Klasifikasi *Sentiment Analysis* pada *Review* Buku Novel Berbahasa Inggris dengan Menggunakan Metode *Support Vector Machine* (SVM)

Chandra Gilang Kencana[1], Yuliant Sibaroni[2]
[1,2]Fakultas Informatika, Universitas Telkom, Bandung
[1]cgkcgkcgk@students.telkomuniversity.ac.id, [2]yuliant@telkomuniversity.ac.id,

Abstrak

Buku novel merupakan suatu karya sastra berbentuk prosa naratif yang panjang, yang memiliki rangkaian cerita tentang kehidupan seorang tokoh dan orang-orang di sekitarnya dengan menonjolkan sifat dan watak dari setiap tokoh pada novel tersebut. Dengan banyaknya *review* yang muncul dari pendapat pembaca, maka semakin sulit untuk menemukan *review* yang sesuai dengan pilihan konsumen. Ini yang menjadi suatu permasalahan yang dimana, konsumen tidak selalu menerima *review* dari pembaca. Untuk memecahkan permasalahan tersebut, diperlukan sebuah metode yang dapat memudahkan untuk menganalisis terkait dengan *review* tersebut. Maka dari itu, solusi yang akan dilakukan yaitu dengan menerapkan klasifikasi *sentiment analysis*. *Sentiment analysis* merupakan penambangan kontekstual data berupa teks, yang bertujuan untuk menganalisa berbagai pendapat atau opini berupa isu, komentar, dan lain-lain terhadap suatu objek atau permasalahan oleh seseorang yang dimana nilai tersebut akan memiliki nilai positif atau negatif. Penelitian ini, memiliki beberapa tujuan yaitu, untuk mengetahui performansi pada sistem klasifikasi *Support Vector Machine (SVM)* yang dibangun. Kedua, untuk mengetahui performansi pada pembobotan fitur *Term Frequency-Inverse Document Frequency (TF-IDF)* dan seleksi fitur *Chi Square*. Ketiga, Untuk mengetahui performansi pada pembobotan fitur *Term Frequency (TF)* dan seleksi fitur *Chi Square.* Dari hasil eksperimen, diperoleh bahwa hasil perfomansi terbaik untuk klasifikasi *sentiment analysis* pada *review* buku novel berbahasa Inggris, yaitu pada penggunaan *kernel Gaussian RBF* untuk setiap kedua pembobotan fitur dengan seleksi fitur yang digunakan dengan nilai performansi sebesar 74.2%.

Kata kunci: ***Support Vector Machine (SVM), Sentiment Analysis, review, Term Frequency-Inverse Document Frequency (TF-IDF), Term Frequency (TF), Chi Square***

Abstract

The novel book is a literary work in the form of a long narrative prose, which has a series of stories about the life of a character and people around him by highlighting the nature and character of each character in the novel. With so many reviews that arise from the opinions of readers, it is increasingly difficult to find reviews that are in accordance with consumer choice. This is a problem in which, consumers do not always receive reviews from readers. To solve this problem, we need a method that can make it easy to analyze related to the review. Therefore, the solution that will be carried out is by applying a sentiment analysis classification. Sentiment analysis is a contextual mining of data in the form of text, which aims to analyze various opinions or opinions in the form of issues, comments, etc. on an object or problem by someone whose value will have a positive or negative value. This study, has several objectives, namely, to determine the performance of the Support Vector Machine (SVM) classification system that was built. Second, to find out the performance in weighting the Term Frequency-Inverse Document Frequency (TF-IDF) feature and Chi Square feature selection. Third, to determine the performance of the Term Frequency (TF) weighting and Chi Square feature selection. From the experimental results, it was found that the best performance results for the classification of sentiment analysis in the review of English-language novel books, namely the use of Gaussian RBF kernels for each of the two weighting features with feature selection used with a performance value of 74.2%.

Keywords: ***Support Vector Machine (SVM), Sentiment Analysis, review, Term Frequency-Inverse Document Frequency (TF-IDF), Term Frequency (TF), Chi Square***

1. Pendahuluan

Buku novel merupakan suatu karya sastra berbentuk prosa naratif yang panjang, yang dimana memiliki rangkaian cerita tentang kehidupan seorang tokoh dan orang-orang di sekitarnya dengan menonjolkan sifat dan watak dari setiap tokoh pada novel tersebut.. Untuk mengetahui apakah buku novel tersebut layak untuk dibaca atau tidak oleh konsumen baik dari kalangan umur, remaja dan dewasa, maka terpapar sebuah pendapat dari suatu *website* yang berisikan komentar berupa *review* dari setiap buku novel semua *genre* berbahasa Inggris.

Review dari konsumen buku novel tersebut, dapat berbeda satu sama lain. Dengan banyaknya *review* yang bermunculan, maka akan semakin sulit untuk menemukan informasi yang sesuai dengan pilihan konsumen. Ini akan menjadi suatu permasalahan yang dimana, pengguna tidak selalu menerima penilaian *review* dari berbagai konsumen. Untuk memecahkan permasalahan tersebut, diperlukan sebuah metode yang dapat memudahkan untuk menganalisis terkait tentang *review* buku novel tersebut. Solusi yang akan dilakukan adalah dengan menerapkan klasifikasi *sentiment analysis*.

Text mining merupakan suatu proses mengeksplorasi dan menganalisis sejumlah besar data teks tidak terstruktur yang dibantu oleh perangkat lunak agar mengidentifikasikan konsep, topik, kata kunci, dan atribut lainnya. *Text mining* juga menjadi lebih praktis bagi para ilmuwan dan pengguna lainnya, karena untuk mengembangkan suatu platform data yang besar, dapat menganalisis suatu kumpulan data yang tidak terstruktur secara besar-besaran. *Text mining* memiliki tujuan, yaitu dapat melakukan klasifikasi *clustering, question and answering, sentiment analysis,* dan komparasi teks [1].

