# Analisis Sentimen *Review* Pengguna *Website* IMDB Menggunakan Klasifikasi *Naïve Bayes*

1st Yuni Kardila
*Fakultas Rekayasa Industri*
*Universitas Telkom*
Bandung, Indonesia
yunikardila@student.telkomuniversity.ac.id

2nd Oktariani Nurul Pratiwi
*Fakultas Rekayasa Industri*
*Universitas Telkom*
Bandung, Indonesia
onurulp@telkomuniversity.ac.id

3rd Faqih Hamami
*Fakultas Rekayasa Industri*
*Universitas Telkom*
Bandung, Indonesia
faqihhamami@telkomuniversity.ac.id

**Abstrak—*Website* IMDB (*Internet Movie Database*) merupakan suatu web yang digunakan untuk menyediakan atau melihat berbagai informasi tentang jutaan film yang telah tayang, yang digunakan untuk melihat *review*, *rating*, pemeran dan kru dari film tersebut. Para pengguna baru biasanya suka melihat *review* pengguna lainnya sebelum menentukan film apa yang akan mereka tonton, namun semakin banyak dan semakin besar *review* yang diberikan akan semakin besar dampak penilaian tersebut terhadap keputusan para pengguna baru, sehingga apabila para pengguna baru tidak dapat mengartikan makna dari *review* yang diberikan, untuk mengatasi hal tersebut dilakukanlah analisis sentimen. Penelitian yang dilakukan kali ini bertujuan untuk menganalisa analisis sentimen terhadap *movie review* yang diberikan oleh pengguna *website* IMDB, dengan menggunakan algoritma *Naïve Bayes Multinomial*. Penelitian ini juga menggunakan *text preprocessing*, dan TF-IDF untuk meningkatkan nilai akurasi dari model tersebut. Hasil evaluasi menggunakan *confusion matrix* menunjukan nilai *accuracy* 88.93%, nilai *precision* 89.07%, nilai *recall* sebesar 89.14% dan nilai F1-*Score* 89.11%, dengan perbandingan data *training* dan *testing* 70:30. Hasil klasfikasi yang dilakukan menghasilkan data 7533 berlabel positif dan 7249 berlabel negatif, nilai tersebut menunjukan sebagian besar para pengguna IMDB berkomentar positif terhadap film yang mereka tonton.**

**Kata kunci—*Naïve Bayes, IMDb, sentiment analysis, text preprocessing***

## I. PENDAHULUAN

*Website* IMDB (*Internet Movie Database*) merupakan suatu web yang digunakan untuk menyediakan atau melihat berbagai informasi tentang jutaan film yang telah tayang, yang digunakan untuk melihat *review*, *rating*, pemeran dan kru dari film tersebut. sehingga menjadikan IMDB sebagai tempat *database* dari opini dan komentar yang beragam [1].

Kumpulan *review* film yang terdapat pada *website* IMDB merupakan hasil penilaian berupa opini dan rating yang diberikan oleh para pengguna. *Review* yang diberikan oleh setiap pengguna dapat dijadikan sebagai bahan referensi untuk para penonton baru, ataupun sebagai bahan kritikan kepada tim produksi mengenai filmnya dan masukan untuk film selanjutnya [2]. Beberapa pengguna baru biasanya membaca beberapa komentar dari pengguna lainnya yang sudah memberikan *review*, sebelum menentukan film mana yang akan mereka tonton, berdasarkan penilaian yang paling positif atau negatif dari film tersebut [3], namun semakin besar dan semakin banyak jumlah *review* yang diberikan, maka akan semakin berpengaruh penilaian tersebut terhadap keputusan para pengguna lainnya, sehingga apabila para pengguna baru tidak dapat mengartikan dan memahami kumpulan *review* tersebut, kesalahan penyampaian makna atau nilai yang ingin disampaikan dari *review* tersebut akan semakin besar [4]. Sehingga untuk menyelesaikan permasalahan tersebut dilakukanlah *sentiment analysis*, karena metode ini yang dapat mengklasifikasi keberhasilan suatu film berdasarkan *review* dari para pengguna lainnya [5].

*Sentiment Analysis* merupakan suatu teknik memahami, mengekstrak pendapat (opini), serta mengolah data *text* secara otomatis untuk mendapatkan nilai yang terkandung dalam suatu pendapat (opini). *Sentiment analysis* hanya berfungsi untuk mengelompokkan dan mengkategorikan opini-opini tersebut bersifat positif maupun negatif, untuk mengetahui tingkat akurasi dari pengelompokan nilai *review* tersebut diperlukan algoritma klasifikasi seperti *Naïve Bayes*, *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN) dan lain-lain.

Dalam penelitian [6] mengatakan algoritma *naïve bayes* memiliki tingkat akurasi dan performansi yang tinggi dalam proses klasifikasi teks. Pendapat yang sama juga dikatakan oleh [7] metode NBC dipilih karena tingkat akurasi yang tinggi, mudah untuk dipahami, dan memiliki cepat dalam mengklasifikasi data. Berdasarkan hasil penelitian terdahulu dan pertimbangan, peneliti memutuskan algoritma yang digunakan dalam penelitian ini adalah algoritma *naïve bayes*.

Penelitian yang dilakukan kali ini bertujuan untuk menganalisa analisis sentimen terhadap *movie review* yang diberikan oleh pengguna *website* IMDB, dengan menggunakan *text pre-processing* melalui tahapan *handling duplicate, case folding, cleaning, stopword removal,* dan *stemming*. Pemilihan *text pre-processing* untuk meningkatkan nilai akurasi yang dihasilkan, dengan menggunakan algoritma *naïve bayes* yang memiliki karakteristik efisien dan sederhana dalam melakukan klasifikasi. Hasil analisis sentimen tersebut akan dilakukan analisa dan klasifikasi menggunakan algoritma *naïve bayes* yang memiliki jenis yaitu *Multinomial Naïve Bayes* (MNB). Pada penelitian ini, klasifikasi dari analisis sentimen yang

dilakukan dengan cara pendekatan menghitung semua kata yang sama dalam satu dokumen, sehingga dapat dicari nilai dari kata tersebut bernilai positif atau negatif, permasalahan tersebut sesuai dengan konsep algoritma MNB yang menghitung semua kata yang serupa muncul berapa kali dalam satu dokumen [5], metode MNB juga sesuai untuk melakukan klasifikasi terhadap *text* pendek (*spam detection, topic categorization, sentiment analysis*) dan mempertimbangkan frekuensi kata, untuk mendapatkan kembali akurasi yang lebih baik daripada hanya memeriksa kemunculan kata [8]. Hasil dari penelitian ini yaitu untuk mengetahui klasifikasi dan tingkat akurasi dari analisis *sentiment* yang dilakukan terhadap *dataset movie review* IMDB, serta mengetahui seberapa besar pengaruh *text preprocessing* dalam pengimplemnetasian algoritma, sehingga dengan adanya penelitian ini dapat menjadi referensi untuk penelitian selanjutnya

## II. KAJIAN TEORI

### A. *Text Mining*

*Text Mining* merupakan suatu metode yang bertujuan untuk memproses suatu data yang banyak dan tidak terstruktur secara bentuk dan model, metode ini sangat cocok untuk memproses data yang berbentuk *text*. *Text mining* juga dapat diartikan sebagai suatu proses penambangan, penggalian suatu informasi secara mendalam dalam suatu kumpulan data dengan skala waktu tertentu [8], secara singkat *text mining* dapat diartikan sebagai suatu proses mengumpulkan data kualitas tinggi dari sebuah kumpulan *text* [9].

