

ANALISIS SENTIMEN MENGGUNAKAN ALGORITMA NAÏVE BAYES PADA REVIEW FILM DI IMDB

Disusun untuk memenuhi tugas besar mata kuliah Pemrosesan Bahasa Alami

Dosen Pengampu: Nelly Indriani Widiastuti, M.T.



Disusun oleh:

Yohana Sri Rejeki – IF-2 - 10119044

Caraka Muhamad Rahman – IF-2 - 10119058

NLP-1

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
2023**

BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Sentiment Analysis atau Analisis Sentimen adalah bidang dalam ilmu komputer yang bertujuan untuk mengidentifikasi dan mengklasifikasikan sentimen atau pendapat yang terkandung dalam teks, baik itu dalam bentuk positif, negatif, atau netral. Dalam beberapa tahun terakhir, *Sentiment Analysis* telah menjadi topik yang menarik dalam penelitian dan aplikasi di berbagai industri, seperti media sosial, *e-commerce*, dan survei pengguna.

Metode Naïve Bayes adalah salah satu pendekatan yang umum digunakan dalam *Sentiment Analysis*. Metode ini didasarkan pada Teorema Bayes dan mengasumsikan bahwa setiap fitur atau kata dalam teks adalah independen dari yang lainnya. Dalam metode ini, teks yang akan dianalisis dipecah menjadi fitur-fitur, seperti kata-kata atau frase, dan kemudian diproses untuk melihat seberapa sering fitur-fitur tersebut muncul dalam kategori sentimen yang berbeda.

Seringkali, sebelum menonton suatu film, penonton akan mencari ulasan terkait film tersebut, dan setelahnya, akan menuliskan ulasan terkait film tersebut. Tidak hanya penonton, melainkan berbagai pemangku kepentingan pun menggunakan ulasan yang telah ditulis oleh para penonton setelah menonton film sebagai acuan terhadap performa dari film tersebut. Oleh karena itu, pada penelitian ini akan dilakukan implementasi algoritma Naive Bayes pada Review Film di IMDB.

1.2 Rumusan Masalah

Berdasarkan pemaparan latar belakang tersebut, disimpulkan suatu rumusan masalah yaitu: Apakah dengan menggunakan algoritma Naive Baiyes dapat melakukan analisis sentimen pada review film di IMDB?

1.3 Maksud dan Tujuan

Maksud dari penelitian ini adalah menerapkan algoritma Naive Baiyes untuk melakukan analisis sentimen pada review film di IMDB. Adapun tujuan dari penelitian ini adalah membantu para pemangku kepentingan untuk melihat performa film yang telah ditayangkan serta membantu untuk memproduksi film yang jauh lebih baik berdasarkan performa film yang didapatkan di masa kini.

1.4 Batasan Masalah

Adapun batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Algoritma yang digunakan adalah algoritma pengklasifikasian bernama Naive Bayes.
2. Dataset yang digunakan bersumber dari Kaggle, dengan judul IMDB Dataset of 50K Movie Reviews.
3. Total Dataset yang digunakan pada penelitian berjumlah 25.000.

1.5 Metodologi Penelitian

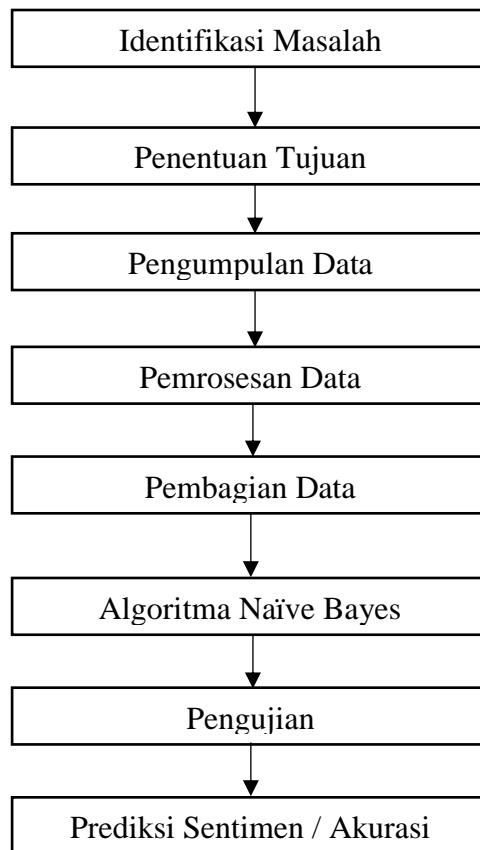
Adapun metodologi penelitian yang digunakan pada penelitian ini adalah