Sentiment analysis merupakan penambangan kontekstual teks yang dapat mengidentifikasi dan mengekstrak informasi subjektif dalam sumber berupa *review* baik itu pendapat maupun opini. Data berupa kumpulan *review* buku novel berbahasa Inggris yang akan dianalisis, merupakan suatu data berupa teks yang dapat diambil dari kolom *review* dari *netizen* di suatu *website* dan juga dari berbagai sumber unggahan dari pengguna yang terkait akan opini terhadap *review* buku novel. Jadi, dapat dengan mudah utnuk mengambil kesimpulan dari *review* para pengguna dengan menggunakan klasifikasi *sentiment analysis. Sentiment analysis* dapat bertujuan untuk menganalisa suatu *review* yang berisikan pendapat maupun opini dengan mengklasifikasikan apakah setiap *review* tersebut bersifat positif atau negatif [2].

Pada penelitian ini, memiliki tujuan yang akan dicapai. Pertama, untuk mengetahui performansi pada sistem klasifikasi *Support Vector Machine (SVM)* yang dibangun. Kedua, untuk mengetahui performansi pada pembobotan fitur *Term Frequency-Inverse Document Frequency (TF-IDF)* dan seleksi fitur *Chi Square*. Ketiga, Untuk mengetahui performansi pada pembobotan fitur *Term Frequency (TF)* dan seleksi fitur *Chi Square.* Alasan metode klasifikasi *Support Vector Machine (SVM)* terpilih, karena adanya penelitian sebelumnya yang menggunakan metode klasifikasi *Support Vector Machine (SVM)* untuk studi kasus *review film* berbahasa Inggris, mampu menghasilkan nilai performansi yang sangat baik sebesar 92,89% [3] .

Pada penelitian tentang klasifikasi *sentiment analysis* yang menggunakan metode *Support Vector Machine (SVM)* dan ekstraksi fitur *Doc2Vec* untuk studi kasus *review film* berbahasa Inggris, didapatkan bahwa dalam menggunakan *Support Vector Machine* (SVM) mampu mengklasifikasikan data *review film* dengan baik. Dengan mengatur komposisi data *training* dapat mempengaruhi akurasi. Hal ini disebabkan, jika semakin tinggi komposisi data *training*, maka jumlah variasi data *training* akan semakin banyak. Sehingga, sistem dapat melakukan klasifikasi lebih baik dan akan berpengaruh pada nilai performansi yang didapat [3] .

Sedangkan, pada penelitian tentang *sentiment analysis* dari *review* produk *smartphones* berbahasa Inggris yang menggunakan metode *Support Vector Machine (SVM).* Didapatkan jika menggunakan klasifikasi *Support Vector Machine (SVM)* dapat memperoleh nilai akurasi yang lebih tinggi jika dibandingkan dengan klasifikasi lainnya. Hal ini dikarenakan adanya pengaruh pada data *review smartphones* yang diolah. Jika semakin banyak data *review* tersebut, maka akan berpengaruh pada nilai akurasi yang diperoleh [4] .

2. Studi Terkait

Beberapa metode klasifikasi *sentiment analysis* selain *Support Vector Machine (SVM)* yaitu, *Naïve Bayes Classification (NBC), Decision Tree (DT),* dan *K-Nearest Neighbor (KNN)* [5]. Pada penelitian tersebut, metode klasifikasi *Support Vector Machine (SVM)* dapat memperoleh nilai akurasi yang lebih baik jika dibandingkan dengan metode klasifikasi *Naïve Bayes Classification (NBC), Decision Tree (DT),* dan *K-Nearest Neighbor (KNN)* dengan nilai sebesar 78,18%. Hal ini dikarenakan, pada penelitian tersebut telah membandingkan pada keempat metode klasifikasi yang digunakan pada studi kasus *review* barang berbahasa Indonesia.

Kemudian, pada penelitian untuk studi kasus pada *review film* dan komentar *twitter* dengan menggunakan klasifikasi *Support Vector Machine* tanpa *kernel* dan menggunakan *kernel Gaussian RBF* serta klasifikasi *Naïve Bayes Classification (NBC),* didapat bahwa menggunakan klasifikasi Support Vector Machine dengan *kernel Gaussian RBF* mampu mendapatkan nilai akurasi tertinggi sebesar 74,74% untuk data *review film* dan 78,18% untuk data komentar pada *twitter* jika dibandingkan dengan klasifikasi *Naïve Bayes Classification (NBC)* dan *Support Vector Machine (SVM)* tanpa *kernel*. Hal ini dikarenakan, penggunaan *kernel* pada *Support Vector Machine (SVM)* dapat mempengaruhi meningkatnya nilai akurasi yang didapat [6] .

Berikutnya adalah pada penelitian studi kasus tentang komparasi teks bersifat negatif yang berisikan data *review film, hotel, product, sports, education, news forum, educational, political news,* dan GPS dengan menggunakan klasifikasi *Support Vector Machine (SVM), Naïve Bayes Classification (NBC),* dan *Decision Tree (DT).* Didapatkan bahwa nilai akurasi pada ketiga metode klasifikasi tersebut, lebih unggul jika menggunakan klasifikasi *Support Vector Machine (SVM)* dengan nilai sebesar 66,4 %. Hal ini disebabkan pada penelitian tersebut telah membandingkan pada ketiga klasifikasi tersebut dengan tahapan ekstrasi fitur berupa *Term Frecquency-*

Inverse Document Frequency (TF-IDF), *Doc2Vec*, dan *Lexicon Based Features*. Maka dari itu, klasifikasi *Support Vector Machine (SVM)* merupakan klasifikasi yang terbaik untuk studi kasus komparasi teks serta tahapan ekstraksi fitur sangat berpengaruh pada nilai akurasi yang didapat. [7] .

Pada penelitian yang mencoba untuk membandingkan studi kasus dengan cara menggunakan *preprocessing* dengan yang tidak, didapatkan bahwa *preprocessing* mampu meningkatkan akurasi mesin yang dibuat dengan selisih sebesar 2,76% [8] . Karena, hal tersebut, pada penelitian ini akan menggunakan *preprocessing* untuk meningkatkan akurasi pada sistem mesin yang dibuat serta sangat berpengaruh pada berbagai studi kasus yang akan diuji maupun diteliti.

