KLASIFIKASI SENTIMEN *REVIEW* FILM MENGGUNAKAN ALGORITMA *SUPPORT VECTOR MACHINE*

SENTIMENT CLASSIFICATION OF MOVIE REVIEWS USING ALGORITHM SUPPORT VECTOR MACHINE

Irene Mathilda Yulietha[1], Said Al Faraby[2], Adiwijaya[3]

[1,2,3] Prodi S1 Teknik Informatika, Fakultas Teknik, Universitas Telkom
[1]irenemys@students.telkomuniversity.ac.id, [2]saidalfaraby@telkomuniversity.ac.id,
[3]adiwijaya@telkomuniversity.ac.id

Abstrak

Dengan kemajuan di bidang teknologi, seluruh informasi tentang semua film sudah tersedia di Internet. Jika informasi tersebut diolah dengan baik maka akan diperoleh kualitas dari informasi tersebut. Tugas Akhir ini bertujuan untuk menjelaskan klasifikasi sentimen pada dokumen *review* film. Satu hal yang penting dalam sebuah *review* atau ulasan yaitu opini yang terkandung di dalamnya. Metode yang digunakan pada penelitian ini adalah *Support Vector Machine* (SVM). Metode ini dipilih karena mampu mengklasifikasikan data berdimensi tinggi sesuai dengan data yang digunakan pada Tugas Akhir ini yaitu berupa teks. Pengklasifikasi *Support Vector Machine* adalah teknik *machine learning* yang populer untuk klasifikasi teks karena dapat melakukan klasifikasi dengan cara belajar dari sekumpulan contoh dokumen yang telah diklasifikasi sebelumnya dan juga mampu memberikan hasil yang baik. Dari uji skenario yang dilakukan, dapat diketahui bahwa algoritma *Support Vector Machine* dapat digunakan untuk kasus *review* film dengan nilai F1-Score sebesar 84.9%.

Kata kunci : analisis sentimen, *support vector machine, review* film, klasifikasi

Abstract

With the advancement in technology, all information about all movies is available on the internet. If the information is processed properly, it will get the quality of the information. This research proposed to do the classification of sentiment on the document movie reviews. One thing that is important in a review is the opinion. The method used in this research is Support Vector Machine (SVM). This method is chosen because it is able to classify high dimensional data in accordance with the data used in this research in the form of text. Classifier Support Vector Machine is a popular machine learning technique for text classification because it can classify by learning from a collection of document samples that have been previously classified and also able to give a good result. From the testing scenario, it can be seen that the Support Vector Machine algorithm can be used for movie review case with the F1-Score value of 84.9%.

Keywords: analysis sentiment, support vector machine, movie reviews, classification

1. Pendahuluan

1.1. Latar Belakang

Suatu kebutuhan atas informasi didorong oleh banyaknya penelitian dan teknologi. Informasi yang bermunculan saat ini sangat tidak terbatas jumlahnya, sehingga dibutuhkan suatu penyajian dari informasi namun tidak mengurangi nilai yang terkandung di dalamnya [1]. Dengan berkembangnya dunia *web*, semakin banyak orang yang menuliskan opini mereka tentang sebuah produk atau jasa. Dalam beberapa tahun terakhir, sejumlah besar situs *web* telah memungkinkan penggunanya untuk berkontribusi, memodifikasi, dan meningkatkan konten [2]. Setiap pengguna internet aktif dengan bebas menyampaikan ekspresi mereka atau pendapat pribadi tentang suatu topik tertentu. Salah satu contohnya yaitu *review* film.

Saat ini masyarakat umum dapat memberikan pendapat dan opininya melalui jejaring sosial [3] [4]. Demikian juga dalam industri film [4]. Dengan kemajuan teknologi yang sangat pesat sekarang ini, seluruh informasi tentang film sudah tersedia di internet. Semakin banyak informasi yang disediakan di internet, maka akan semakin sulit juga untuk menemukan informasi yang sesuai dengan kebutuhan konsumen. Jika informasi tersebut diolah dengan baik maka dapat diperoleh nilai atau kualitas dari informasi tersebut melalui internet. Dari adanya pengklasifikasian tentang *review* film akan memudahkan pengguna untuk mengetahui kualitas film berdasarkan pengalaman penonton lain.