1. *Text Preprocessing*

   *Text Preprocessing* merupakan tahapan pertama atau tahap awal dari *text mining*, pada tahap ini analisis akan melakukan persiapan terhadap dokumen-dokumen yang akan diteliti, dengan tujuan untuk membuat dokumen tersebut yang awalnya dari data yang tidak terstruktur menjadi data yang [9] Tahapan *preprocessing,* bertujuan untuk membersihkan data-data yang digunakan dari jumlah data yang berganda, sementara *text processing* merupakan tahapan yang dilakukan untuk melakukan pemilihan terhadap data yang diproses dari dokumen yang menjadi data penelitian [11].

2. Tahapan *Text Preprocessing*

   Terdapat 6 (enam) tahap yang dilakukan terhadap proses *text preprocessing* yaitu.

   a. *Handling Duplicate*

      *Handling duplicates* merupakan tahapan pertama dalam proses *text preprocessing,* tahapan ini berfungsi sebagai penyaring sebuah data yang sedang diteliti atau analisis, untuk menghindari kemungkinan data memiliki jumlah lebih dari satu dari sekelompok data yang digunakan.

   *b. Case Folding*

      *Case folding* adalah tahapan ketiga dari *text preprocessing* yang digunakan sebagai metode untuk mengubab data text tersebut kedalam bentuk standar yaitu huruf kecil semuanya.

   *c. Cleaning*

      *Cleaning* merupakan tahap dalam *text preprocessing,* tahapan ini bertujuan untuk menghilangkan komponen-komponen atau variabel yang tidak diinginkan ada adalam proses penelitian.

   *d. Tokenize*

      *Tokenize* merupakan tahapn lanjutan yang bertujuan untuk memisahkan katakata dalam kelompok kata dengan batasan tanda baca atau spasi.

   *e. Stopword Removal*

      *Stopword Removal* merupakan tahapan yang dilakukan dalam *text preprocessing* yang digunakan untuk menghapus berbagai macam kata tertentu berdasarkan *stoplist* yang sudah dibuat.

   *f. Stemming*

      *Stemming* merupakan tahap penguraian kata yang terdapat dalam proses *text preprocessing* yang betujuan untuk menghasilkan kata-kata penelitian dalam bentuk dasar.

### *B. Sentiment Analysis*

*Sentiment Analysis* merupakan suatu metode yang digunakan untuk mengklasifikasi suatu sentiment dari sebuah teks tertentu kemudian dilakukan analisis [12] *Sentiment Analysis* juga dapat diartikan sebagai *opinion mining*, yang merupakan sebuah penambangan data dari sebuah opini atau pendapat, yang terdapat dalam sebuah kalimat, kalimat tersebut dikelompokan dalam bentuk *sentiment* positif dan negatif [9]

1. Sub-proses *Sentiment Analysis*

   sub-proses dari analisis sentimen dibagi dalam 3 (tiga) macam katergori besar yaitu

   a. *Subjectivity classification* merupakan sub-proses yang bertujuan untuk menentukan kelimat yang akan diteliti berupa sebuah opini dari suatu kumpulan data teks.

   b. *Orientatiton detection* merupakan tahap kedua dari sub-proses sentiment analisis yang bertujuan untuk meklasifikasikan opini yang ada kedalam kategori tertentu seperti positif, negative, atau netral.

   c. *Opinion holder and target detection* merupakan penentuan bagian yang merupakan *opinion holder* (pemberi opini) dan bagian yang merupakan target.

### C. TF-IDF

Pembobotan TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan proses yang mengubah data teks menjadi data numerik untuk menghitung bobot kata atau fitur individual.

Tahap proses ini memiliki dua bagian yaitu *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). *Term Frequency* (TF) adalah jumlah kemunculan setiap kata dalam suatu dokumen, semakin banyak muncul kata dalam setiap dokumen maka semakin besar nilai TF. *Inverse Document Frequency* (IDF) adalah jumlah nilai dokumen untuk setiap kata dan berbanding terbalik dengan kata lain, jika sebuah kata jarang muncul dalam sebuah dokumen, maka nilai IDF lebih tinggi dari kata yang sering muncul [10].

Metode ini menghitung bobot setiap token t dalam dokumen d dengan menggunakan rumus:

TF-IDF : $Wdt = tfdt * IDFt$ (1)

Keterangan:

| | | |
|---|---|---|
| d | : | dokumen |
| t | : | Kata kunci |
| W | : | bobot jumlah dokumen d dalam kata ke-t |
| tf | : | jumlah kata yang dicari dalam dokumen |
| IDF | : | *Inverse Document Frequency* |

IDF : $\log2 (D/df)$ (2)

Keterangan

| | | |
|---|---|---|
| D | : | dokumen |
| df | : | Kata kunci |

D. Algoritma *Naïve Bayes*

1. Definisi *Naïve Bayes*

Algoritma *Naïve Bayes* merupakan algoritma yang sering digunakan untuk menemukan nilai suatu probabilitas tertinggi dari suatu klasfikasi data uji yang dilakukan terhadap kategori yang paling tepat [11]. *Naïve bayes* memiliki sifat mudah diterapkan pada kasus-kasus tertentu, serta memiliki tingkat keakurasian yang tinggi terhadap dataset yang sangat besar [12]. *Naïve Bayes Classifier* (NBC) adalah pengklasifikasi probabilistik sederhana yang menerapkan Teorema Bayes dengan asumsi independensi tinggi, ini mengacu pada konsep dasar NBC, khususnya Teorema Bayes yang pertama kali dikemukakan oleh Thomas Bayes kemudian teori *naïve bayes* awalnya metode ini digunakan sebagai metode probabilitas dan statistika, yang bertujuan untuk memperkirakan dimasa depan dengan menggunakan ramalan dari data masa sebelumnya [13].