A. Studi Pustaka / *Literature Review*

Metode studi pustaka atau *literature review* dilakukan dengan cara mengumpulkan macam-macam sumber referensi yang memuat banyak teori yang dapat menunjang penelitian.

B. *Data Science Life-Cycle*

Dalam metode *Data Science Life-Cycle* terdapat 8 tahapan yang disesuaikan dengan studi kasus seperti pada gambar di bawah ini:



Gambar 1.1 Data Science Life-Cycle

Adapun penjelasan dari alur tersebut adalah sebagai berikut:

1. Identifikasi Masalah

Mengamati dan menemukan permasalahan pada hasil review film di IMDB.

2. Penentuan Tujuan

Tujuan dari penelitian ini adalah membantu para pemangku kepentingan untuk melihat performa film yang telah ditayangkan serta membantu untuk memproduksi film yang jauh lebih baik berdasarkan performa film yang didapatkan di masa kini.

3. Pengumpulan Data

Pengumpulan data diperoleh dengan cara mengunduh dataset yang tersedia di platform Kaggle yang berjudul IMDB Dataset of 50K Movie Reviews. Keluaran dari sistem ini adalah berupa data testing dengan nilai output berupa sentimen positif dan negatif berdasarkan hasil klasifikasi pada data *training*.

4. Pemrosesan Data

Pemrosesan data merupakan salah satu proses yang penting dalam *data science life-cycle*. Dalam tahapan pemrosesan data terbagi menjadi beberapa tahap, antara lain:

- a. *Cleaning*: Merupakan tahapan pembersihan data dengan melakukan case folding, menghapus karakter selain huruf, menghapus kata dengan karakter/symbol khusus, serta menghapus URL atau link dari setiap komentar.
- b. *Filtering*: Pembuangan kata-kata yang tidak penting. Stopword dilakukan apabila kata-kata pada ulasan film mengandung kata-kata yang sering keluar dan dianggap sebagai kata yang tidak penting, seperti kata penghubung, waktu, dan lain sebagainya.
- c. *Tokenizing*: Setiap kata akan dilakukan pemisalan berdasarkan spasi yang ditemukan.

5. Pembagian Data

Data yang telah dilakukan proses pelabelan positif atau negatif, dilakukan pembagian data. Data latih berjumlah lebih kecil dari data uji.

6. Algoritma Naive Bayes

Algoritma Naive Bayes dapat digunakan untuk data kuantitatif maupun data kualitatif.. Algoritma Naive Bayes dapat digunakan pada kumpulan data yang luas sehingga pada saat proses pemodelan tidaklah rumit. Algoritma ini juga merupakan algoritma yang populer untuk diterapkan pada permasalahan terkait *Natural Language Processing (NLP)*.

7. Pengujian

Setelah diterapkan Algoritma Naive Bayes, akan dilakukan pengujian dengan menggunakan data *testing*.

8. Prediksi Sentimen / Akurasi

Hasil dari pengujian dengan menggunakan data testing pada Algoritma Naive Bayes, akan diukur tingkat akurasinya.

BAB 2

LANDASAN TEORI

2.1 Algoritma Naive Bayes

Algoritma Naive Bayes merupakan salah satu algoritma klasifikasi yang dapat digunakan untuk dokumen teks [1]. Pada Algoritma ini, menggunakan sebuah teorema, bernama Bayes. Teorema ini terdapat pada teori probabilitas dengan menganggap semua atribut memiliki keterhubungan.

