



Perbandingan *Gini Index* dan *Chi Square* pada Sentimen Analisis Ulasan Film menggunakan *Support Vector Machine Classifier*

Mahendra Dwifabri Purbolaksono^{#1}, Deninsyah Tiya Bella Pratama^{#2}, Fahmi Hamzah^{#3}

[#]Universitas Telkom

Jl. Telekomunikasi. 1, Kabupaten Bandung, Jawa Barat 40257

¹mahendradp@telkomuniversity.ac.id

²belladenin@students.telkomuniversity.ac.id

³ibuemar@students.telkomuniversity.ac.id

Abstrak— Pada era informasi ini semakin banyak penilaian, pendapat dan pandangan yang dapat ditemukan secara luas di dunia maya. Contohnya adalah ulasan film, di mana penonton berbagi pandangannya mengenai sebuah film. Ulasan film adalah platform di mana para penggemar film dapat mengungkapkan pendapat mereka, baik itu dalam bentuk komentar negatif atau pun positif. Sebagian besar *website* untuk ulasan film sudah memiliki rating atau bintang, namun rating tinggi tidak selalu diiringi oleh ulasan yang baik begitu pun sebaliknya. Untuk itu, dibutuhkan metode untuk menganalisis teks dengan tujuan mengklasifikasikan apakah ulasan film tersebut termasuk dalam kategori negatif ataupun positif. Teknik yang digunakan adalah analisis sentimen atau *opinion mining*. Analisis sentimen adalah bidang dalam *machine learning* yang bertujuan untuk mengambil informasi bersifat subjektif dari teks ulasan. Salah satu metode klasifikasi *machine learning* adalah *Support Vector Machine* (SVM). Namun semakin banyak data akan muncul beberapa masalah yaitu banyaknya kata atau fitur yang tidak relevan menyebabkan kinerja pengklasifikasian menurun. Fitur tidak relevan akan menyebabkan performansi yang rendah. Seleksi fitur *Gini Indeks* dan *Chi-Square* dibandingkan untuk mengatasi masalah kata yang tidak relevan. Pada penelitian ini, metode klasifikasi SVM dikombinasikan dengan metode seleksi fitur untuk meningkatkan performansi. Kombinasi SVM dan *Gini Index* menghasilkan performansi *F1-score* sebesar 85.8%. Sedangkan menggunakan SVM dan *Chi-Square* menghasilkan performansi *F1-score* tertinggi yaitu sebesar 89.2%.

Kata kunci— *Chi-Square*, *Feature Selection*, *Gini Index*, *Machine Learning*, *Sentiment Analysis*, *Support Vector Machine*

I. PENDAHULUAN

Industri perfilman mengalami pertumbuhan yang cepat dan menarik minat yang besar. Ketersediaan akses yang lebih mudah untuk menikmati film telah berkontribusi pada peningkatan jumlah penggemar film. Ulasan film berperan sebagai platform di mana para pencinta film dapat berbagi

pendapat atau evaluasi untuk konten tersebut. Pengguna yang menggunakan forum, blog, sosial media, atau pun *platform* lain di dunia maya memiliki kesempatan untuk mengekspresikan emosi, evaluasi, atau pendapat mengenai sebuah film [1]. Hal ini dapat menjadi panduan dan acuan bagi pengguna lain yang ingin menonton film tersebut. Ulasan film tersebut dapat berisikan suatu komentar positif atau negatif yang dapat memberikan informasi mengenai suatu film. Sebagian besar *website* untuk ulasan film sudah memiliki rating atau bintang, namun rating tinggi tidak selalu diiringi oleh ulasan yang baik begitu pun sebaliknya.

Metode yang diterapkan untuk mengklasifikasikan apakah ulasan film tersebut termasuk dalam kategori negatif atau positif adalah analisis sentimen atau penambangan opini. Pada saat ini, analisis sentimen sering dilakukan untuk memperoleh informasi serta polaritas mengenai suatu entitas atau objek dalam bentuk tekstual untuk mengetahui apakah kalimat tersebut cenderung positif atau negatif [2]. Sedangkan untuk pendekatan *machine learning*, merupakan pendekatan yang dalam penggunaannya memerlukan data manual dari *dataset* dan membuat *model* untuk melatih *classifier* yang dikenal dengan data *training* yang nantinya akan diujikan dengan data *testing* pada proses klasifikasi sentimen.

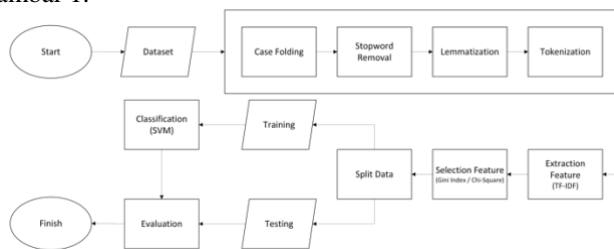
Salah satu penelitian sebelumnya menunjukkan bahwa *Support Vector Machine* sebagai algoritma klasifikasi mempunyai nilai akurasi yang lebih tinggi yaitu 89% daripada pada *Naive Bayes*, *Random Forest*, dan *Decision Tree* [3][4]. Banyaknya fitur yang terkandung dalam dataset serta terdapat fitur yang kurang relevan terhadap kelas keluaran tertentu akan mengakibatkan menurunnya performansi dalam proses klasifikasi yang dibangun. Untuk mengatasi hal tersebut, perlu adanya penggunaan seleksi fitur [5]. Seleksi fitur merupakan salah satu teknik yang digunakan untuk mengurangi dimensi fitur yang tinggi dalam data yang ada dengan menghapus fitur-fitur yang kurang relevan. Pada penelitian [6] yang membandingkan penggunaan metode seleksi fitur, didapatkan bahwa