Dari *review* film yang digunakan pada penelitian ini, tantangan yang dihadapi yaitu menangani masalah kata negasi dalam *review*. Kata negasi merupakan suatu pernyataan yang memiliki sifat penyangkalan atau membalikkan nilai kebenaran. Biasanya kata negasi disertakan dengan kata *'not'*. Kata negasi sendiri memiliki pengaruh pada suatu *review* karena mampu mengubah nilai sentimen maksudnya jika suatu kata sebenarnya bermakna positif namun dengan menggunakan kata negasi maka maknanya berubah menjadi negatif.

Analisis sentimen ini memiliki tugas untuk melihat pendapat atau kecenderungan opini terhadap suatu

film, apakah cenderung beropini positif atau negatif. Penelitian analisis sentimen pada Tugas Akhir ini dilakukan dengan menggunakan pendekatan dalam *machine learning* yang dikenal dengan *Support Vector Machine* dan dikhususkan pada *review* film dengan ulasan bahasa Inggris. Keterlibatan *machine learning* pada penelitian ini karena *machine learning* dapat digunakan pada segala bidang yang telah terkomputerisasi, yaitu bidang *text analysis, image processing, finance, search and recommendation engine, speech understanding,* dan sebagainya. Secara khusus *machine learning* pada kasus ini yaitu untuk mengklasifikasikan sebuah teks berdasarkan opini pengguna yang bersifat positif maupun negatif.

1.2. Perumusan Masalah

Berdasarkan pada permasalahan yang telah dijelaskan pada bagian latar belakang, maka rumusan masalah dapat disusun sebagai berikut:

a. Bagaimana cara mengklasifikasi data teks berupa dokumen *review* film.
b. Bagaimana menganalisis hasil klasifikasi teks dokumen *review* ini.

1.3. Tujuan

Sesuai dengan latar belakang dan rumusan masalah yang sudah dijelaskan, tujuan yang ingin dicapai dari penelitian Tugas Akhir ini adalah sebagai berikut:

a. menerapkan metode klasifikasi pada analisis sentimen *review* film.
b. menganalisis performansi hasil klasifikasi dokumen.

1.4. Batasan Masalah

Adapun batasan masalah yang dipakai pada penelitian ini adalah:

a. Data *review* diambil dari https://www.cs.cornell.edu/people/pablo/movie-review-data/
b. Dataset menggunakan bahasa Inggris.

2. Tinjauan Pustaka

2.1. Studi *Literature*

Ada beberapa penelitian yang sudah dilakukan dalam hal analisis sentimen terhadap data yang tersedia, diantaranya Hidayatullah dan Azhari (2014) menggunakan algoritma *Naïve Bayes Classifier* dan *Support Vector Machine* dalam menganalisis sentimen pada data *tweet* pemilihan presiden tahun 2014. Dari percobaan yang dilakukan, disimpulkan bahwa algoritma *Support Vector Machine* memiliki akurasi yang lebih baik dari algoritma *Naïve Bayes Classifier*. Penelitian lainnya yaitu Songbo Tan et al (2008) membandingkan *Naive Bayes*, *centroid classifier, K-Nearest Neighbor (KNN), winnow classifier,* dan *Support Vector Machine* (SVM) pada *dataset* dokumen berbahasa cina. Didapatkan bahwa SVM memiliki hasil yang terbaik. Dari penelitian diatas, *Support Vector Machine* memiliki tingkat akurasi yang lebih baik dibandingkan algoritma pembanding lainnya. Sehingga penelitian ini menggunakan *Support Vector Machine* untuk melakukan klasifikasi pada *review* film.