Algoritma Naïve Bayes adalah salah satu algoritma klasifikasi yang terkenal. Metode ini diusulkan oleh Thomas Bayes dan digunakan untuk memprediksi peluang di masa depan berdasarkan data di masa sebelumnya. Naïve Bayes atau biasa disebut dengan Multinomial Naïve Bayes merupakan model penyederhanaan dari Metode Bayes yang cocok dalam pengklasifikasian dokumen atau teks [2].

Algoritma Naïve Bayes pun merupakan salah satu algoritma yang paling efektif dan efisien untuk pembelajaran mesin (*Machine Learning*) serta *data mining*. Pada sebuah dataset, dokumen baris i diasumsikan sebagai vektor dari nilai-nilai atribut $\langle x_1, x_2, \dots, x_n \rangle$ di mana setiap nilai-nilai menjadi peninjauan atribut X_i ($i \in [1, n]$). Setiap baris memiliki label kelas $c_i \in \{c_1, c_2, \dots, c_k\}$ sebagai nilai variabel kelas C , sehingga untuk melakukan klasifikasi dapat dihitung nilai probabilitas $p(C = c_i | X = x_j)$. Hal tersebut dikarenakan karena pada Naïve Bayes, diasumsikan setiap atribut saling bebas [3].

Naïve Bayes mengimplementasikan fungsi statistik sederhana berdasarkan Teorema Bayes dengan asumsi bahwa setiap fitur kelas tidak terkait dengan fitur lainnya. Metode Naïve Bayes adalah metode klasifikasi yang menggunakan perhitungan probabilitas.

Persamaan untuk probabilitas *prior* adalah:

$$P(H) = \frac{N_j}{N}$$

di mana N_j adalah jumlah data dalam kelas tertentu, dan N adalah jumlah total data.

Sementara itu, persamaan untuk Teorema Bayes adalah sebagai berikut:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} = P(X|H) \times P(H) / P(X)$$

Keterangan:

X: Data dengan class yang belum diketahui

H: Hipotesis data X yang merupakan suatu class yang lebih spesifik

P(H|X): Probabilitas hipotesis H berdasarkan kondisi X (posteriori probability)

$P(H)$: Probabilitas hipotesis H (prior probability)

$P(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis H

$P(X)$: Probabilitas X

Terdapat lima buah tahapan yang terdapat pada algoritma Naïve Bayes. Tahapan-tahapan tersebut adalah sebagai berikut:

1. Baca data *training*
2. Hitung jumlah *class*
3. Hitung jumlah kasus yang sama dengan *class* yang sama
4. Kaitkan semua nilai hasil sesuai dengan data X yang dicari *class*-nya

Naïve Bayes telah digunakan secara luas dalam berbagai aplikasi klasifikasi, seperti *Sentiment Analysis*, *spam filtering*, dan lainnya. Meskipun asumsi bahwa fitur-fitur adalah independen sering kali tidak realistis, metode Naïve Bayes tetap memberikan hasil yang baik dalam banyak kasus.

2.2 Analisis Sentimen

Analisis Sentimen adalah suatu proses untuk memahami, mengekstrak dan mengolah suatu data untuk mendapatkan informasi sentimen yang terkandung pada suatu kalimat opini. Sedangkan menurut Fanissa “Analisis sentimen adalah bidang interdisipliner, sebuah bidang dimana pendekatan pemecahan masalahnya dengan menggunakan tinjauan dari berbagai sudut pandang ilmu serumpun secara relevan dan terpadu”. Analisis sentimen terdiri dari pemrosesan bahasa, analisis teks dan komputasi linguistik untuk mengidentifikasi suatu sentimen dari dokumen.

Analisis sentimen disebut juga dengan *opinion mining*, sedangkan *opinion mining* bisa dikatakan kombinasi antara *text mining* dan *natural language processing*. *Text mining* memiliki arti penambahan sebuah data yang berupa teks yang bersumber dari sebuah dokumen, tujuan dari *text mining* yaitu mencari kata yang mewakili dari sebuah dokumen sehingga dapat dilakukan sebuah analisis yang terhubung antar dokumen.