metode *Gini Index* dapat menghasilkan performansi yang lebih baik dibandingkan dengan metode seleksi fitur lainnya. Lalu, sebuah studi mengenai pemanfaatan seleksi fitur dengan menggunakan metode Chi-Square pada analisis sentimen dokumen [7] mengambil kesimpulan bahwa penggunaan Chi-Square dalam seleksi fitur menghasilkan peningkatan akurasi sebesar 13,33%. Penggunaan metode seleksi fitur diharapkan mampu untuk meningkatkan kinerja dari algoritma klasifikasi *Support Vector Machine* yang bekerja pada dimensi fitur yang tinggi. Pada penelitian [8] telah melakukan pengujian untuk mengetahui performansi penggunaan seleksi fitur *Gini Index* pada klasifikasi *Multinomial Naïve Bayes*. Dalam penelitian tersebut didapatkan bahwa pemilihan *Gini Index* sebagai seleksi fitur dapat meningkatkan akurasi sebesar 3,5% dengan hasil 59,54%. Serta pada penelitian [9] membandingkan penggunaan TF-IDF dan *Word Embedding* sebagai metode ekstraksi fitur dengan klasifikasi *Artificial Neural Network*. TF-IDF merupakan metode pembobotan yang berdasarkan pada bobot kata. Didapatkan bahwa TF-IDF merupakan metode *feature extraction* terbaik dibandingkan dengan *Word Embedding* dengan hasil akurasi 80,55%. Sehingga didapatkan kesimpulan bahwa pemilihan metode *feature extraction* juga dapat meningkatkan performansi dari algoritma klasifikasi.

Berdasarkan kondisi tersebut, penelitian ini menggunakan pendekatan machine learning yang berfokus pada algoritma *Support Vector Machine* dengan Perbandingan *Gini Index* dan *Chi-Square* sebagai seleksi fitur dalam melakukan analisis sentimen terhadap ulasan film. Adapun penelitian ini dilakukan untuk dapat mengelompokkan suatu pendapat atau opini terhadap suatu film, dengan melihat kecenderungan opini tersebut apakah tergolong suatu komentar positif atau negatif. Serta dilakukan pengujian pada penelitian ini yang berdasarkan pada skenario pengujian untuk mengetahui performansi dari sistem yang dibangun dengan menggunakan evaluasi *F1-Score*, di mana semakin tinggi hasil yang diperoleh maka semakin baik sistem yang dibangun.

II. METODOLOGI PENELITIAN

Pada bagian metodologi penelitian, akan dipaparkan mengenai struktur sistem yang dibangun. Langkah-langkah yang diterapkan dalam analisis sentimen pada ulasan film adalah ekstraksi fitur menggunakan TF-IDF, seleksi fitur dengan *Chi-square* dan *Gini Index*, serta penerapan pada model klasifikasi menggunakan algoritma *Support Vector Machine (SVM)*. Skema sistem yang dirancang ditunjukkan dalam diagram alur seperti yang terlihat pada Gambar 1.



Gambar. 1 Gambaran umum perancangan sistem

A. Dataset

Dataset yang digunakan dalam penelitian ini adalah data ulasan film dalam berbahasa Inggris yang didapat dari *Internet Movie Database (IMDb)* dengan menggunakan format file .csv. *Dataset* ulasan film merupakan kumpulan opini dari penikmat film mengenai film yang ada pada *Internet Movie Database (IMDb)*. *Dataset* yang digunakan dalam penelitian ini, berjumlah 5.000 data ulasan yang terdiri dari dokumen dengan sentimen negatif dan positif. Tabel 1 berikut merupakan contoh data yang digunakan dalam penelitian ini.

TABEL I
DATASET REVIEW MOVIE IMDB

Review	Sentimen
<i>I had the terrible misfortune of having to view this "b-movie" in it's entirety. ... This has got to be the worst b-movie of all time, it shouldn't even be called a b-movie, more like an f-movie! ...</i>	0 (Negative)
<i>An amazing film indeed. If you have 2.5 hours to spare, it's definitely worth watching. ... The music composed by A.R. Rahman might take some time to resonate with you, but after listening to it a few times, you'll find yourself genuinely enjoying it.</i>	1 (Positive)

B. Preprocessing

Tahapan ini merupakan proses pembuangan kata-kata yang tidak relevan yang ada pada dataset. Tujuan dari tahap preprocessing adalah untuk mengurangi jumlah fitur yang tidak penting yang dapat menurunkan efektivitas serta efisiensi pada proses klasifikasi. Hasil yang didapat dari tahap preprocessing yaitu berupa pemotongan kata tiap kalimatnya yang akan dilanjutkan ke dalam tahap pembobotan kata dengan TF-IDF. Berikut merupakan tahapan yang terdapat dalam preprocessing.

1) *Case Folding*: *Case folding* merupakan teknik membersihkan data dengan menghilangkan huruf besar, tanda baca, alamat URL atau situs. Sehingga menyisakan kalimat dalam dataset menjadi huruf kecil untuk semua teks seperti di tabel 2.

