	...	Yokk
0	0.0	0.0	0.0	0.0	...	0.0
1	0.0	0.0	0.0	0.0	...	0.0
2	0.0	0.0	0.0	0.0	...	0.0
3	0.0	0.0	0.0	0.0	...	0.0
...	...	...	...	...	...	...
999	0.0	0.0	0.0	0.0	...	0.0

3.5. Klasifikasi Naïve Bayes

Setelah melalui serangkaian tahapan mulai dari tahap *pre- processing* serta pembobotan menggunakan *tf-idf*, berikutnya masuk ketahap pengujian memakai algoritma *Naïve Bayes*. *Library python* serta dipakai kedalam tahapan ini adalah *GausianNB* dari *sklearn* serta *accuracy score* untuk melihat akurasi yang didapatkan oleh model. Pengujian dilakukan menggunakan 3 rasio dataset. Berikut ialah uji algoritma *Naïve Bayes*.

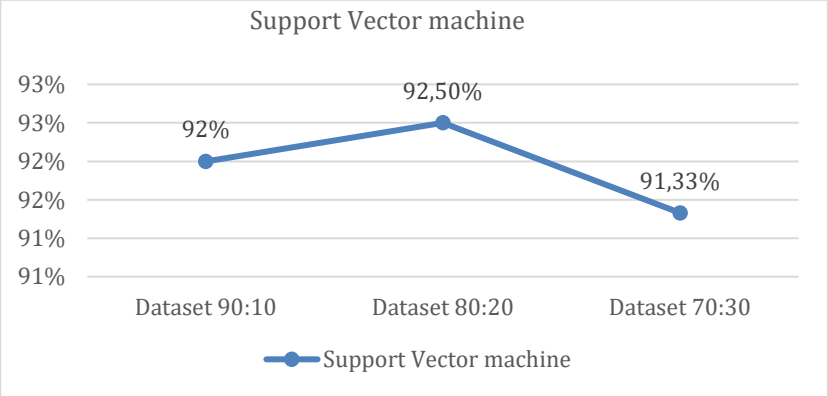
Dari gambar sebelumnya bisa diambil kesimpulan bahwasanya bahwasanya akurasi terbaik dari pengujian memakai algoritma *Naïve Bayes* didapatkan pada *rasio dataset* 80:20 dengan akurasi sebesar 82%.



Gambar 2. Hasil Akurasi Naïve Bayes

3.6. Klasifikasi Support Vector Machine

Setelah lewat rangkaian langkah mulai dari *pre-processing* serta pembobotan menggunakan *tf-idf*, berikutnya masuk ketahap pengujian menggunakan algoritma SVM. *Library python* yang dipakai dalam tahapan ini adalah *svm.SVC* dari *sklearn* serta *accuracy score* untuk melihat akurasi yang didapatkan oleh model. Pengujian dilakukan menggunakan 3 rasio dataset. Berikut adalah hasil pengujian algoritma SVM.

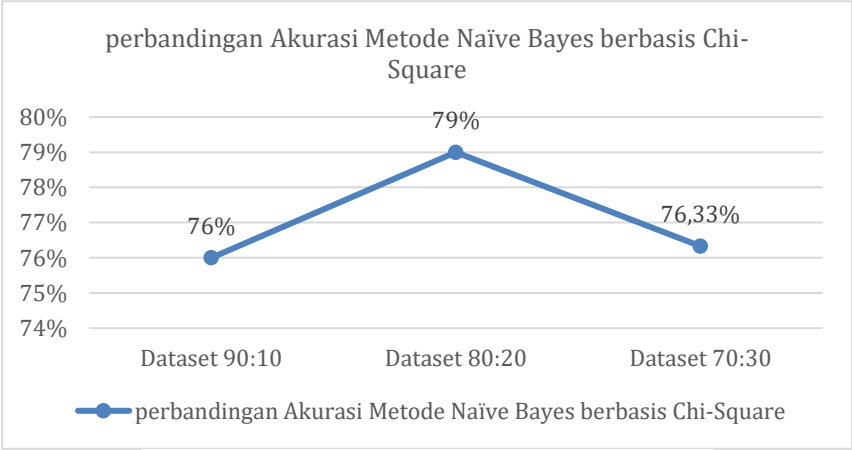


Gambar 3. Hasil Akurasi SVM

Dari gambar sebelumnya bisa diambil kesimpulan bahwasanya akurasi terbaik dari uji menggunakan algoritma *SVM* didapatkan pada *rasio dataset* 80:20 dengan akurasi sebesar 92,50%.