Beberapa penelitian terkait dengan klasifikasi sentimen diantaranya Moraes et al (2013) menganalisa sentimen pada *review* film dan beberapa produk dari Amazon menggunakan *Support Vector Machine* dan *Artificial Neural Network*). Fatimah Wulandini dan Anto Satriyo Nugroho melakukan penelitian tentang *Text Classification using Support Vector Machine for Web Mining Based Spation Temporal Analysis of the Spread of Tropical Diseases*, menggunakan 4 metode klasifikasi dan mengkomparasikan metode yang digunakan. Zhang et al (2011) melakukan analisa klasifikasi sentimen pada *review* restoran di Internet dengan bahasa Canton menggunakan pengklasifikasi *Naïve Bayes* dan *Support Vector Machine*. Setyo Budi (2017) melakukan analisa terhadap *review* film pada dataset dari *Cornell Computer Science* menggunakan algoritma *K-Means.* Ada beberapa metode klasifikasi yang biasa digunakan dalam klasifikasi data baik berupa teks maupun *speech* antara *lain Multinomial Naïve Bayes* [5], *Bayesian Network* [6], dan *Hidden Markov Model* [7] [8].

2.2. *Data Mining*

Data *mining* adalah suatu proses mencari nilai secara *valid* yang sebelumnya tidak diketahui dari suatu *database* besar dengan menggali pola dari data untuk memanipulasi dan menggunakannya untuk membuat keputusan bisnis yang penting [9].

2.3. Klasifikasi

Klasifikasi merupakan proses mencari model atau fungsi yang menunjukkan dan membedakan konsep atau kelas data. Klasifikasi memiliki tujuan untuk memperkirakan kelas yang tidak diketahui dari suatu objek [9] [10]. Proses klasifikasi dibagi menjadi 2 fase, yaitu *learning* dan *test.* Pada fase *learning* akan dibentuk sebuah model dari data yang telah diketahui kelasnya (*training set)*. Kemudian pada fase *test*, model yang sudah terbentuk sebelumnya diuji dengan sebagian data lainnya (*test set)* untuk mengetahui akurasi dari model yang dipakai. Jika akurasinya mencukupi, maka model tersebut dapat dipakai untuk memprediksi kelas data yang belum diketahui [9] [11] .

2.4. Analisis Sentimen

Analisis sentimen merupakan salah satu bidang dari ilmu komputer untuk mengidentifikasi emosi, penilaian, sikap, pendapat, evaluasi seseorang terhadap suatu produk, layanan, organisasi, individu, tokoh publik, topik, acara, ataupun kegiatan tertentu. Tujuannya adalah mengkategorikan suatu opini seseorang sebagai sentimen positif atau sentimen negatif [12].

2.5. *Text Mining*

Text Mining adalah penambangan data berupa teks untuk mendapatkan informasi baru yang tidak diketahui sebelumnya, dengan tujuan mencari kata yang mewakili isi dokumen sehingga dapat dianalisis keterhubungan antar dokumen [13]. *Text mining* merupakan penggalian data yang didapatkan dari dokumen atau kumpulan kalimat yang memiliki tujuan mencari inti dari konten dan selanjutnya dianalisa untuk didapatkan sebuah informasi.

2.6. *Text Preprocessing*

Text preprocessing merupakan tahap awal dalam mengolah data yaitu mempersiapkan data mentah sebelum memasuki proses lain dengan mengubah data dalam bentuk terstruktur. Pada umumnya, *preprocessing* data dilakukan dengan cara mengeliminasi data yang tidak sesuai dengan tujuan data menjadi terstruktur sehingga dapat diolah lebih lanjut [14].

2.7. *Feature Extraction*

Feature extraction atau ekstraksi fitur adalah proses menganalisis data input atau mengekstrak karakteristik data input yang kemudian digunakan dalam proses klasifikasi. Setelah dilakukan tahap *preprocessing*, token dalam dokumen diubah menjadi bentuk yang lebih mudah yakni berupa model vektor [13] [15]. Pada *feature extraction*, untuk membangun model vektor, perlu dilakukan proses pembobotan. Salah satunya yang sering digunakan yaitu *Term-Frequency – Inverser Document Frequency* (TF-IDF).