TABEL II
CONTOH CASE FOLDING

Input	Output
<i>I had the terrible misfortune of having to view this "b-movie" in it's entirety. ... This has got to be the worst b-movie of all time, it shouldn't even be called a b-movie, more like an f-movie!</i>	<i>i had the terrible misfortune of having to view this bmovie in its entirety this has got to be the worst bmovie of all time it shouldnt even be called a bmovie more like an fmovie</i>

2) *Stopword Removal*: merupakan langkah untuk mengeliminasi kata-kata yang memiliki dampak minim, seperti kata ganti orang atau konjungsi. Proses *filtering* dapat digambarkan seperti di tabel 3.

TABEL III
CONTOH STOPWORD REMOVAL

Input	Output
<i>i had the terrible misfortune of having to view this bmovie in its entirety this has got to be the worst bmovie of all time it shouldnt even be called a bmovie more like an fmovie</i>	<i>terrible misfortune view bmovie entirety got worst bmovie time not even called bmovie like fmovie</i>

3) *Lemmatization*: merupakan proses mengembalikan kata menjadi kata dasar yang sesuai dengan kamus dan tetap memiliki makna yang sama. Setelah dilakukan *preprocessing* tahap *filtering* atau *stopword removal*, maka dataset ulasan film akan dilakukan proses *preprocessing* tahap *lemmatization*. Tabel 4 merupakan gambaran terhadap tahapan *lemmatization*.

TABEL IV
CONTOH LEMMATIZATION

Input	Output
terrible misfortune view bmovie entirety got worst bmovie time not even called bmovie like fmovie	terrible misfortune view bmovie entirety get bad bmovie time not even call bmovie like fmovie

4) *Tokenization*: *Tokenization* atau tokenisasi merupakan proses dimana kalimat dalam *dataset* akan dirubah menjadi potongan per kata seperti pada tabel 5.

TABEL V
CONTOH TOKENIZATION

Input	Output
terrible misfortune view bmovie entirety got worst bmovie time not even called bmovie like fmovie	terrible misfortune view bmovie entirety get bad bmovie time not even call bmovie like fmovie

C. Ekstraksi Fitur

Pada tahapan ini akan dilakukan ekstraksi fitur pada dataset setelah dilakukannya tahap *preprocessing*. *Feature extraction* merupakan metode yang digunakan untuk menghasilkan fitur baru yang memiliki dimensi lebih sedikit dari data *real* nya [10]. Hal tersebut dilakukan untuk meningkatkan performansi dari metode machine learning yang digunakan. Salah satu metode *feature extraction* yang sering digunakan adalah TF-IDF [9].

Term Frequency (TF) adalah kemunculan suatu fitur pada dokumen yang bersangkutan. Sedangkan *Inverse Document Frequency* (IDF) adalah perhitungan mengenai suatu fitur yang didistribusikan secara luas pada dokumen. *Term Frequency and Inverse Document Frequency* (TF-IDF) sendiri merupakan kombinasi antara TF dan IDF yang digunakan untuk menganalisa suatu fitur dengan sekumpulan dokumen [9]. TF-IDF sering digunakan untuk *feature extraction* dalam klasifikasi suatu teks karena dapat meningkatkan performansi dari metode algoritma klasifikasi [9][11]. Berikut rumus untuk mendapatkan nilai TF-IDF.

$$tf_{t,d} = f_{t,d} \tag{1}$$

Persamaan (1) merupakan rumus TF yang menyatakan jumlah kemunculan suatu fitur yang terdapat dalam dokumen, dan berikut adalah persamaan mengenai rumus IDF[11].

$$idf_t = 1 + \log \frac{N}{df_t} \tag{2}$$

Persamaan (2) rumus menghitung IDF, dengan N merupakan jumlah dokumen dan df adalah jumlah dokumen yang mengandung suatu fitur t. Dari kedua persamaan tersebut akan digunakan sebagai rumus TF-IDF

pada persamaan (3) untuk menghitung bobot nilai dari fitur yang terdapat dalam dokumen yang dinyatakan dengan w.

$$w = tf_{t,d} \times idf_t \tag{3}$$

D. Seleksi Fitur menggunakan Gini Index

Pada tahapan ini akan dipilih fitur dengan nilai terbaik dan nantinya akan dipakai dalam proses klasifikasi. Menggunakan metode *Gini Index* sebagai *feature selection* dalam memilih fitur, tujuan dari hal tersebut agar mempercepat proses klasifikasi dan meningkatkan performansi dengan memilih fitur yang berpengaruh. Dalam tahapan ini menggunakan input dari hasil pembobotan kata untuk tiap fiturnya yang nantinya akan dihitung nilai *Gini Index* sesuai kemunculan fitur tiap kelasnya. Kemudian output dari seleksi fitur ini adalah urutan fitur dari yang relevan ke yang irrelevan dalam bentuk *vector*.

Gini Index merupakan salah satu metode seleksi fitur yang mengukur tingkat korelasi suatu fitur dengan kelasnya. Pada umumnya, *Gini Index* sering digunakan untuk proses pemilihan fitur pada *decision tree* [6]. Tujuan penggunaan *Gini Index* sebagai seleksi fitur adalah untuk mengurangi dimensi yang tinggi pada suatu dataset dan membantu menyeleksi fitur dengan menggunakan fitur atau parameter yang tingkat relevannya tinggi [12], [13]. Rumus *Gini Index* dapat didefinisikan sebagai berikut [13]: misalkan S merupakan satu set dari s data. Di mana s data tersebut mempunyai sejumlah k kelas yang berbeda ($C_{i=1, \dots, k}$). Berdasarkan perbedaan kelas tersebut, kita dapat membagi S ke dalam sejumlah k *subset* ($S_{i=1, \dots, k}$). Misalkan S_i merupakan himpunan dari dataset yang termasuk ke dalam kelas C_i . Maka nilai *Gini Index* dari himpunan S_i tersebut adalah hasil dari persamaan (4) berikut.