3.7. Klasifikasi Naïve Bayes berbasis Chi-Square

Di pengujian ini ialah mempraktikkan seleksi fitur *chi-square* kedalam algoritma *Naïve Bayes*. Seleksi fitur dipakai buat kurangi atribut ataupun fitur yang tidak relevan adalam dataset serta memacu pemrosesan informasi, *chi- square* dipilih sebab sanggup mengukur ikatan antara variabel nominal serta variabel nominal yang ada serta tingkatkan akurasi prediksi ataupun kinerja dataset dengan memilah fitur yang sangat relevan.

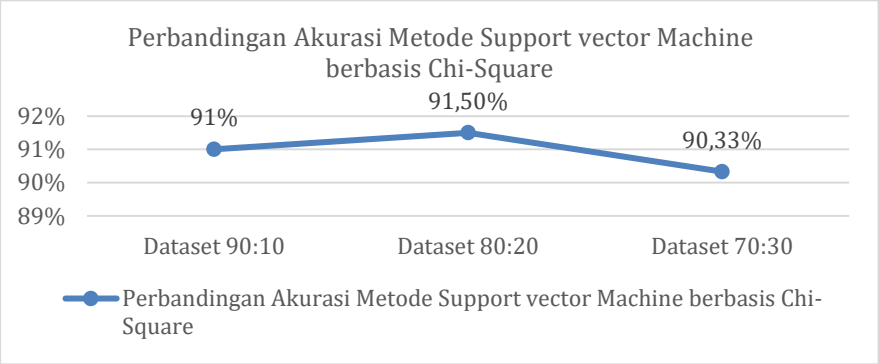


Gambar 4. Hasil Akurasi NB + Chi-Square

Dari gambar sebelumnya bisa diambil kesimpulan bahwasanya akurasi terbaik dari pengujian memakai algoritma *Naïve Bayes* berbasis *Chi-Square* didapatkan pada *rasio dataset* 80:20 dengan akurasi sebesar 79%.

3.8. Klasifikasi SVM berbasis Chi-square

Di pengujian ini yaitu menerapkan penyeleksian fitur *chi-square* ke dalam algoritma SVM. Langkah pertama adalah *Impor library* yang dibutuhkan selanjutnya Inisialisasi *Chi-Square* untuk pemilihan fitur serta Melakukan seleksi fitur pada data training serta menyesuaikan selector dengan data serta label. *Library python* yang dipakai dalam tahapan ini adalah *chi2* dari *sklearn. y_pred = svm.predict(X_test_chi2)* Memprediksi label pada data uji memakai model SVM yang sudah dilatih. Berikut hasil pengujiannya.

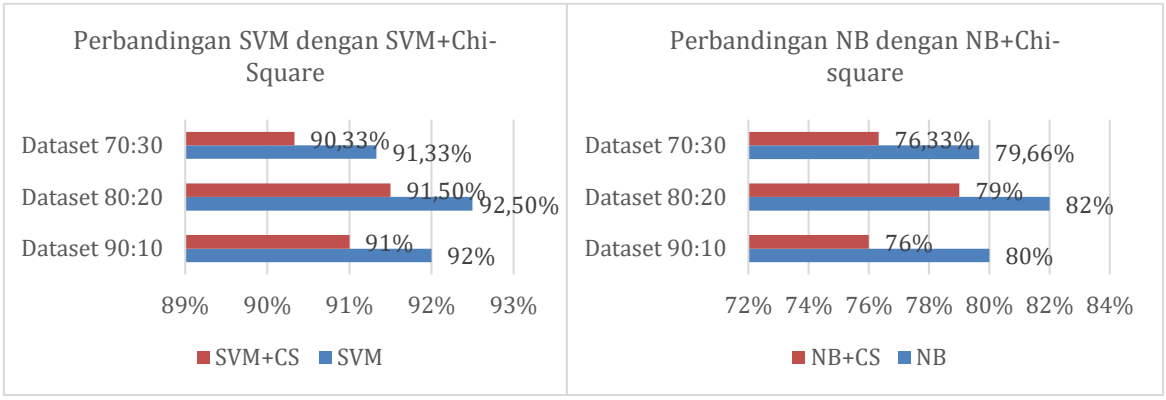


Gambar 5. Hasil Akurasi SVM + Chi-Square

akurasi terbaik dari pengujian menggunakan algoritma *SVM* berbasis *Chi-Square* didapatkan pada *rasio dataset* 80:20 dengan akurasi sebesar 91,50%.

3.9. Rangkuman Penelitian

Penerapan algoritma *Chi-Square* dalam mengoptimalkan metode SVM serta *Naïve Bayes* untuk analisa sentimen yang bertujuan untuk mencari model terbaik serta mencari hasil akurasi yang lebih baik telah mendapatkan hasil sebagai berikut.



Gambar 7. Perbandingan NB dengan NB+Chi-square

Dari grafik diatas diangkat kesimpulan bahwa pengujian metode SVM serta *Naïve Bayes* yang dioptimasi dengan *Chi-Square,* untuk *Naïve Bayes* berbasis *Chi-Square* mengalami penurunan pada setiap dataset, rasio data training serta data testing 90:10 dengan akurasi awal 80% turun dengan selisih 4% dengan akurasi 76%, metode *naïve bayes* berbasis *Chi-Square* rasio 80:20 dengan akurasi awal *Naïve Bayes*

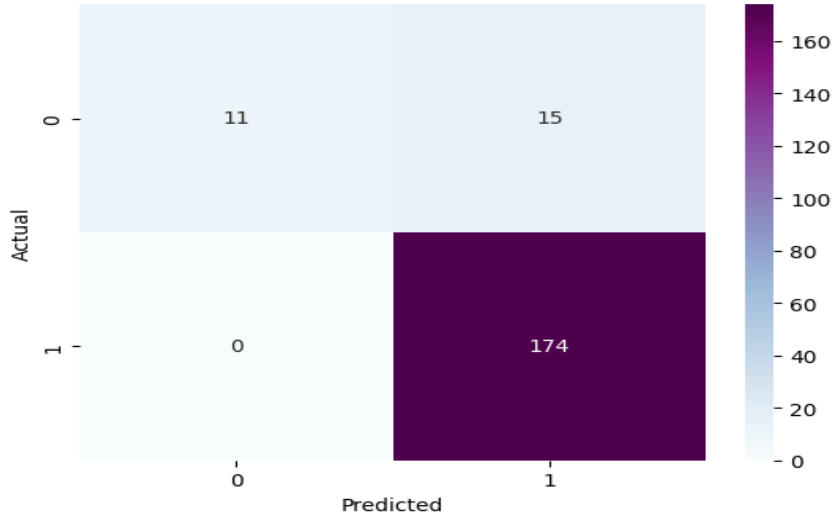
82% turun dengan selisih 3% setelah mengunakan *Naïve Bayes* berbasis *Chi-square* akurasinya menjadi 79% serta rasio 70:30 dengan akurasi awal *Naïve Bayes* 79,66% turun dengan selisih 3,33% setelah mengunakan *Naïve Bayes* berbasis *Chi-square* akurasinya

menjadi 76,33%. Mendapatkan model terbaik mengunakan Naïve Bayes dengan akurasi 82% serta datasetnya 80:20.

Dari grafik diatas dengan perbandingan SVM sebelum serta sesudah mengunakan *Chi-Square* rasio 90:10 dengan akurasi awal SVM 92% turun dengan selisih 1% setelah mengunakan SVM berbasis *Chi-square* akurasinya menjadi 91%, rasio 80:20 dengan akurasi awal SVM 92,50% turun dengan selisih 1% setelah mengunakan SVM berbasis *Chi-square* akurasinya menjadi 91,50% serta rasio 70:30 dengan akurasi awal SVM 91,33% turun dengan selisih 1% setelah mengunakan SVM berbasis *Chi-square* akurasinya menjadi 90,33% mendapatkan akurasi terbaik pada metode SVM pada dataset 80:20 dengan akurasi 92,50%.

3.10. Evaluasi

Sesudah melewati proses verifikasi, dilanjutkan ke tahap penilaian algoritma memakai *confusion matrix*. Pengelompokan memakai pendekatan SVM menjadi struktur unggulan dalam studi ini, ketepatan tertinggi diperoleh pada data latih 80:20, sebagaimana berikut.



Gambar 8. Confusion Matriks SVM Dataset 80:20

Nilai akurasi dihitung dengan persamaan berikut:

$$Confusion\ Matriks : \frac{PB+NS}{PB+NB+PS+NS}\ X\ 100\%$$

$$Confusion\ Matriks : \frac{11+174}{11+15+0+174}\ X\ 100\%$$

$$Confusion\ Matriks : 92{,}5\%$$

4. KESIMPULAN DAN SARAN

Penerapan algoritma *Chi-Square* dalam upaya mengoptimalkan metode SVM serta *Naïve Bayes* tidak berhasil karena kedua metde tidak mengalami peningkatan akurasi semula yang dihasilkan metode SVM serta *Naïve Bayes* setelah ditambahkan dengan *Chi-square*.

Model terbaik yang didapat pada penelitian yaitu metode SVM pada data training serta data testing 80:20, mendapatkan akurasi pada angka 92,50%.

Pada penelitian berikutnya diharapkan untuk memperluas penelitian ini serta mencoba mengunakan metode lain untuk dikombinsikan dengan *Chi-square* supaya medapatakan hasil akurasi yang lebih baik saat melaksanakan analisis senitmen. Pada studi mendatang disarankan

melaksanakan penyempurnaan terhadap pemilihan atribut tambahan guna meningkatkan tingkat ketepatan di metode Naïve Bayes serta SVM saat menelaah opini.

REFERENSI

[1] D. Darwis, N. Siskawati, and Z. Abidin, "Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional," *J. Tekno Kompak*, vol. 15, no. 1, p. 131, 2021, doi: 10.33365/jtk.v15i1.744.

[2] W. C. F. Mariel, S. Mariyah, and S. Pramana, "Sentiment analysis: A comparison of deep learning neural network algorithm with SVM and naïve Bayes for Indonesian text," *J. Phys. Conf. Ser.*, vol. 971, no. 1, 2018, doi: 10.1088/1742-6596/971/1/012049.

[3] N. L. K. I. Pramestia, M. A. Raharjaa, N. A. S. ERa, and I. G. A. Wibawaa, "Analisis Sentimen Ulasan Aplikasi Solusi Kota Cerdas Menggunakan Algoritma Naïve Bayes dan Support Vector Machine (SVM) dengan Seleksi Fitur Chi-Square," *J. Elektron. Ilmu Komput. Udayana p-ISSN*, vol. 2301, p. 5373.

[4] B. R. Atmadja, "Analisis Sentimen Bahasa Indonesia Pada Tempat Wisata Di Kabupaten Sukabumi Dengan Naive Bayes Classifier," *Elkom J. Elektron.* dan *Komput.*, vol. 15, no. 2, pp. 371–382, 2022.

[5] A. Yoga Pratama *et al.*, "Analisis Sentimen Media Sosial Twitter Dengan Algoritma K-Nearest Neighbor dan Seleksi Fitur Chi-Square (Kasus Omnibus Law Cipta Kerja)," *J. Sains Komput. Inform. (J-SAKTI*, vol. 5, no. 2, pp. 897–910, 2021.

[6] C. F. Hasri and D. Alita, "Penerapan Metode Naïve Bayes Classifier dan Support Vector Machine Pada Analisis Sentimen Terhadap Dampak Virus Corona Di Twitter," *J. Inform.* dan *Rekayasa Perangkat Lunak*, vol. 3, no. 2, pp. 145–160, 2022.

[7] M. IRFAN, "Named Entity Recognition Untuk Data Review Tempat Wisata Dengan Metode 'Bidirectional Encoder Representations From Transformers,'" 2021.

[8] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, "Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19)," *J. Media Inform. Budidarma*, vol. 5, no. 2, pp. 406–414, 2021.

[9] J. A. Septian, T. M. Fachrudin, and A. Nugroho, "Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor," *INSYST J. Intell. Syst. Comput.*, vol. 1, no. 1, pp. 43–49, 2019.

[10] A. Z. Amrullah, A. S. Anas, and M. A. J. Hidayat, "Analisis Sentimen Movie Review Menggunakan Naive Bayes Classifier Dengan Seleksi Fitur Chi Square," *J. Bumigora Inf. Technol.*, vol. 2, no. 1, pp. 40–44, 2020.

[11] M. A. Imron, "Peningkatan Akurasi Algoritma K-Nearest Neighbor Menggunakan Normalisasi Z-Score dan Particle Swarm Optimization Untuk Prediksi Customer Churn," 2020.