Pembobotan *Term-Frequency – Inverse Document Frequency* (TF-IDF) diperlukan karena pada suatu kumpulan dokumen terdiri dari dokumen yang memiliki perbedaan *term* di dalamnya. Sehingga untuk mencari informasi dari kumpulan dokumen tersebut dilakukan pembobotan untuk menghitung seberapa penting sebuah kata dalam kumpulan dokumen. Kumpulan kata didapatkan dari proses penggabungan semua hasil *preprocessing* [16]. Kemudian setiap fitur diubah menjadi bentuk matriks berdimensi (*d x n).* Matriks adalah kumpulan dari term yang terdiri dari baris dan kolom [17].

Term Frequency (TF) menyatakan jumlah *term* yang muncul dalam suatu dokumen. Semakin besar jumlah kemunculan *term* dalam dokumen maka semakin besar bobotnya. *Inverse Document Frequency* (IDF) menyatakan tingkat kepentingan suatu *term* pada kumpulan dokumen. Pada IDF, suatu *term* yang banyak muncul di berbagai dokumen dianggap tidak penting nilainya sedangkan *term* yang jarang tidak muncul dalam dokumen diperhatikan untuk pemberian bobot. Semakin sedikit jumlah dokumen yang mengandung *term* yang dimaksud, maka nilai IDF akan besar. Terdapat rumus untuk menghitung bobot (W) tiap dokumen terhadap kunci dengan rumus [18]:

$$w_{ij} = tf_{ij}\ x\ idf_j \quad (1)$$

$$tf_{ij} = f_{(ij)} \quad (2)$$

$$idf_j = \log\frac{N}{df_{(j)}} + 1 \quad (3)$$

Keterangan:

w_{ij} = bobot kata t_j terhadap dokumen d_i

tf_{ij} = jumlah kemunculan *term* t_j dalam dokumen d_i

idf_j = *term* t_j dalam koleksi dokumen

f_{ij} = frekuensi kemunculan *term* ke-i *pada* dokumen ke-j

N = jumlah dokumen yang ada dalam kumpulan dokumen

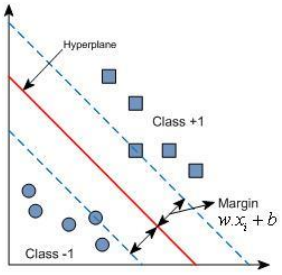
df_j = jumlah dokumen yang mengandung *term* t_j

2.8. *Supervised Learning Method*

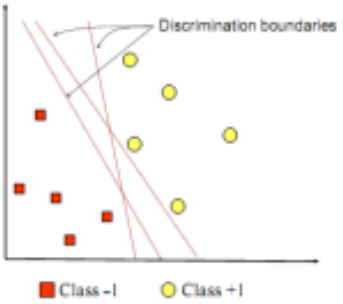
Supervised Learning Method adalah metode pembelajaran untuk mencari hubungan antara atribut input dan atribut target/kelas dari data latih untuk dijadikan model dan dapat digunakan untuk memprediksi nilai atribut target [19]. Dalam metode pembelajaran *supervised*, atribut sudah memiliki label kemudian dijadikan model. Model tersebut digunakan untuk klasifikasi pada tahap uji selanjutnya. Pada analisis sentimen, *supervised learning method* berguna dalam menentukan opini suatu produk lebih cenderung positif atau negatif.

2.9. *Support Vector Machine*

Support Vector Machine (SVM) yang dikembangkan oleh Boser, dan Guyon, Vapnik pada tahun 1992 adalah salah satu teknik klasifikasi yang memiliki tujuan untuk mencari *hyperplane* dengan *margin* terbesar [20]. *Hyperplane* yang baik didapatkan dengan memaksimalkan nilai *margin. Margin* adalah jarak antara *hyperplane* dengan *support vector* (titik yang terdekat dengan *hyperplane*).