$$GINI(S) = 1 - \sum_{i=1}^k P_i^2 \tag{4}$$

Dimana P_i merupakan probabilitas dari setiap sampel kelas yang ada yaitu C_i , perhitungan *Gini Index* ini dimaksudkan untuk menghitung ketidaksesuaian fitur yang akan digunakan untuk klasifikasi. Apabila semua sampel dalam himpunan mempunyai distribusi kelas yang setara, maka $GINI(S)$ maksimum akan menghasilkan nilai gini minimum. Hasil perhitungan gini index yang mempunyai nilai yang rendah maka semakin baik nilai suatu fiturnya [13].

Adapun proses yang dilakukan dalam melakukan *Gini Index* adalah dengan menghitung *Gini Index* untuk setiap fitur yang terkandung dengan menggunakan persamaan (4). Kemudian melakukan pengurutan secara *descending* terhadap hasil perhitungan yang didapat untuk setiap fiturnya.

E. Seleksi Fitur menggunakan Chi-Square

Ini adalah langkah di mana fitur-fitur dipilih untuk mengurangi kata-kata yang dianggap memiliki dampak yang lebih rendah dalam analisis sentimen [7]. Dalam konteks penyeleksian fitur kali ini, penerapan Chi-square bertujuan untuk mengukur tingkat ketergantungan antara kata tertentu dalam sebuah dokumen dengan kelas negatif atau positif. Langkah awal yang dijalankan adalah

menetapkan hipotesis nol (H0) serta hipotesis alternatif (H1).

- Hipotesis 0 (*null hypothesis*) = kata tidak tergantung dengan label kelas.
- Hipotesis 1 (*alternate hypothesis*) = kata tergantung dengan label kelas.

Rumus *Chi-square* (X^2) pada persamaan (5) berikut:

$$X^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{5}$$

Nilai *observed* (o) mencakup jumlah kemunculan suatu fitur dalam setiap kelas, sementara nilai *expected* (E) mencakup nilai yang berasal dari total term yang muncul di seluruh kelas (m), dikalikan dengan nilai total dalam setiap kelas (n), kemudian dibagi oleh seluruh data (D). Berikut rumus dalam menemukan nilai E pada persamaan (6) berikut:

$$E_i = \frac{m \cdot n}{D} \tag{6}$$

Langkah berikutnya yaitu mencari nilai probabilitas (p) fitur yang tidak tergantung kepada suatu kelas dengan melihat *Chi-square table*.

Pada tahap ini, diperlukan nilai *degree of freedom* (DoF) dengan mengurangi satu dari jumlah baris. Sebagai ilustrasi, jika terdapat nilai *Chi-square* sebesar 2,706 pada baris dengan DoF = 1 dan memiliki nilai p = 0,1. Untuk seleksi fitur, diperlukan nilai alpha (α) atau nilai signifikansi untuk melakukan pengujian *hypothesis*. Jika:

- jika nilai $p \leq \alpha$, maka istilah tersebut menolak Hipotesis 0, atau menunjukkan adanya ketergantungan,
- jika nilai $p > \alpha$, maka istilah tersebut menerima Hipotesis 0, atau menunjukkan ketidakbergantungan.

Dalam penelitian ini, untuk *terms* yang memiliki nilai p lebih dari nilai alpha (α) atau nilai signifikansi yang ditentukan, fitur-fitur tersebut akan dihilangkan.

$$\alpha = 1 - \text{Selang Kepercayaan} \tag{6}$$

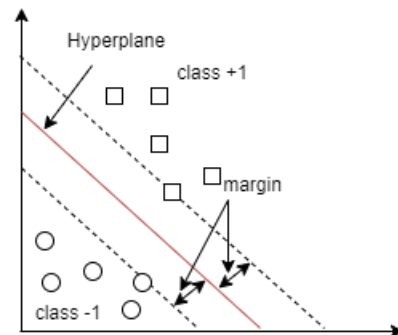
Di mana alpha (α) melambangkan level signifikansi, dan interval kepercayaan mencerminkan level keyakinan terhadap H0, yang berarti menganggap bahwa semua fitur adalah independen [14]. Tabel 6 berikut adalah indeks yang akan digunakan dalam implementasi penelitian ini.

TABEL VI
INDEX CHI-SQUARE YANG DIGUNAKAN DALAM IMPLEMENTASI

dof	1	2	3	4	5	6
0.975	0.000982	0.051	0.216	0.484	0.831	1.237
0.95	0.00393	0.103	0.352	0.711	1.145	1.635
0.9	0.0158	0.211	0.584	1.064	1.61	2.204
0.5	0.455	1.386	2.366	3.357	4.351	5.348
0.2	1.642	3.219	4.642	5.989	7.289	8.558
0.1	2.706	4.605	6.251	7.779	9.236	10.645
0.05	3.841	5.991	7.815	9.488	11.07	12.592
0.025	5.024	7.378	9.348	11.143	12.833	14.449
0.02	5.412	7.824	9.837	11.668	13.388	15.033

F. Support Vector Machine (SVM)

Proses klasifikasi dalam penelitian ini menggunakan algoritma *Support Vector Machine* (SVM). SVM adalah algoritma *supervised learning* yang dapat digunakan dalam klasifikasi atau pun regresi. Namun, sering digunakan dalam penyelesaian kasus klasifikasi [3]. SVM memiliki kemampuan untuk bekerja dengan data yang besar serta dapat mencapai hasil yang tinggi dalam klasifikasi suatu teks [5]. Tujuan dalam penggunaan klasifikasi SVM adalah mencari *hyperplane* terbaik. Terdapat ilustrasi dalam pencarian *hyperplane* terbaik, sebagai berikut:



Gambar. 2 Ilustrasi pencarian *hyperplane*

Pada Gambar 2, menunjukkan gambaran dalam pencarian *hyperplane* terbaik dalam *support vector machine*. *Hyperplane* mempunyai fungsi untuk memisahkan dua buah kelas. Terdapat *pattern* yang menggambarkan anggota antara dua buah kelas yaitu -1 dan +1. Sedangkan margin merupakan jarak terdekat antara *hyperplane* dan *pattern* untuk setiap kelas dan diilustrasikan dengan garis yang terputus-putus. *Hyperplane* optimal dapat diidentifikasi dengan mengidentifikasi pola yang memiliki jarak terdekat dengan *hyperplane* dari kedua kelas. Jarak minimal antara *hyperplane* dan pola tersebut dikenal sebagai vektor pendukung (*support vector*) [15].

Dalam ilustrasi diatas *hyperplane* terbaik dapat direpresentasikan dengan garis solid berwarna merah yang terletak diantara kedua kelas yaitu -1 dan +1. Data yang ada dinotasikan sebagai $\vec{x}_1 \in \mathcal{R}^d$, \vec{w} normal bidang dan b merupakan posisi bidang *relative*. Sedangkan untuk kedua kelas, dinotasikan dengan $y_i \in \{-1, +1\}$ untuk semua i

dengan jumlah sebanyak n . Untuk kelas -1 dan $+1$ dapat diasumsikan bahwa kelas dipisahkan oleh *hyperplane* berdimensi d , dengan persamaan (7) sebagai berikut [15], [16].

$$\vec{w} \cdot \vec{x}_1 + b = 0 \tag{7}$$

Pattern \vec{x}_1 yang didalamnya mengandung kelas $+1$, didapatkan pada persamaan (8) sebagai berikut:

$$\vec{w} \cdot \vec{x}_1 + b \geq +1 \tag{8}$$

Untuk *pattern* \vec{x}_1 yang didalamnya mengandung kelas -1 , didapatkan pada persamaan (9) sebagai berikut:

$$\vec{w} \cdot \vec{x}_1 + b \leq -1 \tag{9}$$

Sedangkan untuk menemukan margin dengan nilai terbesar nilai jarak dapat dimaksimalkan antara *hyperplane* dan titik terdekatnya, yaitu pada persamaan (10) berikut:

$$\frac{1}{\|\vec{w}\|} \tag{10}$$

G. Evaluasi

Setelah semua proses telah dilakukan, maka proses terakhir adalah melakukan evaluasi terhadap metode yang digunakan. Digunakan suatu confusion matrix untuk menghitung performansi dari suatu metode klasifikasi *Support Vector Machine* pada ulasan film. *Confusion matrix* mempunyai 4 istilah dalam penggunaannya [16], yaitu:

TABEL VII
CONFUSION MATRIX

Prediksi	Kelas yang sebenarnya	
	positif	negatif
Prediksi positif	TP	FP
Prediksi negatif	FN	TN

- *True Positive* (TP): jumlah ulasan film prediksi positif dengan klasifikasi positif.
- *True Negative* (TN): jumlah ulasan film prediksi negatif dengan klasifikasi negatif.
- *False Positive* (FP): jumlah ulasan film prediksi negatif dengan klasifikasi positif.
- *False Negative* (FN): jumlah ulasan film prediksi positif dengan klasifikasi negatif.

Confusion matrix tersebut dapat digunakan untuk mengetahui performansi dari metode klasifikasi seperti *Accuracy*, *Precision*, *Recall*, *F1-Score*. Berikut adalah rumus untuk menghitung performansi:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{11}$$

$$Precision = \frac{TP}{TP+FP} \tag{12}$$

$$Recall = \frac{TP}{TP+FN} \tag{13}$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{14}$$

III. HASIL DAN PEMBAHASAN

Penelitian ini telah menyiapkan *dataset* sebesar 5000 ulasan film akan dilakukan proses sampai dengan evaluasi. Hasil evaluasi dinyatakan dalam *Accuracy* pada persamaan (11), serta *F1-Score* pada persamaan (14). *F1-Score* merupakan rata-rata harmonik dari nilai *Precision* pada persamaan (12) dan *Recall* pada persamaan (13). *F1-Score* sendiri memberikan gambaran seberapa presisi dan seberapa baik model dalam memberikan label untuk klasifikasi. Oleh karena itu, *F1-Score* digunakan sebagai metode evaluasi dari sistem yang telah dibangun, di mana sistem yang dibangun semakin baik, maka nilai *F1-Score* yang didapatkan semakin tinggi.

A. Hasil Pengujian dan Analisis Skenario 1

Pengujian pada skenario pertama melakukan tahap *preprocessing* tanpa *lemmatization* dan dengan *lemmatization*. Hasil dari kedua pengujian tersebut digunakan untuk mengetahui seberapa besar pengaruh *lemmatization* pada sistem yang dibangun. Berikut adalah hasil perbandingan pengujian mengenai *preprocessing* dengan menggunakan dan tanpa menggunakan *lemmatization* terhadap proses klasifikasi.