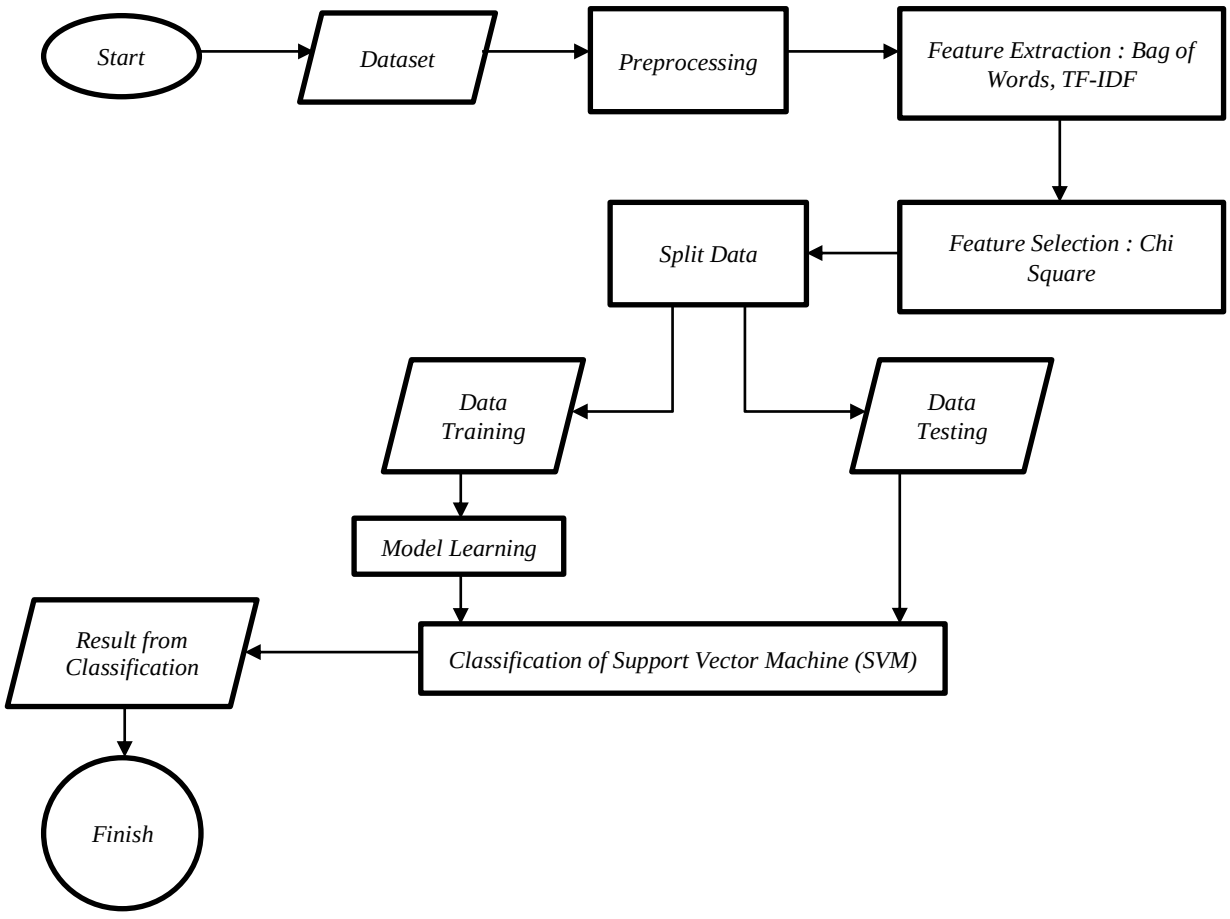
Pada penelitian ini yang mempunyai studi kasus *sentiment analysis* pada opini iklan *website* berbahasa Indonesia dengan membandingankan fitur ekstrasi *Term Frecquency-Inverse Document Frequency (TF-IDF)* dengan *Term Frequency-iGini* (*TF-iGini)* pada metode klasifikasi *Multinomial Naïve Bayes (MNB)*, menunjukkan bahwa dalam penggunaan ekstraksi fitur *Term Frecquency-Inverse Document Frequency* (*TF-IDF)* lebih baik daripada *Term Frequency-iGini* (*TF-iGini)*. Hal ini dikarenakan, jika menggunakan ekstraksi fitur *Term Frequency-iGini* (*TF-iGini)* pada studi kasus *sentiment analysis*, rata-rata nilai akurasi yang dicapai kurang dari 5% daripada menggunakan *Term Frecquency-Inverse Document Frequency* (*TF-IDF)* [9] .

Pada penelitian yang menggunakan statistik *Chi Square*, ekstraksi fitur *Term Frecquency-Inverse Document Frequency* (*TF-IDF)*, dan klasifikasi *Support Vector Machine (SVM)* untuk studi kasus *sentiment analysis* pada *review film*, didapatkan bahwa nilai akurasi dapat lebih baik sebesar 80,2% jika menggunakan statistik *Chi Square*, ekstraksi fitur *Term Frecquency-Inverse Document Frequency* (*TF-IDF)*. Sedangkan, jika tanpa menggunakan statistik *Chi Square*, akan memberikan nilai akurasi yang lebih rendah yaitu sebesar 68,7% [10] .

3. Sistem yang Dibangun

3.1 Rancangan Sistem

Sistem yang akan dibangun pada penelitian ini merupakan sistem yang dapat melakukan klasifikasi pada *review* buku novel berbahasa Inggris secara otomatis. Berikut gambaran sistem yang akan dibangun:



Gambar 3.1 Diagram Sistem pada Klasifikasi *Sentiment Analysis Review* Buku Novel Berbahasa Inggris

3.2 Pengumpulan *Dataset*

Dataset yang akan digunakan pada penelitian ini diambil dari situs *website* https://sites.google.com/eng.ucsd.edu/ucsdbookgraph/home, dengan berisikan *review* buku novel berbahasa Inggris semua *genre* novel. *Dataset* yang diambil ini yang pada mulanya berjumlah 15.700.000 dengan format *.json*, dikonversikan menjadi format *.csv* dengan aplikasi *DataFileConverter.exe* dan hanya diambil sebagian data dengan jumlah 5213. Hal ini dikarenakan, banyak data berisikan *review* yang bukan berbahasa Inggris.