Berikut ini merupakan penurunan persamaan umum mencari nilai probabilitas sederhana dari algoritma *naïve bayes*.

*Naïve Bayes* : $= P(C|X) = \frac{P(X|C)p(C)}{P(X)}$ (3)

$$= \frac{P(C)\,P(X1|C)\,P(X2|C).....P(Xn|C)}{P(X1,X2,X3.....Xn)}$$

$$= \frac{P(C)\,\prod_{i=1}^{n} P(Xn|C)}{P(X1,X2,X3.....Xn)}$$

$$= P(C)\,\prod_{i=1}^{n} P(Xn|C)$$

Keterangan:

| | | |
|---|---|---|
| X | : | Data Training |
| C | : | Kelas klasifikasi (Positif, Negatif dan Netral) |
| P(X\|C) | : | Peluang kejadian data menjadi sesuai kelasnya (*prior probability)* |
| P(C) | : | Probabilitas kelas klasifikasi (*prior probability)* |
| P(C\|X) | : | Peluang kelas klasifikasi pada data |
| P(X) | : | Peluang kemunculan data tersebut |
| n | : | Jumlah term dalam satu kelas tertentu |

2. Jenis-jenis *Naïve Bayes*

*Naïve Bayes Classifier* terbagi menjadi 3 jenis diantaranya sebagai berikut.

a. *Multinomial Naïve Bayes*

*Multinomial Naïve Bayes* adalah merupakan metode yang digunakan dalam *naïve bayes* untuk melihat suatu probabilitas keadaan tertentu. Probabilitas tersebut didapatkan dengan menghitung suatu frekuensi kemunculan kata dalam satu kelas tertentu. Probabilitas yang dilakukan tanpa memperhitungkan urutan kata atau informasi yang terdapat dalam kalimat atau dokumen secara umum. Berikut ini merupakan penurunan rumus dari *multinomial naïve bayes* yang dapat dilihat di bawah ini.

*Naïve Bayes Multinomial* : $= P(Xn|C) = \frac{Nk+1}{|V|+N\prime}$ (4)

*Keterangan* :

| | | |
|---|---|---|
| $\|V\|$ | : | Jumlah seluruh term atau kata unik |
| *Nk* | : | Jumlah Kemunculan tk dalam dalam data yang dilatih dikategori kelas yang sama |
| *N'* | : | Jumlah total *term* yang terdapat pada kategori c dalam data yang dilatih |
| *1* | : | Merupakan penambahan angka *laplace smoothing* |

b. *Bernoulli Naïve Bayes*

*Bernoulli Naïve Bayes* merupakan salah satu jenis dari algoritma naïve bayes yang digunakan untuk melakukan pelatihan dan pengklasifikasian yang bersifat *bolean,* biasanya pada *bernoulli baïve bayes* menggunakan bilangan *binary* (0 dan 1). Berikut ini merupakan hasil turunan rumus *naïve bayes bernoulli* yang dapat dilihat di bawah ini.

*Naïve Bayes Bernoulli* : $= P(Xn|C) = \frac{dfi+1}{df\prime+2}$ (5)

*Keterangan* :

| | | |
|---|---|---|
| *df'* | : | Jumlah seluruh data pada kategori c |
| *dfi* | : | Jumlah data yang dilatih pada kategori c yang mengandung term |
| 1 dan 2 | : | Angka *Laplace Smooting* supaya tidak *zero probability* |

c. *Gaussian Naïve Bayes*

*Gaussian Naïve Bayes* merupakan salah satu metode yang digunakan dalam algoritma *naïve bayes*, yang berfungsi untuk mengklasifikasikan data secara kontinu atau berkelanjutan. berikut rumus yang digunakan dalam *gaussian naïve bayes.*

*Naïve Bayes Gaussian* : $= P(Xn|C) = \frac{1}{\sqrt{2\pi\sigma_y^2}}\exp\left(-\frac{(Xi-\mu_y)^2}{2\pi\sigma_y^2}\right)$ (6)

*Keterangan* :

| | | |
|---|---|---|
| $\mu_y$ | : | Semua data latih yang menjadi milik kelas c |
| $\sigma_y^2$ | : | Varian sampel ($s^2$) dari data latih |

*E. Confusion Matrix*

*Confusion Matrix* merupakan sebuah tabel yang terdiri atas banyaknya baris data uji yang diprediksi benar atau tidak benar oleh model klasifikasi, tabel ini diperlukan untuk menentukan kinerja suatu model klasifikasi. Metode ini sering digunakan dengan kasus *multiple classifiers* atau kelas yang lebih dari dua. Maka dari itu metode ini cocok untuk digunakan dalam penelitian ini guna mengukur seberapa akurat hasil klasifikasi dari model yang telah dibuat [14]. Penggunaan dari *confusion matrix* dijelaskan pada Tabel 1 sebagai berikut.

TABEL 1
*CONFUSION MATRIX*

| | *Predicted Positive* | *Predicted Negative* |
|---|---|---|
| *Actual Positive* | (TP) | (FN) |
| *Actual Negative* | (FP) | (TN) |

keterangan:

1. *True Negative* (TN) = seberapa banyak jumlah *dataset* yang dikelompokkan *negatif* bernilai *negatif*.
2. *True Positive* (TP) = seberapa banyak jumlah *dataset* yang dikelompokkan positif bernilai positif.
3. *False Negative* (FN) = Seberapa banyak jumlah *dataset* yang dikelompokkan positif bernilai negatif.
4. *False Positive* (FP) = seberapa banyak jumlah *dataset* yang dikelompokkan *negatif* bernilai *positif*.

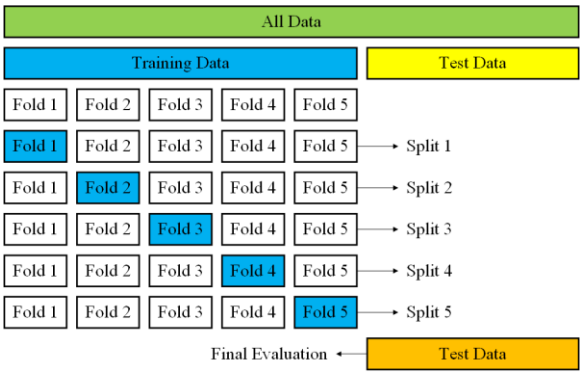
Data yang telah dihasilkan dapat mencari parameter seperti *Accuracy, Precision, Recall* dan F1-*Score*, adapun rumus yang digunakan dalam mengukur parameter yang digunakan dalam *confusion matrix*, yang dapat dilihat pada Tabel 2 di bawah ini.