Analisis sentimen dapat menggunakan algoritma untuk mengolah dan melakukan klasifikasi terhadap data yang dibangun. Terdapat banyak algoritma yang dapat digunakan dalam penelitian analisis sentimen, antara lain *C4.5*, *The K-Means*, *Support Vector Machine*, *Apriori*, *Maximum Entropy PageRank*, *AdaBoost*, *k-Nearest Neighbor*, *Naïve Bayes*, dan *CART*.

BAB 3

IMPLEMENTASI KASUS

3.1 Perhitungan Kasus

Dataset yang digunakan, bersumber dari Kaggle, dengan judul "IMDB Dataset of 50K Movie Reviews". Total data yang terdapat pada dataset tersebut adalah 50.000 data. Pada pengujian ini, digunakan sekitar 8000 data yang di dalamnya terdapat 3997 review positif dan 4003 review negatif. Metode ini diimplementasikan dengan menggunakan Python. Berikut merupakan contoh perhitungan kasus dengan menggunakan Algoritma Naive Bayes.

Tabel 3.1 Pembagian Data *Training* dan *Testing*

Lambang	Class	Data <i>Training</i>	Data <i>Testing</i>
0	Negatif	3605	392
1	Positif	3605	398
	Total	7210	790

Tabel 3.2 Contoh Kemunculan Term Frekuensi

Kata	Frekuensi Kemunculan Kata (<i>Term Frequency</i>)	
	Review Positif	Review Negatif
agree	137	126
comments	101	97
even	1673	1269
go	2656	2530
step	119	138
nothing	700	311
film	2111	2168
worked	101	99
absolutely	252	199
delmar	1	1

Selanjutnya, dilakukan tahap untuk mencari probabilitas kata *agree*, *comments*, *even*, *go*, *step*, *nothing*, *film*, *worked*, *absolutely*, dan *delmar*.

Diketahui:

nReviewNegatif	= 392
nReviewPositif	= 398
Total seluruh kata	= 58.128

$$P(w|positif|negatif) = \frac{(nk(positif/negatif) + 1}{(n. positif/negatif) + |\tau|}$$

1) Probabilitas kata “agree”

$$P(agree|positif) = \frac{137 + 1}{398 + 58128} = 0,00236$$

$$P(agree|negatif) = \frac{126 + 1}{392 + 58128} = 0,00217$$

2) Probabilitas kata “comments”

$$P(agree|positif) = \frac{101 + 1}{398 + 58128} = 0,00174$$

$$P(agree|negatif) = \frac{97 + 1}{392 + 58128} = 0,00167$$

3) Probabilitas kata “even”

$$P(agree|positif) = \frac{1673 + 1}{398 + 58128} = 0,02861$$

$$P(agree|negatif) = \frac{1269 + 1}{392 + 58128} = 0,02167$$

4) Probabilitas kata “go”

$$P(agree|positif) = \frac{2656 + 1}{398 + 58128} = 0,04540$$

$$P(agree|negatif) = \frac{2530 + 1}{392 + 58128} = 0,04325$$

5) Probabilitas kata “step”

$$P(agree|positif) = \frac{119 + 1}{398 + 58128} = 0,00205$$

$$P(agree|negatif) = \frac{138 + 1}{392 + 58128} = 0,00238$$

6) Probabilitas kata “nothing”

$$P(agree|positif) = \frac{700 + 1}{398 + 58128} = 0,01198$$

$$P(agree|negatif) = \frac{311 + 1}{392 + 58128} = 0,00533$$

7) Probabilitas kata “film”

$$P(agree|positif) = \frac{2111 + 1}{398 + 58128} = 0,03610$$

$$P(\text{agree}|\text{negatif}) = \frac{2168 + 1}{392 + 58128} = 0,03710$$

8) Probabilitas kata “worked”