Untuk melakukan pelabelan kelas positif dan negatif, penulis memodifikasikan dari rating 0-5, yang dimana jika nilai rating < 3, maka *review* tersebut adalah kelas negatif. Sedangkan, jika nilai rating > 3 , maka *review* tersebut adalah kelas positif. Untuk rating 3, dilakukan pelabelan secara manual yang dimana *review* tersebut dapat masuk ke dalam kelas positif dan kelas negatif [11] . Untuk *dataset* dengan kelas negatif memiliki jumlah sebesar 2130 dan untuk kelas positif memiliki jumlah sebesar 3093. Berikut merupakan tabel beberapa contoh dataset yang akan diolah :

Tabel 3.1 Beberapa Contoh Dataset yang Akan Diolah dan Sudah Diberikan Pelabelan

Score	*Review Novel*
Negative	*'I would hate to have to see the movie without having read it first...'*
Negative	*This is the best harry potter book so far followed closely by book 6.'*
Positive	*I really enjoyed the Lunar Chronicles and Winter was a very good ending Winter was really enjoyable character.*
Positive	*LOVED IT!! but at the same time I think it was all that needed to be said about a few of them Great stuff*

3.3 *Preprocessing*

Setelah melakukan tahapan pengumpulan dataset yang sudah diberikan pelabelan, berikutnya adalah melakukan tahapan *preprocessing* yang berfungsi untuk mengubah data yang tidak terstruktur menjadi terstruktur. Sehingga, dapat digunakan pada tahap selanjutnya. Pada tahapan *preprocessing*, terdiri dari beberapa bagian, yaitu :

3.3.1 *Case Folding*

Case Folding merupakan tahapan pada *preprocessing* yang paling sederhana dan efektif meskipun sering diabaikan. Tujuan *case folding* salah satunya adalah mengganti seluruh huruf kapital dari suatu kalimat menjadi huruf kecil. Selain itu, *case folding* juga mampu memperoleh fitur yang lebih efisien dalam format yang sama. [12] .

3.3.2 *Tokenizing*

Tokenizing adalah proses untuk memisahkan teks pada suatu kalimat menjadi potongan-potongan kata yang disebut sebagai *token*. Tujuan pada *tokenizing* adalah membentuk token dari suatu kalimat menjadi sebuah fitur yang akan digunakan dalam tahapan analisis model *Bag of Words*. [13] .

3.3.3 *Stopword Removal*

Stopword Removal adalah tahapan untuk menghilangkan fitur yang tidak memiliki makna dan dianggap tidak penting dari hasil *tokenizing*. Contoh pada fitur tersebut yang dihilangkan adalah "*off*", "*the*", "*to*", "*from*", "*and*", dan keseluruhan daftar *library stopword* yang didapat berdasarkan *library* NLTK *python*. [14]

3.3.4 *Stemming*

Stemming adalah tahapan untuk menghilangkan infleksi suatu kata ke bentuk dasarnya. Contoh pada pengelompokkan kata-kata tersebut adalah kata *"consultant", "consulting", dan "consultantative"* yang akan ditransformasi menjadi kata *"consult"*. Pada proses ini, penulis menggunakan *library PorterStemmer*. Hal ini

dikarenakan, *library PorterStemmer* adalah sebuah *library* yang paling umum digunakan dan mempunyai algoritma yang paling intensif secara komputasi [8] .

3.4 Ekstraksi Fitur

Setelah melakukan tahapan *preprocessing*, maka tahapan berikutnya adalah melakukan ekstraksi fitur. Ekstraksi fitur mempunyai tujuan untuk mengubah daftar dalam sebuah fitur menjadi bentuk ciri suatu model. Pada tahapan ekstraksi fitur, terdiri dari beberapa bagian, yaitu :

3.4.1 Fitur *Bag Of Words*

Bag of words merupakan sebuah model ekstraksi fitur yang digunakan dalam pengolahan bahasa alami dan pencarian informasi. Tujuan dari *bag of words*, adalah untuk menganalisa sebuah fitur pada kalimat dan menghitung nilai kemunculan fitur yang diperoleh. [15] .

Tabel 3.2 Contoh Kalimat di setiap Dokumen

Dokumen	Kalimat
D1	*i loved this book*
D2	*this book is absolutely good*
D3	*the best book ever*

Tabel 3.3 *Bag of Words*

TF	*i*	*loved*	*this*	*book*	*Is*	*absolutely*	*good*	*the*	*best*	*ever*
D1	1	1	1	1	0	0	0	0	0	0
D2	0	0	1	1	1	1	1	0	0	0
D3	0	0	0	1	0	0	0	1	1	1

3.4.2 Pembobotan Fitur *Term Frequency-Inverse Document Frequency (TF-IDF)*

TF (Term Frequency) merupakan sebuah metode frekuensi dari kemunculan sebuah *term* dalam teks atau dokumen. Sedangkan, *IDF (Inverse Document Frequency)* merupakan sebuah metode perhitungan yang dimana, *term* didistribusikan secara luas pada suatu dokumen. Tahapan pada pembobotan fitur TF-IDF hanya dapat mempresentasikan kemampuan dari fitur untuk membedakan suatu teks atau dokumen secara bersangkutan saja [16] . *TF-IDF* memiliki sebuah konsep maupun persamaan pada berikut ini :

Persamaan TF :

$$TF = {f_{t,d}} \Big/ {\sum_{t' \in d} f_{t,d}} \tag{3.1}$$

Pada definisi konsep di *TF* ini, memiliki keterangan :

d = dokumen
t = *term*
$f_{t,d}$ = Jumlah kata pada suatu *term* (t) di setiap dokumen (d)
$\sum_{t' \in d} f_{t,d}$ = Jumlah dokumen (d) yang memuat *term* (t)
TF = Hasil dari *Term Frequency* (TF)