TABEL 2
RUMUS PARAMETER *CONFUSION MATRIX*

| **Nama** | **:** | **Rumus** | |
|---|---|---|---|
| *Accuracy* | : | $\frac{TP+TN}{TP+TN+FP+FN}$ | (6) |
| *Precision* | : | $\frac{TP}{TP+FP}$ | (7) |
| *Recall* | : | $\frac{TP}{TP+FN}$ | (8) |
| *F1-Score* | : | $\frac{2(recall \times precision)}{recall+precision}$ | (9) |

F. K-Fold Cross Validation

*K-Fold Cross Validation* di jalankan untuk memilih model data dengan akurasi data terbaik dan tertinggi untuk semua data. *k-fold cross validation* dipilih untuk memisahkan kumpulan data dengan angka acak tetapi seimbang tanpa takut bahwa distribusi data mungkin menemukan banyak atau angka yang berbeda.




GAMBAR 1
*K-FOLD CROSS VALIDATION*

Menurut [18] terdapat 4 langkah-langkah dalam penentuan *k-fold cross validation* yaitu:

1. Data akan dibagi kedalam beberapa jumlah k bagian
2. *Fold* pertama merupakan bagian pertam yang akan menjadi data yang akan diuji, sedangkan data lainnya akan menjadi data yang dilatih, setelah itu dilakukan perhitungan akurasi atau kedekatan suatu hasil pengukuran berdasarkan porsi data yang telah ditentukan, berikut ini persamaan yang digunakan dalam menghitung akurasi.

$$\text{Akurasi} \quad \frac{\sum data\ uji\ benar\ klasifikasi}{\sum total\ data\ uji} x\ 100 \quad (10)$$

3. *Fold* yang kedua merupakan bagian dari data testing yang kedua, dan sisanya menjadi data training atau data yang dilatih, setelah itu dilakukan kembali perhitungan akurasi terhadap proporsi data pada *fold* kedua tersebut.
4. Cara ini akan terus berlangsung hingga mencapai *fold* ke-k, setelah iterasi sudah tercukupi maka rata-rata dari hasil akurasi tersebu akan menjadi akurasi yang digunakan.

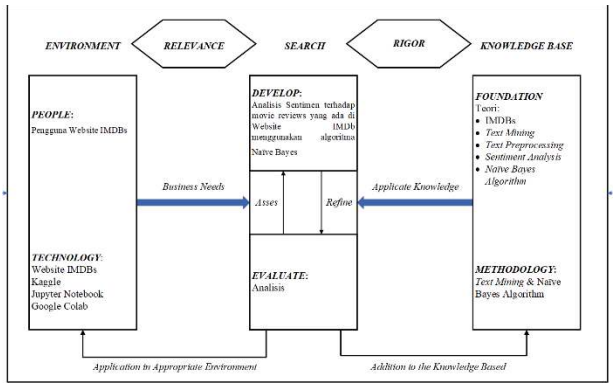
*G. Wordcloud*

*Wordcloud* merupakan teknik visualisasi yang terdiri dari kumpulan kata yang paling sering muncul saat menganalisa kumpulan data menjadi sebuah tampilan yang menarik. Kata-kata yang terdapat pada *wordcloud* merupakan kata kata yang paling sering digunakan dalam dataset. Semakin besar ukuran kata maka semakin sering menonjol kata tersebut [16].

## III. METODE

A. Konseptual Model

Konseptual model merupakan serangkaian tahapan kerja, sistem atau diagram yang menyajikan seperangkat gagasan mengenai partisipasi individu atau peristiwa dalam ilmu pengetahuan dan perkembangannya. Konseptual model dibuat berdasarkan atas dasar teori yang ada pada makalah penelitian sebelumnya [17].




GAMBAR 2
MODEL KONSEPTUAL PENELITIAN

Berdasarkan gambar diatas, dapat dilihat dalam lingkungan penelitian terdapat pengguna *website* IMDB yang melakukan *movie review* sebagai ulasan atau komentar terhadap film yang telah dilihat, dataset dari *movie review* pada *website* IMDB diambil dari situs Kaggle yang berfungsi sebagai tempat kumpulan dari berbagai *dataset* yang dapat digunakan. Teori dasar yang digunakan dalam penelitian ini berdasarkan analisis sentimen, *text processing*, *text mining*, dan penggunaan algoritma *naïve bayes*. Ulasan atau komentar diproses menggunakan salah satu dari *text mining* yaitu Analisis Sentimen dengan beberapa prosedur didalamnya, dan dilanjutkan dengan mengukur tingkat keakuratan menggunakan *text mining* dan algoritma *naïve bayes*.

B. Pengumpulan Data

*Dataset* yang digunakan dalam penelitian ini merupakan *dataset* yang di dalamnya terdapat 50 ribu *movie review* dari *website* IMDB yang telah dikumpulkan dalam satu *file csv*, file tersebut di dapat dari situs Kaggle yang berfungsi sebagai tempat *database opensource*, sehingga siapapun dapat mengunduhnya. Data yang digunakan dalam penelitian ini merupakan kumpulan komentar dari seluruh film pada tahun 2019.

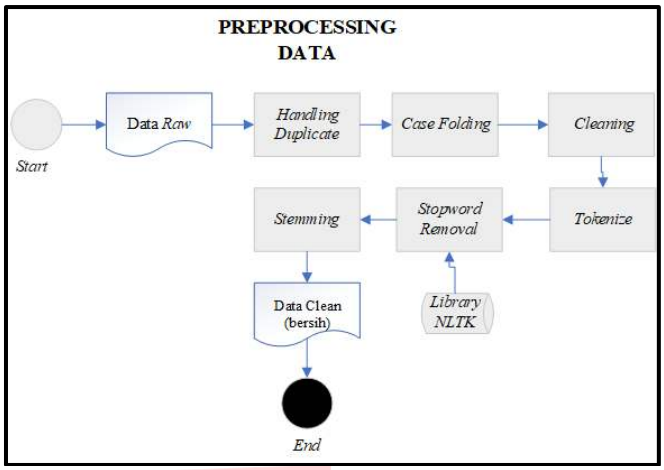
C. Metode Evaluasi

Algoritma *naïve bayes* merupakan metode utama yang digunakan dalam penelitian ini, penggunaan metode ini bertujuan untuk melihat tingkat akurasi yang dihasilkan oleh algoritma *naïve bayes* terhadap *dataset movie review* dari situs Kaggle. Sehingga untuk meningkatkan nilai akurasi dari algoritma *naïve bayes* diperlukan metode tambahan lainnya seperti *text preprocessing*, *confusion matrix*, dan *k-fold cross validation*.