Gambar 2- 1 Struktur Support Vector Machine [13]

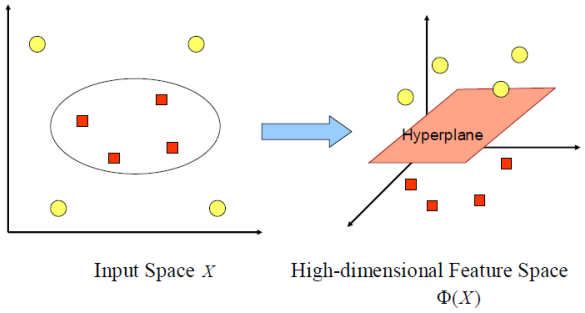


Gambar 2- 2 Hyperplane terbentuk diantara kelas -1 dan +1 [13]

Pada gambar 2-1 menunjukkan bahwa struktur SVM terdiri dari dua kelas data yaitu kelas +1 dan kelas -1. Kedua kelas pada struktur SVM tersebut dipisahkan oleh *hyperplane.* Data terdekat dengan garis *hyperplane* dibatasi oleh *margin* dan data yang terdapat pada *margin* disebut sebagai *support vector*. Garis merah menunjukkan *hyperplane* yang terbaik, yaitu terletak tepat pada tengah-tengah kedua kelas. Jarak *margin* dengan garis *hyperplane* ditentukan oleh nilai pembobot w dan bias b [13].

Gambar 2-2 menunjukkan beberapa *pattern* yang merupakan anggota dari dua buah kelas yaitu +1 dan -1. *Pattern* pada kelas +1 disimbolkan dengan warna kuning (lingkaran), sedangkan *pattern* pada kelas -1 disimbolkan dengan warna merah (kotak). Masalah pada klasifikasi di atas adalah menemukan garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut [21].

Sehingga pada prinsipnya SVM merupakan *linear classifier*, namun telah dikembangkan juga untuk menangani klasifikasi data non-*linear* dengan menggunakan konsep *kernel trick* pada ruang berdimensi lebih tinggi [21].



Gambar 2- 3 Non-Linear SVM [13]

Gambar 2-3 menunjukkan bahwa data pada kelas kuning dan data pada kelas merah berada pada *input space* berdimensi dua yang tidak dapat dipisahkan secara *linear*. Kemudian data pada *input space (X)*, memetakan fungsi Φ tiap data pada *input space* ke ruang vektor baru yang berdimensi lebih tinggi dimana kedua kelas dapat dipisahkan secara *linear* oleh *hyperplane.*

Untuk menyelesaikan problem non-*linear*, SVM dimodifikasi dengan memasukkan fungsi Kernel. Kernel trick berguna dalam pembelajaran SVM karena untuk menentukan *support vector* kita hanya cukup mengetahui fungsi kernel yang dipakai, dan tidak perlu mengetahui wujud dari fungsi non-*linear* Φ. Berikut jenis kernel yang umum digunakan pada Tabel 2-1.

Tabel 2- 1 Jenis kernel yang umum digunakan

Jenis Kernel	Definisi
Polynomial	$K\left(\overrightarrow{x_i}, \overrightarrow{x_j}\right) = \left(\overrightarrow{x_i}, \overrightarrow{x_j} + 1\right)^p$
Gaussian RBF	$K\left(\overrightarrow{x_i}, \overrightarrow{x_j}\right) = exp\left(-\frac{\left\|\overrightarrow{x_i} - \overrightarrow{x_j}\right\|^2}{2\sigma^2}\right)$
Linear	$K\left(\overrightarrow{x_i}, \overrightarrow{x_j}\right) = \overrightarrow{x_i}^t \overrightarrow{x_j}$

2.10. Validasi

Pada tahap validasi di penelitian ini digunakan metode *k-fold cross validation*. Dalam *k-fold cross validation*, data awal dibagi menjadi *k* bagian, yaitu D_1, D_2,…D_k dan masing-masing D memiliki jumlah data yang sama. Kemudian dilakukan proses pengulangan sebanyak *k* kali, dimana tiap perulangan ke-*i*, D_i akan dijadikan data *testing*, dan sisanya akan digunakan sebagai data *training* untuk mendapatkan suatu model klasifikasi yang nantinya digunakan dalam proses *testing.*

2.11.Evaluasi

Evaluasi bertujuan untuk mengukur performansi pada sistem yang dibangun. Hasil penelitian akan dievaluasi dengan menggunakan tabel klasifikasi yang bersifat prediktif, *Confusion Matrix* [22]. *Confusion matrix* merupakan tabel yang digunakan untuk menentukan kinerja suatu model klasifikasi [23]. Contoh tabel *confusion matrix* dapat dilihat pada Tabel 2-2.