TABEL VIII
HASIL PENGUJIAN PREPROCESSING

Uji <i>Pre-processing</i>	<i>Accuracy</i>	<i>Recall</i>	<i>Precision</i>	<i>F1-Score</i>
Tanpa <i>Lemmatization</i>	84,8%	86,87%	85,1%	85,9%
Dengan <i>Lemmatization</i>	85,2%	86,25%	86,07%	86,1%

Tabel 8 menampilkan pengaruh pengujian *preprocessing* tanpa *lemmatization* dan dengan *lemmatization*. Hasil evaluasi menggunakan perhitungan *F1-Score*, dimana semakin tinggi nilai *F1-Score* yang didapatkan, menyatakan bahwa sistem yang dibangun semakin baik. Hasil pengujian dengan nilai *F1-Score* yang terbaik akan digunakan untuk proses seleksi fitur klasifikasi.

Pada tabel 8 dapat diketahui bahwa perbandingan penggunaan *preprocessing* tanpa menggunakan *lemmatization* mendapatkan nilai *F1-Score* sebesar 85,9% dengan fitur yang dihasilkan sebanyak 36.222 fitur. Sedangkan, dengan menggunakan *lemmatization* mempunyai nilai akurasi sebesar 86,1% dengan fitur yang dihasilkan sebanyak 30.242. Berdasarkan hasil tersebut, didapatkan bahwa penggunaan *preprocessing* menggunakan *lemmatization* lebih unggul dibandingkan tanpa menggunakan *lemmatization*. Hal ini disebabkan karena pada tahapan *lemmatization* dilakukannya perubahan bentuk untuk setiap kata menjadi kata dasar. Hal tersebut akan membuat dimensi menjadi lebih kecil dan kata yang memiliki arti sama menjadi satu kata saja [17]. Sebagai contoh kata atau term “*worst*” dan “*bad*”, dalam proses *lemmatization* kata “*worst*” akan diubah kedalam

bentuk dasar “*bad*” sehingga memiliki makna yang sama dengan kata aslinya.

B. Hasil Pengujian dan Analisis Skenario 2

Pada skenario kedua dalam penelitian ini, perbandingan hasil kinerja dari proses klasifikasi dievaluasi dengan mengubah nilai signifikansi dalam Chi-square. Dalam proses ini digunakan kernel linear.

Tabel 9 menunjukkan jumlah fitur yang diambil berdasarkan nilai signifikansi yang diaplikasikan. Penggunaan nilai signifikansi hingga 0,15 dipilih karena mencapai tingkat kepercayaan maksimal bahwa semua istilah independen minimal adalah 0,85.

TABEL IX
PENGAMBILAN FITUR PADA SETIAP NILAI SIGNIFIKASI CHI-SQUARE

Fitur	Signifikansi			Tanpa Seleksi Fitur
	0,05	0,10	0,15	
Dihapus	27.946	27.265	26.607	0
Diambil	989	1.670	2.328	28.935

Menurut tabel 9, semakin besar tingkat signifikansi yang diterapkan, semakin besar juga fitur yang diambil. Begitu juga dengan tingkat signifikansi semakin kecil, maka semakin kecil fitur yang diambil. Tentu saja, penggunaan fitur-fitur ini berpengaruh pada hasil kinerja. Pada tabel 10 yang memperlihatkan performansi untuk setiap nilai signifikansi yang diujikan.

TABEL X
HASIL PENGUJIAN CHI-SQUARE

Evaluasi	Signifikansi			Tanpa Seleksi Fitur
	0,05	0,10	0,15	
Akurasi	83,6%	84,0%	83,9%	84,6%
Persisi	86,4%	86,8%	87,0%	84,8%
Recall	81,81%	82,19%	82,53%	84,46%
F1-score	84,04%	84,43%	84,23%	84,63%

Dalam Tabel 10 di atas, model tanpa seleksi fitur menunjukkan hasil *F1-score* lebih tinggi dibandingkan model yang menggunakan seleksi fitur Chi-square. Hal tersebut disebabkan karena model yang menerapkan seleksi fitur memiliki jumlah fitur yang lebih sedikit, mengakibatkan fitur-fitur yang dapat berpengaruh pada performansi terbuang. Dari Tabel 10, dapat dilihat bahwa semakin banyak fitur, rasio nilai *True Positive* terhadap total nilai positif yang diprediksi semakin besar. Meskipun demikian, pada model tanpa seleksi fitur, nilai presisi menurun dari model lainnya. Penurunan tersebut disebabkan keberadaan *noise* pada kelas positif yang memengaruhi nilai presisi dalam model yang tidak melibatkan seleksi fitur [5].

Fitur yang dipilih semakin banyak akan meningkatkan kemampuan SVM dalam memprediksi. Melihat Tabel 10, nilai *recall* meningkat dengan meningkatnya jumlah fitur yang digunakan. Jika keberimbangan antara *precision* dan *recall* diinginkan, model tanpa seleksi fitur dapat menjadi pilihan karena memiliki nilai *F1-score* tertinggi di antara model-model lainnya.

C. Hasil Pengujian dan Analisis Skenario 3

Hasil terbaik pada pengujian skenario pertama akan digunakan dalam pengujian skenario kedua. Dimana *lemmatization* digunakan pada tahap preprocessing, pada saat melakukan *Gini Index* fitur akan diurutkan berdasarkan nilai *Gini Index* secara *descending*. Selanjutnya, melakukan pengujian mengenai jumlah fitur yang digunakan. Fitur awal sebelum dilakukan proses seleksi fitur adalah sebesar 28935. Selanjutnya, membagi data latih dan data uji sebesar 70:30 sebelum dilakukannya tahap klasifikasi menggunakan *Support Vector Machine* dengan *kernel linear*. Berikut adalah perbandingan hasil pengujian penggunaan fitur dan pengujian terhadap pengaruh penggunaan seleksi fitur *Gini Index*.

TABEL XI
HASIL PERFORMANSI NILAI SIGNIFIKANSI

Jumlah Fitur	Accuracy	Recall	Precision	F1-Score
50	80%	90%	76%	82,7%
100	82,5%	90,83%	79,17%	84,74%
1.000	87,4%	90%	86%	88%
5.000	88,33%	90,41%	88,03%	89,21%
10.000	87,33%	88,12%	88,12%	88,12%
15.000	86,77%	88,54%	86,91%	87,71%
28.935 (Fitur Asli)	84,6%	84,46%	84,8%	84,63%

Pada tabel 11, pengujian didapatkan bahwa saat melakukan perbandingan pengujian mengenai pengaruh *Gini Index* sebagai fitur seleksi terhadap metode klasifikasi SVM. Performansi SVM tanpa menggunakan *Gini Index* sebagai fitur seleksi menghasilkan nilai *F1-Score* sebesar 86,1% dengan menggunakan fitur sebanyak 28.935, sedangkan untuk performansi SVM setelah digunakannya *Gini Index* dengan menggunakan jumlah fitur 5.000 dimana nilai *Gini Index* terbaik diperoleh sebesar 0,499649 dan mampu menghasilkan *F1-Score* sebesar 89,21% serta mengurangi fitur sebanyak 23.935 dari total fitur 28.935. Pada saat pengujian menggunakan fitur 50 dan 100 menghasilkan nilai *F1-Score* di bawah hasil *F1-Score* tanpa *Gini Index* sebagai seleksi fitur. Hal ini dikarenakan fitur-fitur yang berada dalam rentang tersebut masih belum cukup untuk mewakili keseluruhan fitur. Sehingga hal tersebut menurunkan perolehan nilai *F1-Score* artinya diperlukan lagi untuk lebih banyak fitur untuk memberikan hasil performansi yang lebih baik. Pada saat pengujian menggunakan fitur 10.000 dan 15.000 terjadi penurunan perolehan *F1-score* yang dihasilkan. Hal ini menunjukkan bahwa terlalu banyak fitur yang digunakan, maka akan terjadi penurunan *F1-Score*. Begitu juga, jika terlalu sedikit fitur yang digunakan maka perolehan *F1-Score* yang didapatkan menurun. Sedangkan saat menggunakan fitur 28.935 memperoleh nilai *F1-Score* sebesar 86,1% yang

merupakan pengujian tanpa seleksi fitur. Sehingga penggunaan seleksi fitur *Gini Index* dapat menaikkan performansi dari suatu sistem yang dibangun. Hal ini disebabkan karena *Gini Index* melakukan seleksi fitur dengan menggunakan fitur yang mempunyai tingkat relevan yang tinggi untuk dapat dilakukan dalam proses klasifikasi [6]. Selain itu penggunaan seleksi fitur ini bertujuan untuk mengurangi dimensi data yang ada sehingga membuat performansi klasifikasi dapat meningkat.

D. Hasil Pengujian dan Analisis Skenario 4

Tahap pengujian 4 melibatkan penggunaan model dengan metode SVM untuk menemukan parameter teroptimal yang diimplementasikan pada setiap kernel *linear*, *polynomial*, dan RBF dalam SVM. Pengujian ini dilakukan melalui metode *GridSearch* dengan tiga (3) kali validasi silang. Tujuan dari skenario ini adalah membandingkan kinerja masing-masing *kernel*. Pengujian dilakukan menggunakan seleksi fitur *Gini Index*.

TABEL XII
HASIL PERBANDINGAN PENGGUNAAN KERNEL

Evaluasi	Kernel		
	<i>Polynomial</i>	<i>Linear</i>	<i>RBF</i>
Akurasi	83,80%	84,60%	88,30%
Presisi	83,60%	84,46%	90,41%
Recall	83,93%	84,80%	88,03%
F1-Score	83,76%	84,63%	89,21%

Pada tabel 12 dapat dilihat bahwa kernel *Radial Basis Function* (RBF) unggul pada semua nilai evaluasi. SVM dengan RBF terbukti paling optimal untuk dataset atau kasus yang diambil pada penelitian ini. RBF merupakan kernel nonlinear dari SVM. Pada kasus ini linear lebih rendah dari pada nonlinear. Hal tersebut dapat disebabkan kompleksitas masalah kernel yang berbeda-beda dan tergantung dari fitur-fitur yang digunakan [15].

IV. KESIMPULAN

Dalam penelitian ini digunakan 5.000 data ulasan film yang terdiri dari 2.500 data negatif dan 2.500 positif. Berdasarkan hasil skenario, data telah melewati proses pra-proses, kemudian diberi bobot lalu diseleksi menggunakan metode Chi-Square dan Gini Index. Selanjutnya mencari kernel SVM terbaik antara *liner*, *polynomial* dan RBF. Dapat diambil kesimpulan bahwa, klasifikasi sentimen terhadap ulasan film menggunakan *Support Vector Machine* dengan seleksi fitur *Gini Index* dan kernel RBF terbukti dapat memberikan nilai akurasi tertinggi dengan memperoleh nilai *F1-Score* sebesar 89,2% dengan pengurangan fitur menjadi 5000 serta mampu mengurangi fitur sebanyak 23.935 dari total fitur 28.935, dibandingkan tanpa menggunakan seleksi fitur yang hanya memperoleh hasil *F1-Score* sebesar 84,63%.