$$P(\text{agree}|\text{positif}) = \frac{101 + 1}{398 + 58128} = 0,00174$$

$$P(\text{agree}|\text{negatif}) = \frac{99 + 1}{392 + 58128} = 0,00171$$

9) Probabilitas kata “absolutely”

$$P(\text{agree}|\text{positif}) = \frac{252 + 1}{398 + 58128} = 0,00432$$

$$P(\text{agree}|\text{negatif}) = \frac{199 + 1}{392 + 58128} = 0,00342$$

10) Probabilitas kata “delmar”

$$P(\text{agree}|\text{positif}) = \frac{1 + 1}{398 + 58128} = 0,00003$$

$$P(\text{agree}|\text{negatif}) = \frac{1 + 1}{392 + 58128} = 0,00003$$

Setelah mendapatkan hasil dari probabilitas kata, kemudian dilakukan perhitungan probabilitas dari dokumen (*Review*) sampel atau contoh. Dilakukan asumsi $P(V_j)$ (probabilitas kategori dokumen) sama dengan $\text{docs } j$ (jumlah dokumen setiap kategori) dibagi dengan $|\text{contoh}|$ (jumlah dokumen yang digunakan sebagai data training dari seluruh kategori).

$$P(v_j) = \frac{|\text{doc}_j|}{|\text{contoh}|}$$

Perhitungan probabilitas review di mana terdapat 790 review yang terbagi 392 review negatif dan 398 review positif.

$$P(\text{negatif}) = \frac{392}{790} = 0,49620$$

$$P(\text{positif}) = \frac{398}{790} = 0,50380$$

Kata pada review tersebut, didapatkan nilai probabilitas sebagai berikut:

Tabel 3.3 Daftar Uji Kata

No	Kata	Probabilitas Positif V1	Probabilitas Negatif V2
1	agree	0,00236	0,00217
2	comments	0,00174	0,00167
3	even	0,02861	0,02167
4	go	0,04540	0,04325
5	step	0,00205	0,00238
6	nothing	0,01198	0,00533
7	film	0,03610	0,03710
8	worked	0,00174	0,00171
9	absolutely	0,00432	0,00342
10	delmar	0,00003	0,00003

Untuk mendapatkan nilai probabilitas tertinggi, langkah pertama yakni menghitung $P(W_k|V_j)P(V_j)$. Hasil perhitungan $P(V_j)$ dapat dilihat probabilitas dokumen:

Kata (*agree|positif*) : $0,00236 * 0,50380 = 0,00119$

Kata (*agree|negatif*) : $0,00217 * 0,49620 = 0,00108$

Kata (*comments|positif*) : $0,00174 * 0,50380 = 0,00088$

Kata (*comments|negatif*) : $0,00167 * 0,49620 = 0,00083$

Kata (*even|positif*) : $0,02861 * 0,50380 = 0,01441$

Kata (*even|negatif*) : $0,02167 * 0,49620 = 0,01075$

Kata (*go|positif*) : $0,04540 * 0,50380 = 0,02287$

Kata (*go|negatif*) : $0,04325 * 0,49620 = 0,02146$

Kata (*step|positif*) : $0,00205 * 0,50380 = 0,00103$

Kata (*step|negatif*) : $0,00238 * 0,49620 = 0,00118$

Kata (*nothing|positif*) : $0,01198 * 0,50380 = 0,00604$

Kata (*nothing|negatif*) : $0,00533 * 0,49620 = 0,00264$

Kata (*film|positif*) : $0,03610 * 0,50380 = 0,01819$

Kata (*film|negatif*) : $0,03710 * 0,49620 = 0,01841$

Kata (*worked|positif*) : $0,00174 * 0,50380 = 0,00087$

$$\text{Kata (worked|negatif)} : 0,00171 * 0,49620 = 0,00085$$

$$\text{Kata (absolutely|positif)} : 0,00432 * 0,50380 = 0,00218$$

$$\text{Kata (absolutely|negatif)} : 0,00342 * 0,49620 = 0,00170$$

$$\text{Kata (delmar|positif)} : 0,00003 * 0,50380 = 0,00001$$

$$\text{Kata (delmar|negatif)} : 0,00003 * 0,49620 = 0,00001$$

Setelah itu menghitung nilai probabilitas tertinggi dari masing-masing kategori.