Tabel 2- 2 Confusion Matrix

		Actual Class	
		+	-
Predicted Class	+	True Positive (TP)	False Positive (FP)
	-	False Negative (FN)	True Negative (TN)

Dari tabel tersebut, terdapat pengkategorian dokumen dalam suatu proses pencarian [24], yaitu:

1. *True Positives* (TP), yaitu hasil dari prediksi sistem yang positif dan sesuai dengan targetnya yang positif.
2. *True Negatives* (TN), yaitu hasil dari prediksi sistem yang negatif dan sesuai dengan targetnya yang negatif.
3. *False Positives* (FP), yaitu hasil dari prediksi sistem yang positif namun hasil targetnya negatif.
4. *False Negatives* (FN), yaitu hasil dari prediksi sistem yang negatif namun hasil targetnya positif.

Precision adalah klasifikasi positif yang benar (*true positive*) dan keseluruhan data yang diprediksikan sebagai kelas positif [25]. *Precision* diperoleh dengan rumus sebagai berikut.

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Recall adalah jumlah dokumen yang memiliki klasifikasi positif yang benar (*true positive*) dari semua dokumen yang benar-benar positif (termasuk di dalamnya *false negative*) [25]. *Recall* diperoleh dengan rumus sebagai berikut.

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

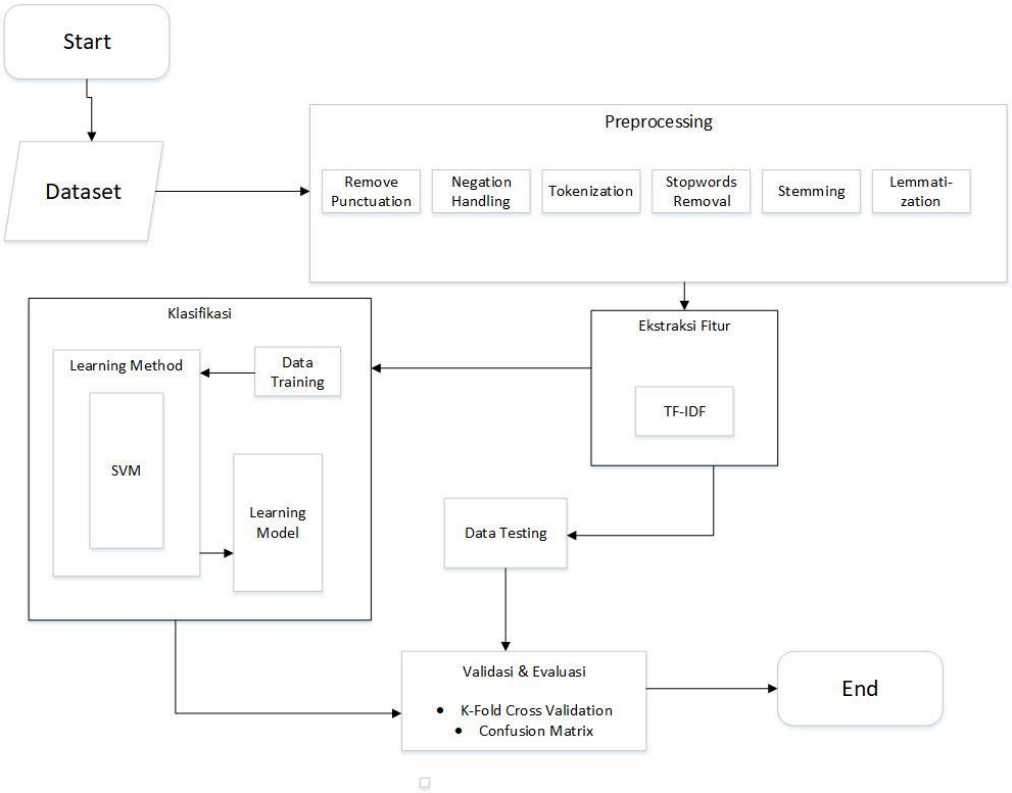
Setelah menghitung parameter dari hasil klasifikasi menggunakan *confusion matrix*, maka dilakukan perhitungan menggunakan *F1-Score.*

F1-Score digunakan untuk mengukur kombinasi nilai yang telah dihasilkan dari *precision* dan *recall*, sehingga menjadi satu nilai pengukuran.

$$F-1\ Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \quad (6)$$

3. Deskripsi Umum Sistem

Dalam penelitian ini, secara umum sistem yang akan dibuat adalah sistem yang mampu mengklasifikasikan sentimen pada *review* film. Gambaran umum yang dilakukan pada sistem ada seperti pada Gambar 3-1 berikut.