Beberapa saran yang dapat dilakukan untuk penelitian selanjutnya adalah menggunakan berbagai metode ekstraksi fitur selain TF-IDF, dikarenakan penggunaan ekstraksi fitur dapat memberikan pengaruh dalam peningkatan akurasi dari metode klasifikasi yang dipakai. Serta menggunakan metode klasifikasi lainnya selain SVM.

REFERENSI

- [1] D. D. Tran, T. T. S. Nguyen, and T. H. C. Dao, "Sentiment Analysis of Movie Reviews Using Machine Learning Techniques," 2022, pp. 361–369. doi: 10.1007/978-981-16-2377-6_34.
- [2] F. Hemmatian, and M. K. Sohrabi. "A survey on classification techniques for opinion mining and sentiment analysis." *Artificial intelligence review* vol 52, pp. 1495-1545, 2019, doi: 10.1007/s10462-017-9599-6.
- [3] Y. Al Amrani, M. Lazaar, and K. E. El Kadiri, "Random Forest and Support Vector Machine based Hybrid Approach to Sentiment Analysis," *Procedia Comput Sci*, vol. 127, pp. 511–520, 2018, doi: 10.1016/j.procs.2018.01.150.
- [4] M. Guia, R. Silva, and J. Bernardino, "Comparison of Naïve Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis," in *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, SCITEPRESS - Science and Technology Publications, pp. 525–531, 2019, doi: 10.5220/0008364105250531.
- [5] Kou, G., Yang, P., Peng, Y., Xiao, F., Chen, Y., & Alsaadi, F. E, "Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods," in *Applied Soft Computing*, vol. 86, pp 105836, 2020, doi: 10.1016/j.asoc.2019.105836.
- [6] Deng, X., Li, Y., Weng, J., and Zhang, J, "Feature selection for text classification: A review," *Multimedia Tools and Applications*, vol 78, pp 3797-3816, 2019, doi: 10.1007/s11042-018-6083-5.
- [7] Nurhayati, A. E. Putra, L. K. Wardhani, and Busman, "Chi-Square Feature Selection Effect on Naive Bayes Classifier Algorithm Performance For Sentiment Analysis Document," in *2019 7th International Conference on Cyber and IT Service Management (CITSM)*, IEEE, Nov. 2019, pp. 1–7. doi: 10.1109/CITSM47753.2019.8965332.
- [8] Riko Bintang Purnomoputra, Adiwijaya Adiwijaya, and Untari Novia Wisesty, "Sentiment Analysis of Movie Review Using Naïve Bayes Method with Gini Index Feature Selection," *Journal of Data Science and Its Applications*, vol. 2, no. 2, pp. 85–94, Jul. 2019.
- [9] A. W. Romadon, K. M. Lhaksana, I. Kurniawan, and D. Richasdy, "Analyzing TF-IDF and Word Embedding for Implementing Automation in Job Interview Grading," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*, IEEE, Jun. 2020, pp. 1–4. doi: 10.1109/ICoICT49345.2020.9166364.
- [10] Riski Anisa, "Pendekatan Metode Feature Extraction Dengan Algoritma Naïve Bayes," in *Konferensi Nasional Ilmu Sosial dan Teknologi*, 2017, pp. 19–24.
- [11] M. Das, and P. J. A. Alphonse. "A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset." *arXiv preprint arXiv:2308.04037*, 2023 doi: 10.48550/arXiv.2308.04037.
- [12] Tyas Setiyorini and Rizky Tri Asmono, "Penerapan Metode K-Nearest Neighbor Dan Gini Index Pada Klasifikasi Kinerja Siswa," *Jurnal TECHNO Nusa Mandiri*, vol. 16, no. 2, pp. 121–126, 2019.
- [13] A. S. Manek, P. D. Shenoy, M. C. Mohan, and V. K. R, "Aspect term extraction for sentiment analysis in large movie reviews using Gini Index feature selection method and SVM classifier," *World Wide Web*, vol. 20, no. 2, pp. 135–154, Mar. 2017, doi: 10.1007/s11280-015-0381-x.
- [14] M. Bendtsen, "A Gentle Introduction to the Comparison Between Null Hypothesis Testing and Bayesian Analysis: Reanalysis of Two Randomized Controlled Trials," *J Med Internet Res*, vol. 20, no. 10, p. e10873, Oct. 2018, doi: 10.2196/10873.
- [15] D. A. Pisner, and D. M. Schnyer. "Support vector machine," *Machine learning*, Academic Press, pp 101-121, 2020 doi: 10.1016/B978-0-12-815739-8.00006-7
- [16] M. A. Chandra, and S. S. Bedi. "Survey on SVM and their application in image classification." *International Journal of Information Technology*, vol. 13, pp 1-11, 2021, doi: 10.1007/s41870-017-0080-1.
- [17] M. D. Purbolaksono, F. D. Reskyadita, Adiwijaya, A. A. Suryani, and A. F. Huda, "Indonesian text classification using back propagation and sastrawi stemming analysis with information gain for selection feature," *International Journal on Advanced Science, Engineering and Information Technology*, Issue 1, pp 234 – 238, 2020, doi: 10.18517/ijaseit.10.1.8858.