$$\begin{aligned} \text{Probabilitas positif tertinggi} &= 0,00119 * 0,00088 * 0,01441 * 0,02287 * 0,00103 * \\ &0,00604 * 0,01819 * 0,00087 * 0,00218 * 0,00001 = 7.40700\text{e-}28 \end{aligned}$$

$$\begin{aligned} \text{Probabilitas negatif tertinggi} &= 0,00108 * 0,00083 * 0,01075 * 0,02146 * 0,00118 \\ &* 0,00264 * 0,01841 * 0,00085 * 0,00170 * 0,00001 = 1.71375\text{e-}28 \end{aligned}$$

Tabel 3.4 Nilai Probabilitas

Probabilitas Positif Tertinggi	Probabilitas Negatif Tertinggi
7.40700e-28	1.71375e-28

Pada **Tabel 3.3**, dapat dilihat hasil dari keseluruhan proses perhitungan probabilitas tertinggi dengan Naive Bayes. Pada tabel tersebut probabilitas negatif memiliki nilai tertinggi. Sehingga dapat dipastikan *review* yang dipilih merupakan dokumen review negatif.

BAB 4 PROGRAM

4.1 Screenshot dan Deskripsi

Setelah dilakukan tahap perhitungan, dilakukan implementasi perhitungan tersebut pada sebuah program. Program dibuat dengan menggunakan bahasa python. Berikut merupakan tampilan screenshot fungsi pada program, disertai deskripsi.

Tabel 4.1 Screenshot Fungsi pada Program beserta Deskripsinya

Screenshot	Nama Fungsi/Metode	Deskripsi
<pre>df = pd.read_csv('IMDB Dataset.csv') df = df[:8000] df.info() df.head()</pre>	Metode pembacaan dataset	Membaca dataset dalam bentuk csv dengan <i>library</i> pandas.
<pre>def case_folding(text): text = text.lower() return text</pre>	Fungsi <i>Case Folding</i>	Mengubah tipe teks menjadi huruf kecil.
<pre>def remove_punctuations_and_html_encoding(text): patterns = '[^\w\s]' patterns2 = 'br' text = re.sub(patterns, '', text) text = re.sub(patterns2, '', text) return text</pre>	Fungsi Regex	Menghilangkan HTML <i>Encoding</i> dan tanda baca.
<pre>def stopword_removal(text): stop_words = set(stopwords.words('english')) words = nltk.word_tokenize(text) filtered_words = [word for word in words if word.casefold() not in stop_words] return ' '.join(filtered_words)</pre>	Fungsi Stopwords	Menghilangkan kata yang tidak memiliki makna berarti.
<pre>df['sentiment'] = df['sentiment'].replace({'positive': 1, 'negative': 0}) df</pre>	Metode Pelabelan	Melabelkan kata yang bernilai positif menjadi 1 dan negatif menjadi 0

<pre>X = df['review'].values y = df['sentiment'].values X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)</pre>	Metode Pemisahan Data <i>Training</i> dan Data <i>Testing</i>	Data yang akan dilatih dan diuji adalah data <i>review</i> dan <i>sentiment</i> dengan perbandingan 90:10 (data latih:data uji)
<pre>word2id = {} counter = 0 for sentence in X_train: tokens = sentence.split() for token in tokens: if token not in word2id: word2id[token] = counter counter += 1</pre>	Metode pemisahan kata	Kata dipisah dan dan banyaknya kata dihitung.
<pre>for index_sen, sentences in enumerate(X_train): for term, index_term in word2id.items(): if term in sentences: X_vec_matrix[index_sen][index_term] += 1</pre>	Metode matriks	Kata yang telah dihitung, dipetakan dalam label positif (0) dan negatif (1) dalam matriks.
<pre>smote = SMOTE()</pre>	Metode <i>Smote</i>	Menyeimbangkan antara data <i>review</i> positif dan <i>review</i> negatif dengan <i>oversampling</i> .
<pre>word2count = {} counter = 0 vocab = list(word2id.keys()) for l in range(2): word2count[l] = defaultdict(lambda: 0) for i in range(X_oversampled.shape[0]): l = y_oversampled[i] for j in range(len(word2id)): word2count[l][vocab[j]] += X_oversampled[i][j]</pre>	Metode Hitung Banyak Kata	Menghitung banyak kata untuk setiap <i>review</i> positif dan negatif.
<pre>def laplace_smoothing(n_label_items, vocab, word_counts, word, text_label): a = word_counts[text_label][word] + 1 b = n_label_items[text_label] + len(vocab) return np.log10(a/b)</pre>	Fungsi <i>Laplace Smoothing</i>	Menghitung probabilitas nol dalam Naive Bayes.

<pre>def group_by_label(x, y, labels): data = {} for l in labels: data[l] = x[np.where(y == l)] return data</pre>	Fungsi Pengelompokkan Data	Kata dikelompokkan berdasarkan label.
<pre>def fit(x, y, labels): n_label_items = {} log_label_priors = {} n = len(x) grouped_data = group_by_label(x, y, labels) for l, data in grouped_data.items(): n_label_items[l] = len(data) log_label_priors[l] = np.log10(n_label_items[l] / n) return n_label_items, log_label_priors</pre>	Fungsi <i>Training</i>	Kata yang telah dihitung diterapkan fungsi <i>training</i> .
<pre>def predict(n_label_items, vocab, word_counts, log_label_priors, labels, x): result = [] for text in x: label_scores = {l: log_label_priors[l] for l in labels} words = text.split() for word in words: if word not in vocab: continue for l in labels: log_w_given_l = laplace_smoothing(n_label_items, vocab, word_counts, word, l) label_scores[l] += log_w_given_l result.append(max(label_scores, key=label_scores.get)) return result</pre>	Fungsi Prediksi	Setelah dilakukan <i>training</i> , kata dilakukan prediksi dengan kata baru.
<pre>labels = [0,1] n_label_items, log_label_priors = fit(X_oversampled,y_oversampled,labels) pred = predict(n_label_items, vocab, word2count, log_label_priors, labels, X_test) print(f'Accuracy of prediction on test set : {round(accuracy_score(y_test,pred), 3) * 100}%')</pre>	Fungsi Akurasi	Mengecek tingkat akurasi dari model Naive Bayes.

BAB 5

KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, maka terdapat beberapa kesimpulan:

1. Analisis sentimen pada review film IMDB dapat diterapkan algoritma Naive Bayes dengan dua buah kelas yaitu positif dan negatif.
2. Proses ekstraksi data pada review film IMDB menggunakan bahasa pemrograman Python dengan tools Google Colab.
3. Tingkat akurasi yang didapatkan adalah dengan 84,0%.

DAFTAR PUSTAKA

- [1] A. Dwi Herlambang and S. Hadi Wijoyo, “ALGORITMA NAÏVE BAYES UNTUK KLASIFIKASI SUMBER BELAJAR BERBASIS TEKS PADA MATA PELAJARAN PRODUKTIF DI SMK RUMPUN TEKNOLOGI INFORMASI DAN KOMUNIKASI,” vol. 6, no. 4, pp. 431–436, 2019, doi: 10.25126/jtiik.201961323.
- [2] D. Darwis, N. Siskawati, and Z. Abidin, “Penerapan Algoritma Naive Bayes untuk Analisis Sentimen Review Data Twitter BMKG Nasional,” vol. 15, no. 1.
- [3] A. Ashari Muin, “Metode Naive Bayes Untuk Prediksi Kelulusan (Studi Kasus: Data Mahasiswa Baru Perguruan Tinggi),” *Jurnal Ilmiah Ilmu Komputer*, vol. 2, no. 1, 2016, [Online]. Available: <http://ejournal.fikom-unasman.ac.id>