Gambar 3- 1 Gambaran umum sistem

4.1. Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah:

1. Menganalisis pengaruh perbandingan data *training* dan data *testing* terhadap peembentukan model klasifikasi.
2. Menganalisis SVM Linear dan Non-Linear untuk kasus *review* film.
3. Menganalisis *negation handling* terhadap metode *Support Vector Machine* pada *review* film.
4. Menganalisis pengaruh tahap *preprocessing* yang digunakan terhadap *Support Vector Machine*.

4.2. Dataset

Dataset yang digunakan pada penelitian ini adalah dataset *review* film yang diambil dari www.cs.cornell.edu/People/pabo/movie-review-data/. *Dataset* ini sudah terbagi menjadi 2 kelompok dokumen yaitu 1000 dokumen yang berlabel positif dan dan 1000 dokumen berlabel negatif. Dataset ini dikumpulkan dari IMDb (*Internet Movie Database)* [26].

Tabel 4- 1 Contoh Dataset

No.	Kalimat Komentar	Label
1.	this film is extraordinarily horrendous and i'm not going to waste any more words on it .	Negatif
2.	claire danes , giovanni ribisi , and omar epps make a likable trio of protagonists , but they're just about the only palatable element of the mod squad , a lame-brained big-screen version of the 70s tv show . the story has all the originality of a block of wood (well , it would if you could decipher it) , the characters are all blank slates , and scott silver's perfunctory action sequences are as cliched as they come . by sheer force of talent , the three actors wring marginal enjoyment from the proceedings whenever they're on screen , but the mod squad is just a second-rate action picture with a first-rate cast .	Negatif
3.	kolya is one of the richest films i've seen in some time . zdenek sverak plays a confirmed old bachelor (who's likely to remain so) , who finds his life as a czech cellist increasingly impacted by the five-year old boy that he's taking care of .	Positif

	though it ends rather abruptly-- and i'm whining , 'cause i wanted to spend more time with these characters-- the acting , writing , and production values are as high as , if not higher than , comparable american dramas . this father-and-son delight-- sverak also wrote the script , while his son , jan , directed-- won a golden globe for best foreign language film and , a couple days after i saw it , walked away an oscar . in czech and russian , with english subtitles .	
4.	this three hour movie opens up with a view of singer/guitar player/musician/composer frank zappa rehearsing with his fellow band members . all the rest displays a compilation of footage , mostly from the concert at the palladium in new york city , halloween 1979 . other footage shows backstage foolishness , and amazing clay animation by bruce bickford . the performance of " titties and beer " played in this movie is very entertaining , with drummer terry bozzio supplying the voice of the devil . frank's guitar solos outdo any van halen or hendrix i've ever heard . bruce bickford's outlandish clay animation is that beyond belief with zooms , morphings , etc . and actually , it doesn't even look like clay , it looks like meat .	Positif

4.3. Skenario Pengujian

a. Pengujian perbandingan komposisi data *training* dan data *testing*
Pengujian perbandingan komposisi data *training* dan data *testing* dilakukan untuk melihat apakah berpengaruh pada model yang dibuat oleh sistem klasifikasi. Pengujian ini dilakukan dengan mengubah komposisi dari data *training* dan data *testing*, kemudian dibandingkan dan dianalisa hasil dari masing-masing komposisi.