## IV. HASIL DAN PEMBAHASAN

A. *Text Preprocessing*

Tahapan *preprocessing* terdiri dari beberapa jenis diantaranya yaitu *handling duplicate, case folding, cleaning, tokenize, stopword removal, stemming*. Berikut ini merupakan hasil tahapan-tahapan yang dilakukan dalam text preprocessing yang dapat dilihat pada Gambar di bawah ini.




GAMBAR 3
*TEXT PREPROCESSING*

1. *Handling Duplicate*

TABEL 3
*HANDLING DUPLICATE*

| Hasil Codingan *Handling Duplicate* | |
|---|---|
| *Before* | `Out[3]: 50000` |
| *After* | `Out[4]: 49582` |

Hasil tahap *handling duplicate* ini adalah menghilangkan *movie review* yang sama dari *dataset* yang digunakan, berdasarkan tabel di atas dapat dilihat, sebelum melakukan *handling duplicate* terdapat 50 ribu komentar pada *dataset* tersebut, namun setelah dilakukan *handling duplicate* sisa komentar yang tersisa adalah 49.582 setelah dilakukan penghapusan terhadap 418 data yang sama, karena data yang bersifat sama dihilangkan dari *dataset* yang digunakan, dengan perintah "df.drop_duplicates( )".

2. *Case Folding*

TABEL 4
*CASE FOLDING*

| *Review* | *Case Folding* |
|---|---|
| *A rating of "1" does not begin to express how dull, depressing and relentlessly bad this movie is.* | *a rating of "1" does not begin to express how dull, depressing and relentlessly bad this movie is.* |
| *The only thing serious about this movie is the humor. Well worth the rental price. I'll bet you watch it twice. It's obvious that Sutherland enjoyed his role.* | *the only thing serious about this movie is the humor. well worth the rental price. i'll bet you watch it twice. it's obvious that sutherland enjoyed his role.* |

Hasil yang didapat dari tahap ini dilihat pada kolom *review dataset* yang digunakan masih memiliki huruf kapital di awal kalimat, setelah dilakukan proses *case folding*, data yang terbaru dimasukan kedalam kolom baru yang bernama *Case Folding*, pada kolom tersebut dapat dilihat bahwa *dataset* sudah dalam huruf kecil semuanya.

3. *Cleaning*

Tahap *cleaning* pertama dilakukan untuk menghapus karakter, pada dataset *movie review* IMDb, lalu data yang sudah dihilangkan karakter tersebut disimpan ke dalam kolom baru yang bernama remove_chr. Tahap kedua

dilakukan penghapusan kata yang paling sering muncul dalam *dataset*, pada tahap ini terlebih dahulu dilakukan pencarian kata yang sering muncul, setelah kata-kata tersebut didapatkan maka dilanjutkan dengan kodingan penghapusan kata-kata tersebut, lalu hasil dari tahap ini akan disimpan kedalam kolom baru dengan nama *"freqwords"*. Tahap terakhir ini dilakukan penghapusan terhadap kata berulang, setelah proses ini selesai dijalankan maka disimpan ke dalam kolom baru yang bernama *cleaning*. Adapun hasil dari tahap *cleaning* yang dapat dilihat sebagai berikut.

| remove_chr | freqwords | cleaning |
|---|---|---|
| one of the other reviewers has mentioned that ... | one other reviewers has mentioned after watchi... | one other reviewers has mentioned after watchi... |
| a wonderful little production <br ><br >the fi... | wonderful little production <br ><br >the film... | wonderful litle production br br the filming t... |
| i thought this was a wonderful way to spend ti... | thought was wonderful way spend time on too ho... | thought was wonderful way spend time on to hot... |
| basically theres a family where a little boy j... | basically theres family where little boy jake ... | basicaly theres family where litle boy jake th... |

GAMBAR 4
HASIL TAHAP *CLEANING*

4. *Tokenize*

Tahap *tokenize* bertujuan untuk memisahkan kata per kata dari dataset *movie review* IMDb, pada tahap ini hasil dari *tokenize* akan disimpan kedalam kolom yang baru dengan label 'tokenize', setiap kata yang sudah di *tokenize* akan dipisahkan menggunakan tanda koma dan spasi. Berikut merupakan hasil tahap *tokenize* dari beberapa *review*.

| cleaning | tokenize |
|---|---|
| one other reviewers has mentioned after watchi... | [one, other, reviewers, has, mentioned, after,... |
| wonderful litle production br br the filming t... | [wonderful, litle, production, br, br, the, fi... |
| thought was wonderful way spend time on to hot... | [thought, was, wonderful, way, spend, time, on... |
| basicaly theres family where litle boy jake th... | [basicaly, theres, family, where, litle, boy, ... |

GAMBAR 5
HASIL TAHAP TOKENIZE

5. *Stopword*

| tokenize | stopword |
|---|---|
| [one, other, reviewers, has, mentioned, after,... | [reviewers, mentioned, watching, episode, youl... |
| [wonderful, litle, production, br, br, the, fi... | [wonderful, litle, production, filming, techni... |
| [thought, was, wonderful, way, spend, time, on... | [thought, wonderful, way, spend, time, hot, su... |
| [basicaly, theres, family, where, litle, boy, ... | [basicaly, theres, family, litle, boy, jake, t... |

GAMBAR 6
HASIL TAHAP STOPWORD

Hasil dari *stopword* dapat terlihat dengan jelas pada gambar diatas, pada tahap *tokenize* masih terdapat beberapa kata yang tidak penting seperti "one, of, the, a, etc", namun setelah dilakukan proses *stopword*, kata-kata tersebut sudah tidak ada, sehingga dapat dilanjutkan dengan tahap *stemming*

6. *Stemming*

| stopword | stemming |
|---|---|
| [reviewers, mentioned, watching, episode, youl... | [review, mention, watch, episod, youl, hoke, r... |
| [wonderful, litle, production, filming, techni... | [wonder, litl, product, film, techniqu, unasum... |
| [thought, wonderful, way, spend, time, hot, su... | [thought, wonder, way, spend, time, hot, sumer... |
| [basicaly, theres, family, litle, boy, jake, t... | [basicali, there, famili, litl, boy, jake, thi... |

GAMBAR 7
HASIL TAHAP STEMMING

Hasil yang diharapkan dari tahap *stemming* adalah imbuhan yang terdapat disetiap kata dapat dihilangkan, pada kolom *stemming* dapat dilihat beberapa kata yang awalnya memiliki imbuhan "e, es, ed, ly", setelah dilakukan proses *stemming* tidak memiliki imbuhan lagi, hal ini dapat dikatakan berhasil.