Persamaan IDF :

$$IDF = \log\left({}^{N}/_{df_t}\right) \quad (3.2)$$

Pada definisi konsep di *IDF* ini, memiliki keterangan :

N = Jumlah dokumen keseluruhan
df_t = Jumlah kata pada *term* (*t*) yang ada pada seluruh dokumen (*N*)
IDF = Hasil dari *Inverse Document Frecuency* (*IDF*)

Sehingga, untuk persamaan *TF-IDF*, memiliki persamaan pada berikut ini :

$$TF\text{-}IDF = TF \; x \; IDF \quad (3.3)$$

Dengan keterangan bahwa hasil dari persamaan *TF* dan persamaan *IDF*, dikalikan hingga menghasilkan nilai TF-IDF tersebut.

Pada sebuah metode yang digunakan pada *TF-IDF* tersebut, maka akan menghasilkan sebuah fitur dengan nilai yang terbesar hingga yang terkecil untuk dapat mengetahui seberapa pentingnya dalam fitur tersebut pada sebuah penelitian ini, Berikut, merupakan sebuah ilustrasi sebuah fitur yang didapat. Dengan diberikan nilai kemunculan setiap fitur yang ada pada dokumen di tabel 3.2 :

Tabel 3.4 Ilustrasi Perhitungan *Term Frequency (TF)* berdasarkan Konsep Tahapan (3.1)

TF	*i*	*loved*	*this*	*book*	*is*	*absolutely*	*good*	*The*	*best*	*ever*
D1	1/4	1/4	1/4	1/4						
D2			1/5	1/5	1/5	1/5	1/5			
D3				1/4				¼	1/4	1/4

Keterangan :

1/4 : 0,25
1/5 : 0,2

Tabel 3.5 Ilustrasi Perhitungan *Inverse Document Frequency (IDF)* berdasarkan konsep Tahapan (3.2)

IDF	*i*	*loved*	*this*	*book*	*is*	*absolutely*	*good*	*The*	*best*	*ever*
D1	Log (3/1)	Log (3/1)	Log (3/2)	Log (3/3)						
D2			Log (3/2)	Log (3/3)	Log (3/1)	Log (3/1)	Log (3/1)			
D3				Log (3/3)				Log (3/1)	Log (3/1)	Log (3/1)

Keterangan :

Log (3/1) : 0,477
Log (3/2) : 0,176
Log (3/3) : 0

Tabel 3.6 Ilustrasi Perhitungan *Term Frequency-Inverse Document Frequency (TF-IDF)* berdasarkan Konsep Tahapan (3.3)

TF-IDF	*i*	*loved*	*this*	*book*	*is*	*absolutely*	*good*	*the*	*best*	*ever*
D1	0.12	0.12	0.04	0	0	0	0	0	0	0
D2	0	0	0,03	0	0,09	0,09	0,09	0	0	0
D3	0	0	0	0	0	0	0	0,12	0,12	0,12

Berdasarkan hasil perhitungan *TF-IDF* diatas, diambil sebuah kesimpulan bahwa *TF-IDF* dapat bernilai 0 apabila suatu kata tidak muncul pada suatu dokumen atau kata tersebut selalu muncul pada setiap dokumen. Dalam menjalankan tahapan ekstrasi fitur, maka nilai *TF-IDF* pada suatu kata akan dilihat terlebih dahulu. Jika kata tersebut berulang lebih dari persentase dokumen yang di tentukan, maka kata tersebut tidak akan disertakan dalam *TF-IDF*.

3.5 Seleksi Fitur

Setelah melakukan tahapan ekstraksi fitur, maka tahapan selanjutnya adalah melakukan seleksi fitur. Tujuan pada seleksi fitur adalah memilih daftar pada suatu kata yang diproses oleh ekstraksi fitur. Dalam pemilihan suatu fitur, berdasarkan nilai fitur yang terbaik. Jika pada fitur tersebut mempunyai nilai terbesar, maka fitur tersebut dianggap penting, sedangkan jika pada fitur tersebut mempunyai nilai terkecil, maka fitur tersebut dianggap fitur yang tidak penting. Pada tahapan seleksi fitur, terdiri dari beberapa bagian, yaitu :

3.5.1 *Chi Square*

Chi Square merupakan metode dengan jenis uji komparatif non parametris yang dilakukan pada dua variabel, dimana skala data kedua variabel adalah nominal. Apabila dari 2 variabel, terdapat 1 variabel dengan skala nominal maka dilakukan uji *chi square* dengan merujuk bahwa harus menggunakan uji pada derajat yang terendah [17] . Berikut ini merupakan sebuah definisi konsep pada *chi square* :

$$\chi^2(t,c) = \frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)} \quad (3.4)$$

Pada definisi konsep *Chi Square* ini, memiliki keterangan :

t = Fitur atau kata
c = Kelas atau kategori
N = Jumlah seluruh dokumen
A = Jumlah banyaknya dokumen pada kategori (*c*) yang memuat (*t*)
B = Jumlah banyaknya dokumen bukan kategori (*c*) yang memuat (*t*)
C = Jumlah banyaknya dokumen pada kategori yang tidak memuat (*t*)
D = Jumlah banyaknya dokumen bukan kategori yang tidak memuat (*t*)

Dengan metode yang digunakan pada *Chi Square* tersebut, maka akan menghasilkan sebuah fitur dengan nilai yang terbesar hingga yang terkecil yang telah dilakukan dengan cara memberikan seleksi berdasarkan seberapa pentingnya fitur tersebut. Berikut, ini merupakan sebuah ilustrasi tentang alur proses pada metode seleksi fitur *Chi Square*, yang dimana jika fitur yang telah melalui tahapan pembobotan fitur *TF-IDF* akan diseleksi berdasarkan persentase 20% fitur terbaik pada tabel 3.6 :