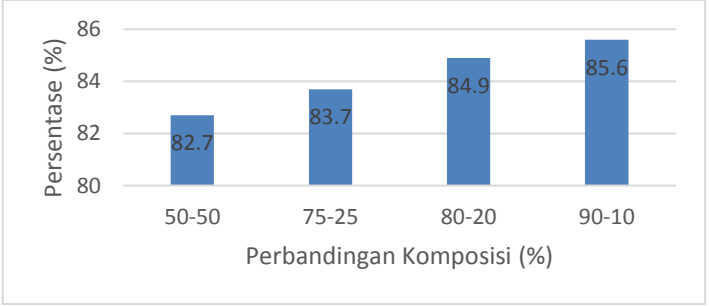
b. Perbandingan fungsi kernel linear dan non-linear pada SVM
Pengujian perbandingan fungsi kernel linear dan non-linear pada SVM dilakukan dengan menggunakan fungsi kernel yang berbeda yaitu fungsi kernel linear, *Radial Basis Function* (RBF), dan *polynomial*. Analisa dilakukan dengan membandingkan hasil akurasi dari tiap fungsi kernel untuk masalah klasifikasi sentimen analisis *review* film.

c. Pengaruh *negation handling* terhadap *Support Vector Machine*
Pengujian negation handling terhadap *Support Vector Machine* dilakukan untuk melihat seberapa pengaruh *negation handling* dalam sistem. Analisa dilakukan dengan membandingkan hasil *F1-Score* dengan *negation handling* dan tanpa *negation handling*.

d. Pengaruh *preprocessing* terhadap *Support Vector Machine*
Pengujian pengaruh *preprocessing* dilakukan untuk melihat apakah proses *stemming* atau *lemmatization* memiliki hasil lebih baik. *Stemming* dan *lemmatization* merupakan proses yang sama yaitu bertujuan untuk mereduksi representasi sebuah kata menjadi bentuk dasar dari kata tersebut. Pengujian ini dilakukan untuk melihat hasil dari proses pemotongan kata yang dilakukan oleh *stemming* dan *lemmatization.*

4.4. Analisis Hasil Pengujian

4.4.1 Analisis Jumlah Dataset



Gambar 4- 1 Perbandingan hasil pengujian komposisi data training dan data testing

Berdasarkan hasil analisis skenario yang telah dilakukan, dapat dilihat bahwa semakin banyak jumlah data yang diberikan pada data *train* maka semakin tinggi akurasi yang didapatkan. Hal tersebut bisa disebabkan karena dengan banyaknya data *train*, model yang terbentuk dapat menangani lebih

banyak keberagaman data yang ada sehingga mampu mengklasifikasikan dengan lebih baik ketika melakukan *testing*. Hasil pengujian dengan komposisi data *training* dan data *testing* sebesar 90%-10% memiliki rata-rata akurasi yang lebih tinggi yaitu 85.6% dengan data *train* berjumlah 1800 data dan data *test* berjumlah 200 data. Pada komposisi 50% data *train* dan 50% data *test* dengan jumlah data *train* dan data *test* masing-masing sebanyak 1000 data memiliki hasil *F1-Score* paling kecil namun rata-rata yang didapatkan tergolong tinggi yaitu 82.7%. Hal ini bisa disebabkan ketika jumlah data *train* yang digunakan lebih sedikit namun karakteristik data *train* tersebut baik (data memiliki perbedaan yang jelas untuk dapat diklasifikasi) maka model yang dibentuk juga akan menghasilkan model yang baik juga sehingga hasilnya yang didapatkan cukup tinggi.

4.4.2 Analisis Fungsi *Kernel* SVM

a. *Kernel* Linear

Tabel 4- 2 Hasil pengujian menggunakan Kernel Linear

No.	Nilai C	Rata-rata *F-1Score* (%)
1.	0.01	32,7%
2.	0.1	38,3%
3.	1	**84,9%**
4.	10	83,6%
5.	100	83,3%
6.	1000	83,7%

Berdasarkan hasil analisis skenario yang telah dilakukan, dapat dilihat perbandingan nilai akurasi pada SVM Linear dengan menggunakan nilai konstanta C sebagai pembandingnya. Sesuai dengan Tabel 4-2, dapat dikatakan bahwa hasil yang terbaik dari SVM Linear adalah menggunakan nilai konstanta C = 1.0, dengan nilai rata-rata *F1-Score* sebesar 84.9%. