"BIMASAKTI"
Jurnal Riset Mahasiswa Bidang Teknologi Informasi

Volume 8 Nomor 1
P ISSN : 2355-4401–E ISSN 3109-1652
Available online at https://ejournal.unikama.ac.id/index.php/JFTI

OPTIMALISASI ANALISIS SENTIMEN FILM PADA YOUTUBE DENGAN ALGORITMA CHI-SQUARE PADA NAÏVE BAYES DAN SUPPORT VECTOR MACHINE

Oky Kurnia Wardana[1)], Alexius Endy Budianto[2)], Moh Ahsan[3)]

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

Abstrak

Analisis opini terhadap film di platform YouTube mengunakan algoritma Naïve Bayes dan Support Vector Machine. Fokus penelitian adalah meningkatkan akurasi klasifikasi melalui penerapan seleksi fitur mengunakan metode Chi-square. Data yang digunakan diperoleh melalui proses web scraping pada ulasan pengguna di Google Play Store. Labeling secara otomatis dengan bantuan library Transformers, dengan hasil label positif 221 dan negatif 759 dari 1000 ulasan. Tahapan penelitian meliputi crawling data, pelabelan otomatis mengunakan library transformers, pre-processing (case folding, tokensiasi, stopword removal, normalisasi, dan stemming), pembobotan dengan metode TF-IDF, dan pengujian akurasi model mengunakan rasio pembagian data 90:10, 80:20 dan 70:30. Hasil penelitian menunjukan bahwa 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, dan akurasi Naïve Bayes berbasis Chi-square 79% pada dataset 80:20. Chi-square tidak mampu meningkatkan performa dari metode Naïve Bayes dan Support Vector Machine pada penelitian ini.

Kata Kunci : Sentiment Analysis; Naïve Bayes; Support Vector Machine; Chi- square; YouTube.

Abstract

Analyze opinions on films on the YouTube platform using the Naïve Bayes algorithm and Support Vector Machine. The focus of the study is to improve classification accuracy through the application of feature selection using the Chi-square method. The data used were obtained through the web scraping process on user reviews on the Google Play Store. Automatic labeling with the help of the Transformers library, with the results of 221 positive labels and 759 negative labels from 1000 reviews. The research stages include data crawling, automatic labeling using the transformers library, pre-processing (case folding, tokenization, stopword removal, normalization, and stemming), weighting with the TF-IDF method, and testing model accuracy using a data sharing ratio of 90:10, 80:20 and 70:30. The results of the study showed that the Support Vector Machine method obtained an accuracy of 92.5% on the 80:20 dataset, while the Chi-square-based Support Vector Machine obtained an accuracy of 91.5% on the 80:20 dataset, Naïve Bayes obtained an accuracy of 82% on the 80:20 dataset, and the accuracy of Chi-square-based Naïve Bayes was 79% on the 80:20 dataset. Thus, Chi-square was unable to improve the performance of the Naïve Bayes and Support Vector Machine methods in this study.

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

1. PENDAHULUAN

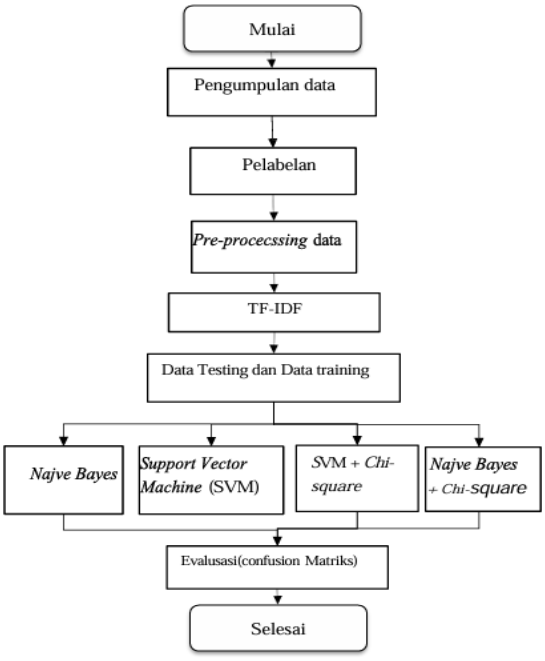
Banyak film yang bisa diakses melalui internet terdapat banyak website yang menyediakan film, setiap kalangan Masyarakat dapat dengan mudah menonton film, seperti di bioskop, televisi, aplikasi, dan website. Dengan kemajuan teknologi saat ini yang begitu pesat juga menimbulkan adanya pergantian aktivitas kehidupan manusia dalam berbagai bidang, yang secara langsung sudah mempengaruhi lahirnya berbagai bentuk perbuatan hukum yang baru. YouTube adalah layanan yang memungkinkan pengguna untuk menonton acara favorit mereka kapan saja, di mana saja di hampir semua media (smartphone, smart TV, tablet, PC, laptop). Pemrosesan analisis sentimen terdapat algoritma yang bisa digunakan, seperti algoritma *Native*

Bayes. Algoritma *Native Bayes* merupakan salah satu metode klarifikasi yang banyak digunakan pada Data Mining ataupun *Text Mining*. Metode *Native Bayes* berdasarkan *teorema bayes* bahwa semua kegiatan memberikan sebuah kontribusi yang sama penting atau saling bebas pada pemilihan kelas tertentu. Salah satu metode klarifikasi untuk menemukan gambaran persepsi Masyarakat di dalam *Text Mining* adalah metode *Native Bayes* yang sering disebut dengan *Native Bayes Classifier* (Darwis et al., 2021). *Support Vector Machine* merupakan sistem pembelajaran yang menggunakan hipotesis berupa fungsi-fungsi linear dalam sebuah fitur yang berdimensi tinggi dan dilatih dengan menggunakan algoritma pembelajaran yang didasarkan dengan teori optimasi *Support Vector Machine* diperkenalkan pertama kali oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep konsep unggulan dalam bidang pattern recognition Tingkat akurasi pada model yang akan dihasilkan oleh proses peralihan pada svm sangat bergantung pada fungsi kernel dan parameter yang digunakan. Analisis Sentimen merupakan kajian tentang cara menyelesaikan dan memecahkan masalah dari berdasarkan opini Masyarakat, Sikap serta emosi suatu entitas, Dimana entitas tersebut dapat mewakili individu. Analisis Sentimen atau juga disebut *opinion mining* merupakan proses memahami, mengekstrak serta mengolah data tekstual secara otomatis guna mendapatkan informasi yang terkandung dalam suatu kalimat opini. Dilakukan analisis sentiment ini bertujuan untuk melihat pendapat atau kecendrungan opini terhadap suatu masalah ataupun objek oleh seseorang, apa memiliki kecendrungan positif ataupun negatif.

2. METODE / ALGORITMA

1) Rancangan Penelitian

Dalam menerapkan *Chi-Square* untuk meningkatkan performa algoritma *Naïve Bayes* dan *support vector machine (SVM)*, berikut rancangan penelitian yang akan dilakukan:



Gambar 1 Rancangan Penelitian

2) Crawling Data

Pada tahapan ini, pengumpulan data dilakukan dengan teknik *crawling* menggunakan *python*. *Crawling* dilakukan untuk mengambil data ulasan terhadap aplikasi YouTube. Data yang berhasil didapatkan berjumlah 1000, dengan hanya mengambil data pada *atribut* content dan *score.* lalu akan disimpan untuk masuk ke tahap selanjutnya.

3) Pelabelan

Setelah dilakukan pengumpulan data, masuk ke tahapan pelabelan. Dalam prosesnya data akan dibaca oleh sistem, lalu ulasan akan diterjemahkan kedalam bahasa inggris terlebih dahulu dikarenakan sistem tidak mengenali bahasa Indonesia, barulah sistem akan memberi label kepada ulasan secara keseluruhan.

4) Pre-Processing

Tahapan awal dalam proses klasifikasi menggunakan algoritma *Naïve Bayes* berbasis *Chi-Square* dan kedua *Support Vector Machine* berbasis *Chi-Square* adalah proses *pre-processing*. Tujuan *pre- processing* (pra-pemrosesan) dalam analisis data adalah untuk membersihkan, mempersiapkan, dan mentransformasikan data mentah menjadi bentuk yang lebih sesuai untuk analisis. *Pre-processing* merupakan tahap penting dalam proses analisis data, karena kualitas dan relevansi data yang digunakan secara langsung mempengaruhi hasil analisis yang dihasilkan. Adapun tahapan yang digunakan dalam *pre-processing* sebagai berikut:

Tabel .1 Tahapan Pre-processing

No	Tahapan	Penjelasan
1	*Case Folding*	Merubah seluruh kata menjadi *lower case.* Lalu menghapus *url*, tanda baca, dan *emoji*
2	*Tokenisasi dan Frekuensi Token*	Memisahkan kata menjadi pertoken dan menghitung kemunculan kata didalam dokumen.
3	*Stopword Removal*	Menghapus kata yang tidak memiliki makna penting.
4	*Normalisasi Kata*	Mengembalikan kata singkat dan slang ke kata sebenarnya.
5	*Stemming*	Mengembalikan kata menjadi kata dasar.

5) TF-IDF

Setelah tahapan *pre-processing* sudah dilakukan, tahap selanjutnya penghitungan bobot dengan *tf-idf*. Pembobotan *tf-idf (term frequency-inverse document frequency)* adalah salah satu metode yang umum digunakan dalam pengolahan teks. Tujuannya adalah untuk mengukur relevansi setiap kata dalam sebuah dokumen berdasarkan seberapa sering kata tersebut muncul di dokumen tersebut dan seberapa umum kata tersebut dalam seluruh korpus dokumen. Pembobotan *tf-idf* menggabungkan dua konsep penting: *Term Frequency (TF)*: Mengukur seberapa sering sebuah kata muncul dalam sebuah dokumen. Semakin sering kata tersebut muncul, semakin tinggi bobotnya. Kemudian, *Inverse Document Frequency (IDF)*: Mengukur seberapa umum sebuah kata dalam seluruh korpus dokumen. Kata-kata yang umum seperti "dan" atau "the" memiliki *IDF* yang rendah, sedangkan kata- kata yang lebih jarang muncul memiliki *IDF* yang tinggi. Bobot *tf-idf* diperoleh dengan mengalikan nilai *TF* dengan nilai *IDF*. Dengan menggunakan metode ini, kata-kata yang sering muncul di dalam dokumen tetapi jarang muncul di seluruh korpus akan memiliki bobot yang tinggi, menunjukkan tingkat relevansi yang lebih tinggi dalam dokumen tersebut.

6) Data Testing dan Data Training

Pada tahapan ini, pembagian *data testing* dan *data training* dilakukan secara otomatis pada sistem menggunakan model *train test split*. Rasio data yang digunakan pada penelitian ini adalah 70:30,80:20, dan 90:10, menggunakan 3 rasio dataset yaitu dengan tujuan mendapatkan

lebih banyak variasi akurasi untuk mencari model terbaik penelitian (Rahayu et al., 2021)

7) Klasifikasi Support Vector Machine (SVM), Naïve Bayes dan Chi-Square

Setelah melalui tahapan *pre-processing*, pembobotan dan pembagian *data testing* dan *data training*, selanjutnya masuk ke tahapan seleksi fitur menggunakan *chi-square* yang berguna untuk mempercepat pemrosesan dan mengurangi atribut yang kurang relevan pada data. Lalu berikutnya masuk ke tahapan klasifikasi menggunakan algoritma *Naïve Bayes*. Pengujian akan dilakukan menggunakan tiga rasio perbandingan data, tahapan klasifikasi akan diuji menggunakan algoritma *Naïve Bayes* dan *Support Vetor Machine* berbasis *chi-square*. Dari semua hasil klasifikasi yang telah didapatkan, akan dicari model terbaik dari kedua algoritma.

3. HASIL DAN PEMBAHASAN

1) Hasil dan Pebahasan Crawling Data

Dalam penelitian ini, data yang digunakan adalah data yang diperoleh dari ulasan pengguna aplikasi *YouTube*. Pengumpulan data menggunakan teknik *scrapping* pada *Google Play Store*. Adapun tahapan yang dilakukan adalah install *google play scraper* pada *python*, lalu memasukan id aplikasi yang dituju, dalam hal ini 'com.YouTube.mediaclient'. Data yang diperoleh berjumlah 1000 data yang diambil pada tanggal 8 juli 2024 dengan hanya mengambil data pada *atribut* content dan *score.* Berikut adalah *source code* yang digunakan untuk *scrapping*.

```
from google_play_scraper import app
        from google_play_scraper import Sort, reviews
        result, continuantion_token = reviews(
          'com.youtobe.mediaclient',
          lang='id',
          country='id',
          sort=Sort.NEWEST,
          count=1000,
          filter_score_with=None
df_busu = pd.DataFrame(np.array(result),columns=['review'])
df_busu = df_busu.join(pd.DataFrame(df_busu.pop('review').tolist()))
        data_baru = df_busu[['score', 'content']]
data_baru.to_csv("data_scraping.csv", index = False)
```

Gambar 2 Source Code Scriping Data

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 (Estika et al., 2021) Akan tetapi dalam pemberian label tetap merujuk pada hasil dari *library transformers*

Tabel .2 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

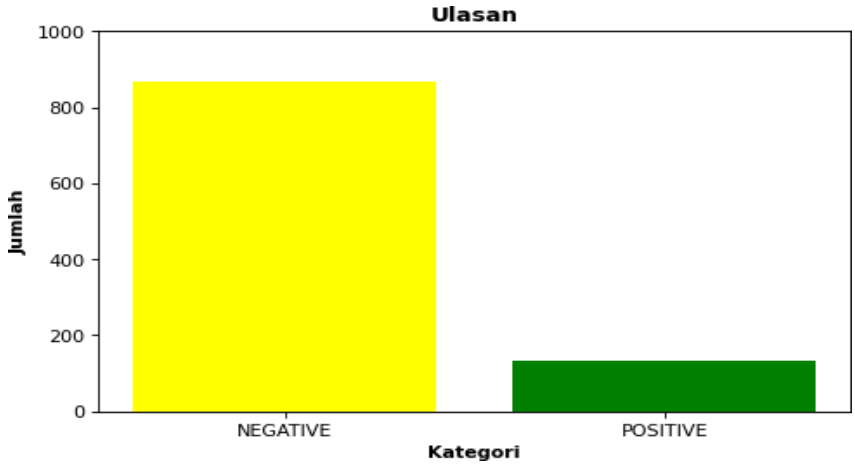
2) Labeling Ulasan

Dalam penelitian ini dalam melakukan pelabelan secara otomatis menggunakan bantuan dari *library transformers*, dengan hasil ulasan positif dan 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 dan berlabel negatif di *replace* dengan angka 1. Berikut hasil labeling secara otomatis menggunakan bantuan *library transformers.*

Tabel 3 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 221 dan ulasan berlabel negatif berjumlah 759, dapat dilihat pada gambar berikut.



Gambar 3 Diagram Jumlah Hasil Labeling

3) Pre-precossing Data

Ulasan pengguna pada aplikasi *YouTube* sudah melalui beberapa tahapan yakni labeling otomatis menggunakan bantuan *library transformers* dan *replace values* pada atribut label. Berikutnya masuk ketahap *pre-processing* data yang bertujuan agar mendapatkan data yang lebih bersih dan baik untuk masuk ketahapan berikutnya. Adapun tahapan ini terdiri dari *case folding*, *tokenisasi* dan *frekuensi token*, *stopword removal,* normalisasi kata, dan *stemming.*

4) Pembobotan TF-IDF

Setelah melalui tahapan *pre-processing* selanjutnya dilakukan tahap pembobotan kata, tahapan ini bertujuan untuk menghitung bobot nilai dari frekuensi kata didalam dokumen dan juga bobot nilai dari frekuensi kata dalam banyak. Pembobotan ini untuk melihat sejauh mana tingkat relevan sebuah kata didalam dokumen.

```
# Mendapatkan Feature Extraction
from sklearn.feature_extraction.text import
TfidfVectorizer tfidf =
TfidfVectorizer().fit(dataa['komentar'])
tfidf.fit(dataa['komentar'])
(len(tfidf.get_feature_names_out()))
# Melihat Matriks Jumlah Token
x_tf_idfaja = tfidf.transform(dataa['komentar']).toarray()
x_tf_idfaja # Melihat Matriks Jumlah Token Menggunakan
TF-IDF
# Data Ini Siap Untuk Dimasukkan Dalam Proses Pemodelan
```

Gambar 4. Source Code Pemobotan TF-IDF

Setelah melalui tahapan pembobotan *tf-idf*, dapat dilihat hasilnya pada tabel berikut.

Tabel 4. Hasil TF-IDF

	abdi	**abis**	**abissitu**	**about**	**abu**	…	**Yokk**	**You**
0	0.0	0.0	0.0	0.0	0.0	…	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	…	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	…	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	…	0.0	0.0
…	…	…	…	…	…	…	…	…
999	0.0	0.0	0.0	0.0	0.0	…	0.0	0.0

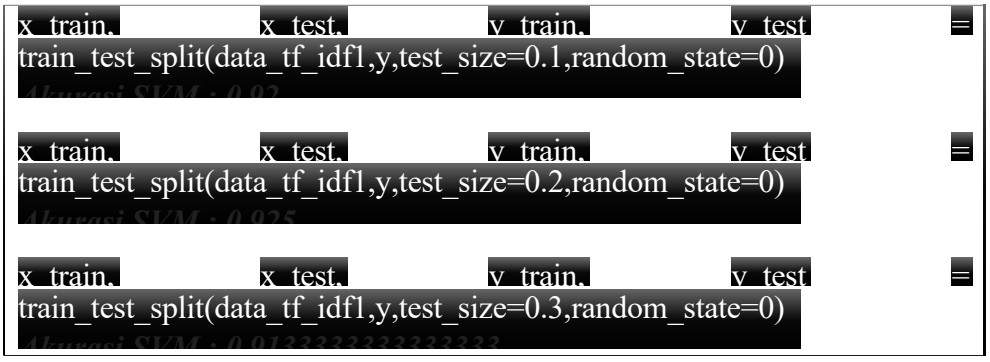
5) Data Testing dan Data Training

Setelah melalui tahapan pembobotan menggunakan *tf-idf*, selanjutnya masuk ketahapan pembagian data *testing* dan *training*. Dalam penelitian ini memakai 3 rasio perbandingan untuk data *testing* dan data *training*, yakni 90:10, 80:20, dan 70:30. Pembagian data ini dilakukan secara otomatis menggunakan model *train test split.*

6) Klasifikasi

a) Support Vector Machine

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



Gambar 5 Hasil Pengujian SVM

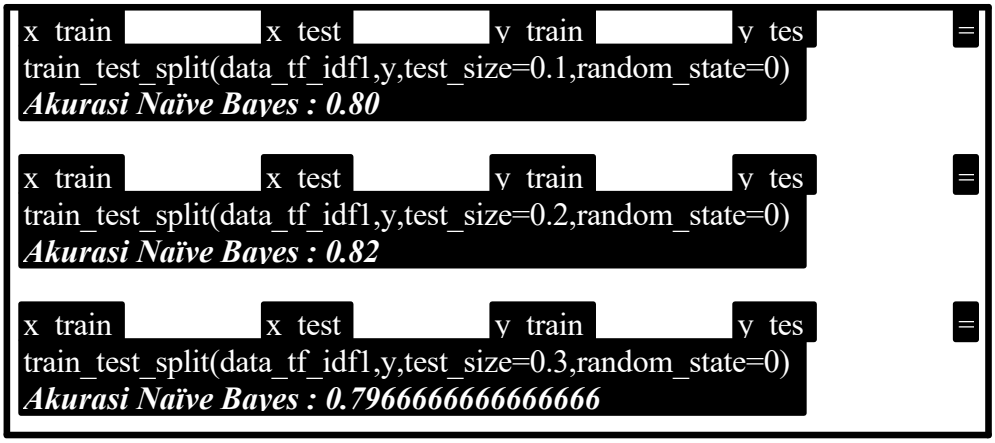
Dari gambar diatas, pada *rasio dataset* 90:10 algoritma SVM mendapatkan akurasi sebesar 92%, pada *rasio dataset* 80:20 mendapatkan akurasi sebesar 92. 5%, dan pada *rasio dataset* 70:30 mendapatkan akurasi sebesar 91.33333333333333%. Setelah dilakukan pengujian dapat diangkat kesimpulan bahwa akurasi terbaik didapatkan pada *rasio dataset* 90:10 dengan akurasi sebesar 92 %.
Perhitungan manual:
Misal:
Dataset total = 1000 data
Rasio 90:10 → Data Training = 900, Data Testing = 100
Katakan dari 100 data testing, 92 data diprediksi benar → berarti:

$$\text{Akurasi} = \left(\frac{92}{100}\right) \times 100\% = 92\%$$

Kalau datamu lebih banyak, prinsipnya tetap sama.

b) Naïve Bayes

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



Gambar 6 Hasil Pengujian Naïve Bayes

Dari gambar diatas, pada *rasio dataset* 90:10 algoritma *Naïve Bayes* mendapatkan akurasi sebesar 80%, pada *rasio dataset* 80:20 mendapatkan akurasi sebesar 82%, dan pada *rasio dataset* 70:30 mendapatkan akurasi sebesar 79,66%. Setelah dilakukan pengujian dapat diangkat kesimpulan bahwa akurasi terbaik didapatkan pada *rasio dataset* 80:20 dengan akurasi sebesar 82%.
Perhitungan Manual :

$$\text{Akurasi} = \left(\frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Data Uji}}\right) \times 100\%$$

Langkah menghitung manual:

a) Tentukan jumlah data total di dataset.
b) Hitung berapa data testing (sesuai rasio: 90:10, 80:20, 70:30).
c) Cocokkan jumlah prediksi benar (berdasarkan akurasi yang diperoleh).

d) Masukkan ke rumus.
Contoh Perhitungan:
Misal jumlah total data = 300 data.
e) Rasio 80:20 berarti:
Data Training: 80% × 300 = 240 data
Data Testing: 20% × 300 = 60 data
Dari hasil kamu, akurasi 82% pada rasio 80:20.
Berarti jumlah prediksi benar:
Jumlah Prediksi Benar=82%×60=49.2\text{Jumlah Prediksi Benar} = 82\% \times 60 = 49.2Jumlah Prediksi Benar=82%×60=49.2
Karena jumlah data harus bulat, biasanya dibulatkan → sekitar 49 atau 50 data benar.
Kalau 49 data benar

$$\text{Akurasi} = \left(\frac{49}{60}\right) \times 100\% = 81.666\%$$

Kalau 50 data benar

$$\text{Akurasi} = \left(\frac{50}{60}\right) \times 100\% = 83.333\%$$

c) Support Vector Machine Berbasis Chi-Square

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

Gambar .7 Hasil Pengujian SVM Berbasis Chi-square

Dari gambar diatas, pada *rasio dataset* 90:10 algoritma *SVM* berbasis *Chi-square* mendapatkan akurasi sebesar 91%, pada *rasio dataset* 80:20 mendapatkan akurasi sebesar 91,5%, dan pada *rasio dataset* 70:30 mendapatkan akurasi sebesar 90,33%. Setelah dilakukan pengujian dapat diangkat kesimpulan bahwa akurasi terbaik didapatkan pada *rasio dataset* 80:20 dengan akurasi sebesar 91,5%. Inti dasar menghitung akurasi manual tetap:

$$\text{Akurasi} = \left(\frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Data Uji}}\right) \times 100\%$$

Perhitungan manual:

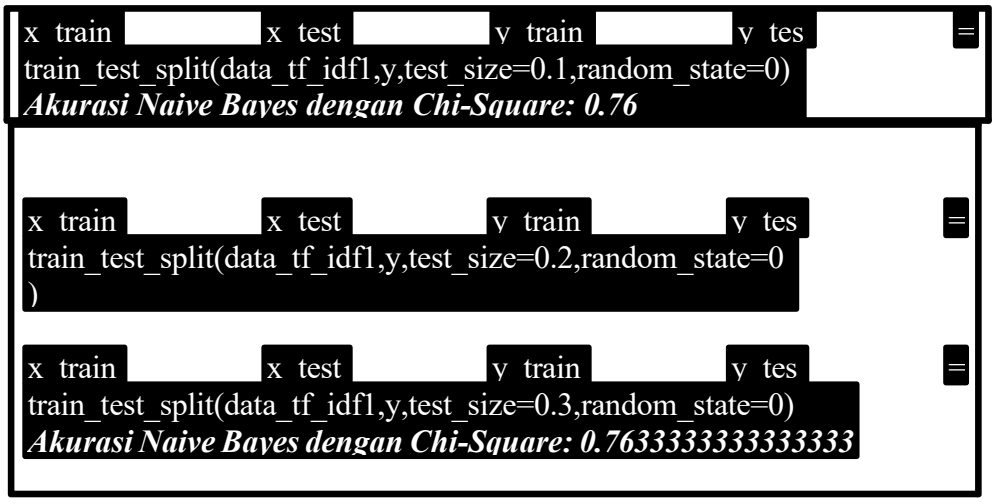
1. Tentukan total data di dataset (contohnya: 300 data, 1000 data, dll).
2. Hitung jumlah data uji berdasarkan rasio (misal 80:20 → 20% untuk testing).
3. Gunakan akurasi yang didapat (misal 91,5%) untuk mencari jumlah prediksi benar.
4. Cek dengan rumus.

Contoh perhitungan (anggap dataset total 300 data):

- Rasio 80:20, artinya:
  Training = 80% × 300 = 240 data
  Testing = 20% × 300 = 60 data
  Diketahui akurasi pada rasio ini = 91,5%.
  Mencari jumlah prediksi benar:
  $$\text{Jumlah Prediksi Benar} = 91.5\% \times 60 = 54.9$$
  Karena jumlah data harus bulat, sekitar 55 data benar dari 60 data testing.
  Verifikasi akurasi:
  Kalau 55 data benar:
  $$\text{Akurasi} = \left(\frac{55}{60}\right) \times 100\% = 91.6666\%$$
  Hampir sama dengan 91,5%, hanya selisih kecil, biasanya diterima dalam praktik.

d) Naïve Bayes Berbasis Chi-Square

Pada pengujian ini yaitu menerapkan seleksi fitur *chi-square* kedalam algoritma *Naïve Bayes*. Seleksi fitur digunakan untuk mengurangi atribut atau fitur yang tidak relevan adalam *dataset* dan mempercepat pemrosesan data, *chisquare* dipilih karena mampu mengukur hubungan antara variabel nominal dan variabel nominal lainnya dan meningkatkan akurasi prediksi atau kinerja dataset dengan memilih fitur-fitur yang paling relevan.
Berikut adalah hasil pengujian menggunakan algoritma *Naïve Bayes.*



Gambar .8 Hasil Pengujian Naïve Bayes Berbasis Chi-square

Dari gambar diatas, pada *rasio dataset* 90:10 algoritma *Naïve Bayes* berbasis *Chi-Square* mendapatkan akurasi sebesar 76%, pada *rasio dataset* 80:20 mendapatkan akurasi sebesar 79%, dan pada *rasio dataset* 70:30 mendapatkan akurasi sebesar 76,33%. Setelah dilakukan pengujian dapat diangkat kesimpulan bahwa akurasi terbaik didapatkan pada *rasio dataset* 80:20 dengan akurasi sebesar 79%.
Menghitung Manual, rumus dasara akurasi :

$$\text{Akurasi} = \left(\frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Data Testing}} \right) \times 100\%$$

Langkah-langkah menghitung manual:

1. Tentukan jumlah total data (contoh: 300 data, 500 data, dsb).
2. Hitung jumlah data testing berdasarkan rasio.
   Misal rasio 80:20 → testing = 20% × total data.
3. Kalikan jumlah data testing dengan akurasi untuk dapat jumlah prediksi benar.
4. Cek ulang dengan rumus akurasi.

Perhitungan (anggap dataset total 300 data):

a) Rasio 80:20 berarti:
   Data training: 80% × 300 = 240 data
   Data testing: 20% × 300 = 60 data
   Akurasi pada rasio ini: 79%.
   Hitung jumlah prediksi benar:
   $\text{Jumlah Prediksi Benar} = 79\% \times 60 = 47.4$
   Karena data harus bulat, maka kira-kira 47 atau 48 data diprediksi benar.
   Kalau 47 benar:
   $$\text{Akurasi} = \left(\frac{47}{60} \right) \times 100\% = 78.33\%$$
   Kalau 48 benar:
   $$\text{Akurasi} = \left(\frac{48}{60} \right) \times 100\% = 80\%$$
   Kesimpulannya:
   a) Prediksi benar sekitar 47–48 data dari 60 data testing.
   b) Akurasi mendekati 79%, sesuai hasil pengujian

Kesimpulan Umum dari Semua Perhitungan Manual:

1. Cara menghitung akurasi manual untuk semua algoritma (SVM, Naïve Bayes, SVM-ChiSquare, Naïve Bayes-ChiSquare) adalah sama, yaitu:
   $$\text{Akurasi} = \left(\frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Data Testing}} \right) \times 100\%$$
2. Langkah-langkah Umum:
   - ✓ Tentukan jumlah total dataset.
   - ✓ Bagi dataset sesuai rasio (90:10, 80:20, 70:30).
   - ✓ Hitung jumlah data testing berdasarkan rasio.
   - ✓ Dari akurasi (%) yang didapat → cari berapa banyak prediksi benar.
   - ✓ Verifikasi perhitungan menggunakan rumus akurasi.
3. Kesimpulan Khusus dari Hasil:
   - ✓ SVM:
   - Rasio 80:20 menghasilkan akurasi 92.5% → terbaik.
   - ✓ Naïve Bayes:
   - Rasio 80:20 menghasilkan akurasi 82% → terbaik.
   - ✓ SVM berbasis Chi-Square:
   - Rasio 80:20 menghasilkan akurasi 91.5% → terbaik.
   - ✓ Naïve Bayes berbasis Chi-Square:
   - Rasio 80:20 menghasilkan akurasi 79% → terbaik.

   Jadi, di semua kasus, akurasi terbaik didapatkan saat menggunakan rasio dataset 80:20.

Berdasarkan hasil perhitungan akurasi manual terhadap masing-masing metode, diperoleh bahwa akurasi terbaik secara konsisten dicapai pada rasio dataset 80:20. Hal ini menunjukkan bahwa pembagian data dengan rasio 80% untuk pelatihan dan 20% untuk pengujian mampu memberikan model yang lebih optimal dibandingkan rasio lainnya.

4. KESIMPULAN

Hasil penelitian yang telah dilakukan, dapat disimpulan bahwa:

1. Pada penelitian ini, penerapan algoritma *Chi-Square* dalam mengoptimalkan metode *Support Vector Machine* dan *Naïve Bayes* tidak berhasil meningkatkan akurasi awal yang dihasilkan oleh metode SVM dan Naïve Bayes.
2. Model terbaik yang didapatkan dalam penelitian ini yaitu pada metode *Support Vector Machine* dan *Naïve Bayes* yang dioptimasi dengan mengunakan *Chi- Square* mengalami penurunan pada setiap *rasio data training* dan *data testing* 80:20, pada *Support Vector Machine* dengan mendapatkan akurasi pada angka 92,50% adalah model terbaik dan untuk *Naïve Bayes* dengan *rasio data training* dan *data testing 80:20* mendapatkan akurasi pada angka 82% model terbaik.

5. REFERENSI

[1] Abdillah, Rezky. (2023). Analisis Sentimen Ulasan Aplikasi Wetv Untuk Peningkatan Layanan Menggunakan Metode Support Vector Machine. Journal of Information System Research (JOSH) Volume 4, No. 3, April 2023, pp 865−873

[2] Allorerung, Petronilia Palinggik . (2023). Analisis Sentimen Pada Ulasan Aplikasi WeTV di Google Play Store Menggunakan Algoritma NBC dan SVM. SISTEMASI: Jurnal Sistem Informasi ISSN:2302-8149 Volume 12, Nomor 2, Mei 2023: 404-414 Atmadja, B. R. (2022). Analisis Sentimen Bahasa Indonesia Pada Tempat Wisata Di Kabupaten Sukabumi Dengan Naive Bayes Classifier. *Elkom: Jurnal Elektronika Dan Komputer*, *15*(2), 371–382.

[3] Atmadja, B. R. (2022). Analisis Sentimen Bahasa Indonesia Pada Tempat Wisata Di Kabupaten Sukabumi Dengan Naive Bayes Classifier. *Elkom: Jurnal Elektronika Dan Komputer*, *15*(2), 371–382.

[4] Amrullah, A. Z., Sofyan Anas, A., & Hidayat, M. A. J. (2020). Analisis Sentimen Movie Review Menggunakan Naive Bayes Classifier Dengan Seleksi Fitur Chi Square. *Jurnal*, *2* (1), 40–44. https://doi.org/10.30812/bite.v2i1.804

[5] Anraeni, S. (2022). *Analisis penerapan metode K- Nearest Neighbor (K-NN) pada dataset citra penyakit malaria*. *3*(1), 17–29.

[6] Darwis, D., Siskawati, N., & Abidin, Z. (2021). Penerapan Algoritma Naive Bayes Untuk Analisis Sentimen Review Data Twitter Bmkg Nasional. *Jurnal Tekno Kompak*, *15*(1), 131. https://doi.org/10.33365/jtk.v15i1.744

[7] Era, D., Andryana, S., & Rubhasy, A. (2023). Perbandingan Algoritma Naïve Bayes Dan K-Nearest Neighbor pada Analisis Sentimen Pembukaan Pariwisata Di Masa Pandemi Covid 19. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *7*(1), 263–272.

[8] Fairuz, A. L., Ramadhani, R. D., & Tanjung, N. A. F. (2021). Analisis Sentimen Masyarakat Terhadap COVID-19 Pada Media Sosial Twitter. *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, *1*(1), 42–51. https://doi.org/10.20895/dinda.v1i1.180

[9] Fikri, M. I., Sabrila, T. S., & Azhar, Y. (2020). Comparison of Naïve Bayes and Support Vector Machine Methods in Twitter Sentiment Analysis. *Smatika Jurnal*, *10*(02), 71–76.

Irfan, M. (2022). Named Entity Recognition Untuk Data Review Tempat Wisata dengan

Metode "Bidirectional Encoder Representations from Transformers." *Universitas Islam Indonesia*.

[11] Khoirunnisaa, N., Nabila, K., Kesuma, N., Setiawan, S., Yunizar, A., & Yusuf, P. (2024). Klasifikasi Teks Ulasan Aplikasi YouTube Pada Google Play Store Menggunakan Algoritma Naive Bayes dan SVM. *SKANIKA: Sistem Komputer Dan Teknik Informatika*, *7*(1), 64–73.

[12] MP Firdaus. (2023). Perbandingan Algoritma K-Nearest Neighbor (KNN) dan Naive Bayes Classifier (NBC) dengan pelabelan Transformers serta Ektraksi Fitur TF-IDF dan N-Gram untuk Analisis Sentimen Terhadap Penundaan Pemilu. *Perbandingan Algoritma K-Nearest Neighbor (KNN) Dan Naive Bayes Classifier (NBC) Dengan Pelabelan Transformers Serta Ektraksi Fitur TF-IDF Dan N-Gram Untuk Analisis Sentimen Terhadap Penundaan Pemilu*, 5–24. https://repository.uinjkt.ac.id/dspace/handle/123456789/72466

[13] Septian, J. A., Fachrudin, T. M., & Nugroho, A. (2019). Analisis Sentimen Pengguna Twitter Terhadap Polemik Persepakbolaan Indonesia Menggunakan Pembobotan TF-IDF dan K-Nearest Neighbor. *Journal of Intelligent System and Computation*, *1*(1), 43–49. https://doi.org/10.52985/insyst.v1i1.36

[14] Yoga Pratama, A., Umaidah, Y., Voutama, A., Informatika, T., Ilmu Komputer, F., Singaperbangsa Karawang Ds Paseurjaya, U., Telukjambe Timur, K., Karawang, K., & Barat, J. (2021). Analisis Sentimen Media Sosial Twitter Dengan Algoritma K-Nearest Neighbor Dan Seleksi Fitur Chi-Square (Kasus Omnibus Law Cipta Kerja). *Jurnal Sains Komputer & Informatika (J-SAKTI*, *5*(2), 897–910.

[15] Rahayu, A. S., Fauzi, A., Rahmat, R., Navisa, S., Luqman Hakim, Aulia Nabilah, Rifa'I, A., Ardhani, R., Pratama, D., Fatihanursari, F., Kadeari, N. L. E., Heryanda, K. K., Netti, S. Y. M., Irwansyah, I., Rhajendra, M. D., Trianasari, N., Astawa, P., Hasan, J. S., Lubis, F. Guitar, E. (2021). Analisis Sentimen Ulasan Pengguna Untuk Peningkatan Layanan Menggunakan Algoritma Naïve Bayes (Studi Kasus : Bukalapak) Sentiment Analysis of User Review for Service Improvement Using Naive Bayes Algorithm (Case Study : Bukalapak). *Multitek Indonesia*, *4* (1), 28–44.