Tabel 3.7 Ilustrasi Hasil Perhitungan *Chi Square* Berdasarkan Seleksi Fitur 20% Terbaik

No	Kata	Hasil Perhitungan *Chi Square*
1	*I*	0,043
2	*Loved*	0,043

Tabel 3.8 Matriks *TF-IDF* yang Terpilih Berdasarkan Seleksi Fitur 20% Terbaik

TF-IDF	*I*	*Loved*
D1	0.12	0.12
D2	0	0
D3	0	0

3.6 Klasifikasi *Support Vector Machine* (SVM)

Support Vector Machine pertama kali dikembangkan oleh Boser, Guyon, Vapnik ini sebenarnya merupakan kombinasi dari teori-teori yang sudah ada seperti *margin*, *hyperplane*, dan *kernel*. Konsep dasar dari SVM adalah dengan mencari sebuah hyperplane dengan *margin* terbesar maupun yang terkecil berdasarkan jarak antara letak suatu data dengan *margin* [18] . Metode SVM secara umum telah memberikan solusi lebih baik dibanding model konvensional seperti neural network. *Hyperplane* yang baik bisa didapatkan dengan memaksimalkan jarak *margin*.

Berikut, diasumsikan jika pada kelas -1 dan +1 dapat terpisah secara sempurna oleh *hyperplane*, yang dimana dapat diberikan sebuah definisi :

$$\vec{w}.\vec{x} + b = 0 \tag{3.5}$$

Jika $\vec{w}$ berada di kelas +1, maka dapat dituliskan sebuah definisi berikut ini :

$$\vec{w}.\overrightarrow{x_t} + b \geq +1 \tag{3.6}$$

Sedangkan, Jika $\vec{w}$ berada di kelas -1, maka dapat dituliskan sebuah definisi berikut ini :

$$\vec{w}.\overrightarrow{x_t} + b \leq -1 \tag{3.7}$$

Ilustrasi cara kerja Support Vector Machine
Misalkan terdapat atribut-atribut seperti pada tabel berikut :

Tabel 3.9 Kelas *Hyperplane*

X1	X2	Kelas (y)
1	1	Positif
1	-1	Negatif
-1	-1	Negatif
-1	-1	Negatif

Sehingga, didapatkan sebuah persamaan *hyperplane* berikut ini :

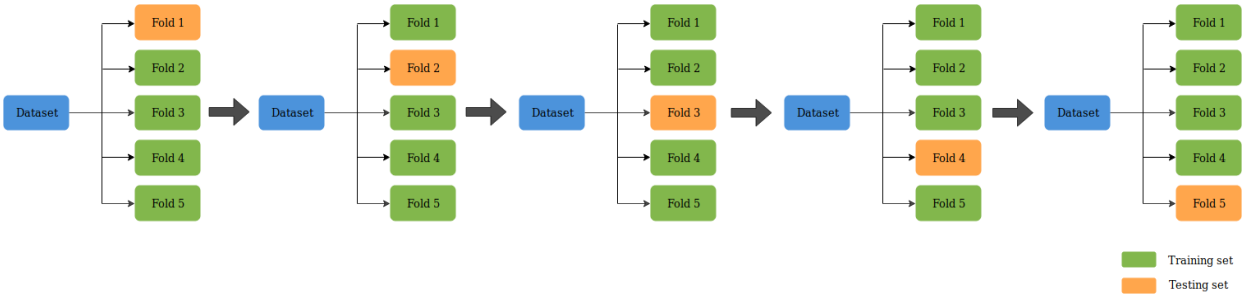
$$\begin{gathered} w_1.x_1 + w_2.x_2 + b = 0 \\ x_1 + x_2 - 1 = 0 \\ x_2 = 1 - x_1 \end{gathered} \tag{3.8}$$

Tabel 3.10 *Support Vector Machine* dengan Jenis *Kernel* serta Definisi Rumus

Jenis Kernel	Definisi Rumus
Linear	$K(\vec{x_i}, \vec{x_j}) = \vec{x_i}^t, \vec{x_j}$
Gaussian RBF	$K(\vec{x_i}, \vec{x_j}) = exp\left(-\frac{\|\|(\vec{x_i} - \vec{x_j})\|\|^2}{2\sigma^2}\right)$

3.7 Evaluasi

Untuk menguji hasil performansi dari sistem yang dibangun dengan *dataset* yang akan digunakan, maka akan dilakukan proses validasi. Pada penelitian ini, akan menggunakan sebuah metode yang disebut sebagai *K-Fold Cross Validation*. *K-Fold Cross Validation* akan melakukan pengacakan suatu data yang sudah diolah menjadi sebuah partisi dengan cara dilakukan sebanyak *k* kali, yang dimana jika nilai *k* tersebut adalah 5, maka pada partisi *dataset* 1/5 menjadi *testing* dan pada partisi *dataset* 4/5 menjadi *training* [19] .



Gambar 3.1 Ilustrasi pada *K-Fold Cross Validation*

Tabel 3.11 *Confusion Matrix*

	Kelas Aktual	
	Negative	*Positive*
Kelas Prediksi	*True Negative (TN)*	*False Positive (FP)*
	False Negative (FN)	*True Positive (TP)*

Keterangan :

a. *True Positive* (TP) : Jumlah dokumen yang aktualnya *positive* dan diidentifikasi oleh sistem sebagai *positive*
b. *False Positive* (FP) : Jumlah dokumen yang aktualnya *negative* dan diidentifikasi oleh sistem sebagai *positive*
c. *False Negative* (FN) : Jumlah dokumen yang aktualnya *positive* dan diidentifikasi oleh sistem sebagai *negative*
d. *True Negative* (TN) : Jumlah dokumen yang aktualnya *negative* dan diidentifikasi oleh sistem sebagai *negative*

Setelah mendapatkan nilai dari *confusion matrix*, maka akan mendapatkan sebuah nilai dari *accuracy*, *precision*, dan *f1-score*.