## V. ANALISIS

Implementasi algoritma *naïve bayes* pada penelitian ini menggunakan data yang sudah melalui tahap *text preprocessing*, data tersebut dilakukan pembagian data atau *split data* menjadi data *training* dan data *testing* dalam 3 (tiga) jenis perbandingan yaitu 80:20, 70:30, dan 60:40, adapun jumlah data dari setiap perbandingan dapat dilihat sebagai berikut.

TABEL 5
PERBANDINGAN PENGGUNAAN DATA TESTING DAN DATA TRAININg

| Perbandingan | Data Training | Data Testing | Total |
|---|---|---|---|
| 80:20 | 39517 | 9879 | 49396 |
| 70:30 | 34577 | 14819 | 49396 |
| 60:40 | 29638 | 19758 | 49396 |

Perbandingan data tersebut akan dilakukan pengimplementasian terhadap algoritma *naïve bayes* untuk dilakukan pencarian tingkat akurasi dari masing-masing perbandingan data tersebut, namun sebelum proses tersebut dilakukan TF-IDF.

Proses TF-IDF merupakan tahapan yang dilakukan dalam text processing, tahapan ini melakukan pembobotan setiap kata yang terkandung dalam dataset yang digunakan dalam penelitian ini, proses ini memiliki kemiripan terhadap contoh proses TF-IDF manual yang telah dijabarkan pada point bab IV.4.1, namun perbedaannya adalah proses TF-IDF dataset ini dilakukan oleh *machine learning* menggunakan perintah "TfidfVectorizer( )", hal ini dikarenakan banyaknya data yang akan diolah sehingga tidak memungkinkan untuk dilakukan secara manual. Hasil dari proses ini akan disimpan dan diubah kedalam bentuk matix, namun karena data yang sangat banyak, sehingga machine learning tidak dapat menampilkan secara keseluruhan. Berikut ini merupakan tampilan hasil TF-IDF yang dilakukan menggunakan bantuan *machine learning* dapat dilihat pada gambar

```
<49396x102047 sparse matrix of type '<class 'numpy.float64'>'
        with 4528772 stored elements in Compressed Sparse Row format>
```

GAMBAR 8
HASIL TF-IDF

Data hasil TF-IDF tersebut dilanjutkan dengan proses hasil tingkat akurasi dari masing-masing perbandingan yang digunakan dalam pengimplementasian *naïve bayes* yang dapat dilihat sebagai berikut.

TABEL 6
HASIL PERBANDINGAN TINGKAT AKURASI

| Perbandingan | Tingkat Akurasi |
|---|---|
| 80:20 | 88.81% |
| 70:30 | 88.93% |
| 60:40 | 88.51% |

Hasil implemnetasian algoritma naïve bayes tersebut dievaluasi menggunakan *confusion matrix,* yang dapat dilihat sebagai berikut.

TABEL 7
HASIL PENGUJIAN CONFUSION MATRIX

| | *Predicted Positive* | *Predicted Negative* |
|---|---|---|
| *Actual Positive* | 6710 | 817 |
| *Actual Negative* | 823 | 6469 |

Berdasarkan tabel di atas dapat dilihat bahwa jumlah *movie review* bernilai positif yang benar diprediksi oleh sistem atau bernilai *true positive* sebanyak 6710 sedangkan untuk jumlah *review* yang bernilai *positive* yang salah diprediksi oleh sistem atau bernilai *false positive* berjumlah 823 *review*, dan dapat dilihat juga untuk *review* yang bernilai *negative* yang diprediksi benar oleh sistem berjumlah atau *true negative* berjumlah 6469 dan *review* yang bernilai negatif yang diprediksi salah oleh sistem atau *false negative* berjumlah 817 *review*.

Hasil data yang telah didapat pada *confusion matrix* dapat dihitung tingkat akurasi, *precision*, *recall*, dan F1-*Score* secara manual menggunakan rumus sebagai berikut.

$$\text{Akurasi/ } Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$= \frac{6469+6710}{6469+6710+823+817}$$

$$= \frac{13179}{14819}$$

$$= 0.8893$$

$$\text{Presisi/}Precision = \frac{TP}{TP+FP} \quad (5)$$

$$= \frac{6710}{6710+823}$$

$$= \frac{6710}{7533}$$

$$= 0.8907$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

$$= \frac{6710}{6710+817}$$

$$= \frac{6710}{7527}$$

$$= 0.8914$$

$$f1-Score = \frac{2(recall \times precision)}{recall+precision} \quad (7)$$

$$= \frac{2(0.8914 \times 0.8907)}{0.8914+0.8907}$$

$$= \frac{1,588126}{1,7821}$$

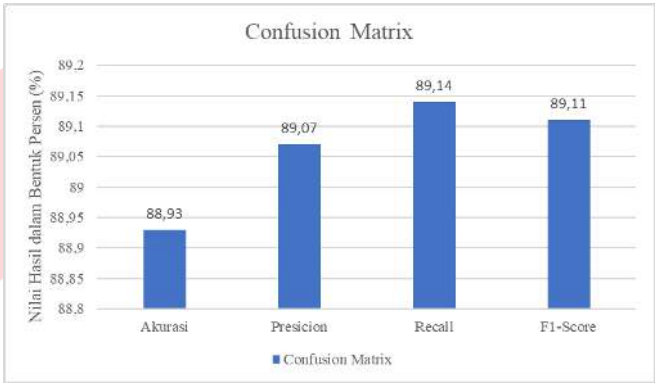
$$= 0.8911$$

Hasil-hasil diatas tersebut akan dirangkum dalam Tabel 11 di bawah ini.

TABEL 8
HASIL *CONFUSION MATRIX*

| Confusion Matrix | Hasil |
|---|---|
| Akurasi/*Accuracy* | 0.8893 |
| Presisi/Precision | 0.8907 |
| *Recall* | 0.8914 |
| F1-*Score* | 0.8911 |

Hasil confusion matrix tersebut dapat dikonversikan kedalam bentuk histogram yang dapat dilihat pada Gambar di bawah ini.



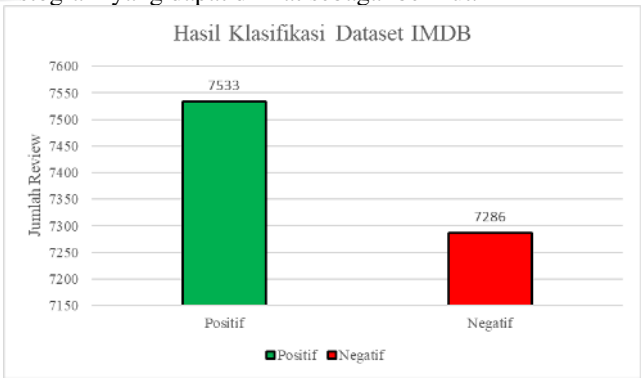

GAMBAR 9
HASIL CONFUSION MATRIX

Berdasarkan hasil klasifikasi yang dilakukan menggunakan algoritma *Naïve Bayes Multinomial* didapatkan bahwa, analisis sentimen yang dilakukan dengan menggunakan *split data* perbandingan 70:30 dengan menggunakan data *testing* menghasilkan tingkat akurasi sebesar 88.93% dengan total data *review* sebanyak 7533 berlabel positif dan 7286 berlabel negatif, hasil data klasifikasi tersebut menunjukan sebagian besar pengguna *website* IMDB memberikan masukan yang positif terhadap film-film yang mereka tonton. Hasil klasifikasi dari *dataset* analisis sentimen IMDB dapat dilihat sebagai berikut.

TABEL 12
HASIL KLASIFIKASI *NAIVE BAYES* PERBANDINGAN DATA 70:30

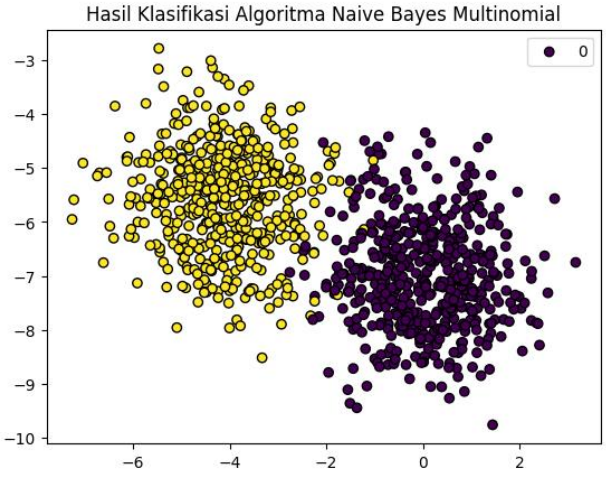
| Keterangan | Jumlah |
|---|---|
| Positif | 7533 |
| Negatif | 7286 |
| Jumlah | 14819 |

Data hasil klasifikasi tersebut dibuat dalam bentuk histogram yang dapat dilihat sebagai berikut.




GAMBAR 10
HASIL KLASIFIKASI DATASET REVIEW FILM DI IMDB

Hasil persebaran klasifikasi yang dilakukan menggunakan algoritma Naïve Bayes di visualisasikan menggunakan diagram scatter yang dapat dilihat pada Gambar 11 di bawah ini.



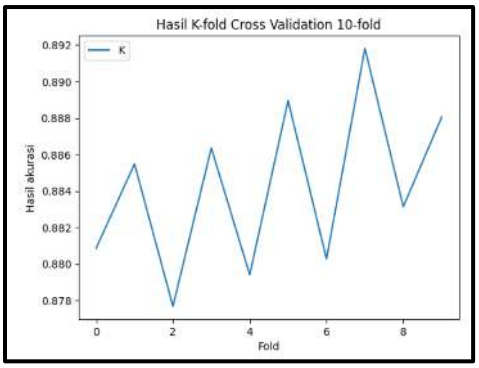

GAMBAR 11
HASIL PERSEBARAN KLASIFIKASI DALAM BENTUK SCATTER

Model pada penelitian ini akan dilakukan uji validasi menggunakan metode *k-fold cross validation*, perbandingan data yang dilakukan pengujian dengan nilai 10-fold, yang artinya perbandingan tersebut akan membagi data ke dalam 10 (sepuluh) bagian, lalu dilakukan pengujian sebanyak 10, setiap bagian akan bergantian menjadi data *testing* dan data *training* hingga 10 iterasi. berikut ini merupakan hasil pengujian yang dilakukan menggunakan 10-fold dapat dilihat sebagai berikut.

TABEL 9
HASIL PENGUJIAN 10-FOLD

| Pengujian | Hasil *Score* |
|---|---|
| 1 | 0.88085599 |
| 2 | 0.88548294 |
| 3 | 0.87767496 |
| 4 | 0.88635049 |
| 5 | 0.87941006 |
| 6 | 0.88895315 |
| 7 | 0.88027762 |
| 8 | 0.89181371 |
| 9 | 0.88313567 |
| 10 | 0.88805323 |
| Rata-rata | 0.8842007809 |

Berdasarkan tabel di atas dapat dilihat bahwa nilai akurasi yang tertinggi pada pengujian 8 (delapan) dengan nilai akurasi sebesar 0.89181371, dan yang paling terkecil pada pengujian 3 (tiga) dengan nilai 0.87767496. berdasarkan hasil 10 kali iterasi yang telah dilakukan terhadap dataset IMDB mneghasilkan nilai rata-rata sebesar 0.8842007809 atau 88.42%, dan hasil-hasil tersebut dikonversikan kedalam sebuah grafik garis yang dapat dilihat sebagai berikut.




GAMBAR 12
HASIL K-FOLD CROSS VALIDATION

Visualisasi yang dilakukan dalam penelitian ini menggunakan *wordcloud, wordcloud* merupakan tahap terakhir dari proses *processing*, yang berfungsi untuk menampilkan kata-kata yang paling sering muncul dari masing masing klasifikasi *sentiment analysis* positif dan negatif, berikut ini merupakan hasil *wordcloud* pada *sentiment* positif dan *sentiment* negatif yang digunakan dalam penelitian ini, dapat dilihat pada Gambar 11 dan Gambar 12 di bawah ini.


GAMBAR 13
SENTIMENT POSITIF

Hasil *wordcloud* di atas menunjukan bahwa, pada klasifikasi sentimen analisis positif kata yang sering muncul adalah *Absolute, Love, Time, Show*. Kata "*Absolute"* banyak digunakan dalam sentiment positif karena para pengguna ingin menunjukan bahwa mereka sangat menyukai film atau unsur film tersebut dengan menggunakan kata "*Absolutely"* (benar-benar) yang memiliki kata dasar "*Absolute"*. Perubahan ini terjadi karena pada saat melakukan proses *text preprocessing*, *dataset* yang digunakan melewati *proses stemming* yang bertujuan menghilangkan imbuhan dari setiap, salah satu contoh kalimat *review* yang mengandung kata *"Absolute" sentiment* positif adalah "*What an absolutely stunning movie, if you have 2.5 hrs to kill, watch it, you won't regret it, it's too much fun! Rajnikanth carries the movie on his shoulders and although there isn't anything more other than him, I still liked it. The music by A.R.Rehman takes time to grow on you but after you heard it a few times, you really start liking it.".*


GAMBAR 14
SENTIMENT NEGATIF

Kata "*Absolute"* juga sering ditemukan dalam sentiment negatif, jika kata *absolute* digunakan dalam sentiment positif untuk menunjukan bahwa pengguna sangat menyukai film tersebut, pada *sentiment* negatif kata "*Absolute"* digunakan untuk menunjukan bahwa para penonton benar-benar tidak menyukai film tersebut salah satu contoh kata "*Absolute*" yang *sentiment* negatif adalah "*Furthermore you can't emphasize with any of the characters and as thus, have absolutely no vested interest in them. Technically not an all-together bad movie just an extremely forgettable oneEye Candy: Dara Tomanovich gets topless; Sally Kirkland also shows some skin Where I saw it: Showtime Showcase*".

## VI. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini yaitu. Proses analisis klasifikasi algoritma *naïve bayes* terhadap *review movie* pada *website* IMDB dilakukan dalam beberapa tahapan yaitu tahapan pertama *dataset* yang digunakan akan dilakukan proses *text preprocessing* yang bertujuan untuk membersihkan dan mengolah *dataset* agar siap untuk dilakukan *processing*, kemudian dilanjutkan dengan tahapan perhitungan pembobotan kata menggunakan TF-IDF, setelah nilai bobot didapatkan data tersebut dilakukan klasifikasi menggunakan *naïve bayes*. Nilai akurasi hasil klasifikasi yang didapatkan yaitu dengan rasio 70:30 dengan k-8 dengan nilai akurasi sebesar 0.89181371

Berdasarkan hasil evaluasi menggunakan *confusion matrix* dan *k-fold cross validation* yang didapatkan dari proses *training*, menghasilkan akurasi klasifikasi algoritma *naïve bayes* sebesar 88.93% dengan simulasi menggunakan perbandingan data 70:30, hasil evaluasi *confusion matrix* menunjukan nilai *precision* sebesar 89.07%, nilai *recall* sebesar 89.14% dan nilai F1-*score* sebesar 89.11%. Hasil validasi yang didapatkan dengan pengujian k-10 terdapat nilai rata-rata sebesar 0.8842007809.

Berdasarkan hasil klasifikasi yang dilakukan menunjukan terdapat 7533 data berlabel positif dan 7286 data yang berlabel negatif, hal ini menunjukan mayoritas pengguna IMDB memberikan komentar dan opini yang bersifat positif terhadap film yang mereka tonton.

## REFERENSI

[1] B. Lopez and X. Sumba, "IMDb Sentiment Analysis," pp. 2–6, 2019.

[2] F. Savira and Y. Suharsono, "Analisa dan Klasifikasi Sentiment Opini Penonton pada Website Imdb.Com dengan Algoritma Support Vector Machine," *J. Inform. dan Bisnis*, vol. 6, no. 2, pp. 18–28, 2017.

[3] N. G. Ramadhan and T. I. Ramadhan, "Analysis Sentiment Based on IMDB Aspects from Movie Reviews using SVM," *Sinkron*, vol. 7, no. 1, pp. 39–45, 2022, doi: 10.33395/sinkron.v7i1.11204.

[4] M. A. A. Jihad, Adiwijaya, and W. Astuti, "Analisis sentimen terhadap ulasan film menggunakan algoritma random forest," *e-Proceeding Eng.*, vol. 8, no. 5, pp. 10153–10165, 2021.

[5] A. Singh, C. Kulkarni, and N. A. Ayan, "Sentiment Analysis of IMDB Movie Reviews." 2022.

[6] P. Routray, C. K. Swain, and S. P. Mishra, "A survey on sentiment analysis challenges," *Int. J. Comput. Appl.*, vol. 76, no. 4, pp. 330–338, 2013, doi: 10.1016/j.jksues.2016.04.002.

[7] F. Nurhuda, S. Widya Sihwi, and A. Doewes, "Analisis Sentimen Masyarakat terhadap Calon Presiden Indonesia 2014 berdasarkan Opini dari Twitter Menggunakan Metode Naive Bayes Classifier," *J. Teknol. Inf. ITSmart*, 2016.

[8] A. Rifa'i, H. Sujaini, and D. Prawira, "Sentiment Analysis Objek Wisata Kalimantan Barat Pada Google Maps Menggunakan Metode Naive Bayes," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 3, p. 400, 2021, doi: 10.26418/jp.v7i3.48132.

[9] V. I. Santoso, G. Virginia, and Y. Lukito, "Penerapan Sentiment Analysis Pada Hasil Evaluasi Dosen Dengan Metode Support Vector Machine," *J. Transform.*, vol. 14, no. 2, p. 72, 2017, doi: 10.26623/transformatika.v14i2.439.

[10] T. Winarti, J. Kerami, and S. Arief, "Determining Term on Text Document Clustering using Algorithm of Enhanced Confix Stripping Stemming," *Int. J. Comput. Appl.*, vol. 157, no. 9, pp. 8–13, 2017, doi: 10.5120/ijca2017912761.

[11] D. Rustiana and N. Rahayu, "Analisis sentimen pasar otomotif mobil:," *J. SIMETRIS*, vol. 8, no. 1, pp. 113–120, 2017.

[12] A. Andilala, "Movie Review Sentimen Analisis Dengan Metode Naïve Bayes Base on Feature Selection," *Pseudocode*, vol. 3, no. 1, pp. 1–9, 2016, doi: 10.33369/pseudocode.3.1.1-9.

[13] J. LING, I. P. E. N. KENCANA, and T. B. OKA, "Analisis Sentimen Menggunakan Metode Naïve Bayes Classifier Dengan Seleksi Fitur Chi Square," *E-Jurnal Mat.*, vol. 3, no. 3, p. 92, 2014, doi: 10.24843/mtk.2014.v03.i03.p070.

[14] A. Firmansyah Sulaeman, A. Afif Supianto, and F. Abdurrachman Bachtiar, "Analisis Sentimen Opini Mahasiswa Terhadap Saran Evaluasi Kinerja Dosen Menggunakan TF-IDF dan Support Vector Machine," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.* , vol. 3, no. 6, pp. 5647–5655, 2019.

[15] R. T. Handayanto, "Cross Validation dengan Scikit-Learning Python," *Artificial Neural Network, Python*, 2020.

[16] A. Alamsyah and F. N. Zuhri, "Measuring Public Sentiment Towards Services Level in Online Forum using Naive Bayes Classifier and Word Cloud," *CRS-ForMIND Int. Conf. Work. 2017*, no. October, 2017.

[17] D. Edi and S. Betshani, "Analisis Data dengan MenggunakanERD dan Model Konseptual Data Warehouse," *J. Inform.*, vol. 5, no. 1, pp. 71–85, 2009.