a. *Precision* : *Precision* dapat digunakan untuk menghitung rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positf. Yaitu dengan cara menentukan nilai *True Positive (TP)* yang akan dibagi dengan nilai perjumlahan dari *True Positive (TP)* dan *False Positive (FP)*.

$$Precision = \frac{TP}{TP+FP} \quad (3.9)$$

b. *Recall* : *Recall* dapat digunakan untuk menghitung rasio dari prediksi *True Positive (TP)* yang dibagi dengan perjumlahan *True Positve (TP)* dengan *False Negative (FN)*.

$$Recall = \frac{TP}{TP+FN} \quad (3.10)$$

c. *F1-score* : *F1-score* dapat digunakan untuk menghitung rata-rata harmonik antara nilai *Precision* dengan *Recall*.

$$\text{F1} = \frac{2(Precision+Recall)}{(Precision+Recall)} \quad (3.11)$$

4. Evaluasi

4.1 Skenario Pengujian

Pada pengujian ini, terdapat 2 skenario yang akan dilakukan pada penelitian ini yakni :

Skenario 1: Untuk mengetahui performansi pada pembobotan fitur *Term Frequency-Inverse Document Frequency (TF-IDF)* dan seleksi fitur *Chi Square*

Pada skenario ini, untuk mengetahui pengaruh pada kegunaan pembobotan fitur dan seleksi fitur dapat bekerja dengan baik. Maka, pada skenario 1 ini akan menggunakan *Term Frequency-Inverse Document Frequency (TF-IDF)* dan *Chi Square* sebagai titik pengaruh untuk melihat perkembangan, lalu persentase fitur seleksinya berkisar 20%, 40%, 60%, 80%, dan 100% serta pengetahuan apakah akan meningkatkan nilai performansi yang lebih baik jika dimasukkan ke dalam alur klasifikasi *Support Vector Machine (SVM)* dengan 2 *kernel*, yaitu *Linear* dan *Gaussian RBF* dengan nilai *K-Fold* sama dengan 5.

Skenario 2: Untuk mengetahui performansi pada pembobotan fitur *Term Frequency (TF)* dan seleksi fitur *Chi Square*

Pada skenario ini, untuk mengetahui pengaruh pada kegunaan pembobotan fitur dan seleksi fitur dapat bekerja dengan baik. Maka, pada skenario 2 ini akan menggunakan *Term Frequency (TF)* dan *Chi Square* sebagai titik pengaruh untuk melihat perkembangan. lalu persentase fitur seleksinya berkisar 20%, 40%, 60%, 80%, dan 100% serta pengetahuan apakah akan meningkatkan nilai performansi yang lebih baik jika dimasukkan ke dalam alur klasifikasi *Support Vector Machine (SVM)* dengan 2 *kernel*, yaitu *Linear* dan *Gaussian RBF* dengan nilai *K-Fold* sama dengan 5.

4.2 Hasil Analisa Pengujian pada Skenario 1

Tabel 4.1 Nilai Performansi pada Skenario 1

Skenario 1			
Kernel Linear		***Kernel Gaussian RBF***	
Nilai Persentase Seleksi Fitur yang Digunakan	**Nilai Performansi pada *F1-Score***	**Nilai Persentase Seleksi Fitur yang Digunakan**	**Nilai Performansi pada *F1-Score***
20%	70.7%	20%	**74.2%**
40%	69.4%	40%	**74.2%**
60%	68.4%	60%	**74.2%**
80%	67.8%	80%	**74.2%**
100%	67.6%	100%	**74.2%**

Pada tabel 4.1 di skenario 1, pada bagian *kernel Linear* didapatkan nilai performansi *f1-score* dengan nilai sebesar 70,7% pada persentase seleksi fitur *Chi Square* yang digunakan sebesar 20%. Sedangkan, pada *kernel Gaussian RBF*, didapatkan bahwa nilai performansi *f1-score* tidak mempengaruhi seluruh nilai persentase pada seleksi fitur *Chi Square* yang digunakan meskipun mendapatkan nilai performansi 74,2%.

4.3 Hasil Analisa Pengujian pada Skenario 2

Tabel 4.2 Nilai Performansi Skenario 2

Skenario 2			
Kernel Linear		***Kernel Gaussian RBF***	
Nilai Persentase Seleksi Fitur yang Digunakan	**Nilai Performansi pada *F1-Score***	**Nilai Persentase Seleksi Fitur yang Digunakan**	**Nilai Performansi pada *F1-Score***
20%	71%	20%	**74.2%**
40%	70.3%	40%	**74.2%**
60%	70%	60%	**74.2%**
80%	69.6%	80%	**74.2%**
100%	69.2%	100%	**74.2%**

Pada tabel 4.2 di skenario 2, pada bagian *kernel Linear* didapatkan nilai performansi *f1-score* dengan nilai sebesar 71% pada persentase seleksi fitur *Chi Square* yang digunakan sebesar 20%. Sedangkan, pada *kernel Gaussian RBF*, didapatkan bahwa nilai performansi *f1-score* tidak mempengaruhi seluruh nilai persentase pada seleksi fitur *Chi Square* yang digunakan meskipun mendapatkan nilai performansi 74,2%.

5. Kesimpulan

Pada penelitian ini, pengujian yang dilakukan pada pengaruh proses pembobotan fitur dan seleksi fitur pada *Term Frequency-Inverse Document Frequency (TF-IDF)* dengan *Chi Square* maupun *Term Frequency (TF)* dengan *Chi Square* yang kemudian masuk ke dalam tahapan klasifikasi menggunakan metode *Support Vector Machine (SVM)* pada 2 *kernel* yaitu *kernel Linear* dan *Gaussian RBF* untuk mengeksekusi klasifikasi *sentiment analysis* pada *review* buku novel berbahasa Inggris.

Diperoleh bahwa hasil perfomansi terbaik untuk klasifikasi *sentiment analysis* pada *review* buku novel berbahasa Inggris, yaitu pada penggunaan *kernel Gaussian RBF* untuk setiap kedua pembobotan fitur dengan seluruh nilai persentase seleksi fitur yang digunakan dengan nilai performansi terbaik sebesar 74.2%.

Selain itu, dari hasil penelitian pada kedua proses pembobotan fitur dalam satu seleksi fitur serta penggunaan *kernel Linear* pada klasifikasi *Support Vector Machine (SVM)*, diperoleh bahwa hasil performansi yang diperoleh sebesar 70,7% jika menggunakan pembobotan fitur *Term Frequency-Inverse Document Frequency (TF-IDF)* dan seleksi fitur *Chi Square*. Sedangkan, hasil performansi yang diperoleh sebesar 71% % jika menggunakan pembobotan fitur *Term Frequency (TF)* dan seleksi fitur *Chi Square*. Hal ini didapatkan, bahwa

dengan penggunaan *kernel Gaussian RBF* untuk kedua proses pembobotan fitur dengan seluruh nilai persentase seleksi fitur yang digunakan, mampu meningkatkan nilai performansi lebih baik jika dibandingkan dengan penggunaan *kernel Linear*.

Daftar Pustaka

[1] J. Ipmawati, Kusrini, and E. Taufiq Luthfi, "Komparasi Teknik Klasifikasi Teks Mining Pada Analisis Sentimen," *Indones. J. Netw. Secur.*, vol. 6, no. 1, pp. 28–36, 2017.

[2] O. Somantri, D. Apriliani, J. T. Informatika, P. Harapan, and B. Tegal, "Support Vector Machine Berbasis Feature Selection Untuk Sentiment Analysis Kepuasan Pelanggan Terhadap Pelayanan Support Vector Machine Based on Feature Selection for Sentiment Analysis Customer Satisfaction on Culinary," vol. 5, no. 5, pp. 537–548, 2018.

[3] W. C. Widyaningtyas, A. Adiwijaya, and S. Al Faraby, "Klasifikasi Sentiment Analysis Pada Review Film Berbahasa Inggris Dengan Menggunakan Metode Doc2vec Dan Support Vector Machine (svm)," *eProceedings Eng.*, vol. 5, no. 1, 2018.

[4] E. Tyagi and A. K. Sharma, "Sentiment Analysis of Product Reviews using Support Vector Machine Learning Algorithm," *Indian J. Sci. Technol.*, vol. 10, no. 35, pp. 1–9, 2017.

[5] D. J. Haryanto, L. Muflikhah, and M. A. Fauzi, "Analisis Sentimen Review Barang Berbahasa Indonesia Dengan Metode Support Vector Machine Dan Query Expansion," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 2909–2916, 2018.

[6] B. Jadav and V. Vaghel, "Sentiment Analysis using Support Vector Machine based on Feature Selection and Semantic Analysis," *Int. J. Comput. Appl.*, vol. 146, no. 13, pp. 26–30, 2016.

[7] S. Almatarneh and P. Gamallo, "Comparing supervised machine learning strategies and linguistic features to search for very negative opinions," *Inf.*, vol. 10, no. 1, 2019.

[8] G. Angiani *et al.*, "A comparison between preprocessing techniques for sentiment analysis in Twitter," *CEUR Workshop Proc.*, vol. 1748, no. Ml, 2016.

[9] M. A. Imtiyazi, M. A. Bijaksana, and M. Tech, "Sentiment Analysis Berbahasa Indonesia Menggunakan Improved Multinomial Naive Bayes Indonesian Sentiment Analysis Using Improved Multinomial Naïve Bayes ABSTRAKSI Penggunaan Multinomial Naïve Bayes sebagai classifier dalam kasus sentiment analysis sudah j," vol. 2, no. 2, pp. 6331–6335, 2015.

[10] U. I. Larasati, M. A. Muslim, R. Arifudin, and A. Alamsyah, "Improve the Accuracy of Support Vector Machine Using Chi Square Statistic and Term Frequency Inverse Document Frequency on Movie Review Sentiment Analysis," *Sci. J. Informatics; Vol 6, No 1 Mei 2019*, vol. 6, no. 1, pp. 138–149, 2019.

[11] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," *ACL 2007 - Proc. 45th Annu. Meet. Assoc. Comput. Linguist.*, pp. 440–447, 2007.

[12] S. Mujilahwati, "Pre-Processing Text Mining Pada Data Twitter," *Semin. Nas. Teknol. Inf. dan Komun.*, vol. 2016, no. Sentika, pp. 2089–9815, 2016.

[13] V. S and J. R, "Text Mining: open Source Tokenization Tools – An Analysis," *Adv. Comput. Intell. An Int. J.*, vol. 3, no. 1, pp. 37–47, 2016.

[14] S. Vijayarani, J. Ilamathi, and M. Nithya, "Preprocessing Techniques for Text Mining-An Overview," *Int. J. Comput. Sci. Commun. Networks*, vol. 5, no. 1, pp. 7–16, 2015.

[15] S. A. Memon, F. Akthar, T. Mahmood, M. Azeem, and Z. Shaukat, "3D shape retrieval using bag of word approaches," *2019 2nd Int. Conf. Comput. Math. Eng. Technol. iCoMET 2019*, no. March, pp. 1–7, 2019.

[16] B. Das and S. Chakraborty, "An Improved Text Sentiment Classification Model Using TF-IDF and Next Word Negation," 2018.

[17] C. F. Suharno, M. A. Fauzi, and R. S. Perdana, "Klasifikasi Teks Bahasa Indonesia Pada Dokumen Pengaduan Sambat Online Menggunakan Metode K-Nearest Neighbors Dan Chi-square," *Syst. Inf. Syst. Informatics J.*, vol. 3, no. 1, pp. 25–32, 2017.

[18] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Application of Support Vector Machine in Bioinformatics," *Proceeding Indones. Sci. Meet. Cent. Japan*, 2003.

[19] A. F. Hidayatullah, "Analisis Sentimen dan Klasifikasi Kategori Terhadap Tokoh Publik Pada Twitter," *Semin. Nas. Inform. 2014 (semnasIF 2014)*, vol. 2014, no. semnasIF, p. A-1, 2014.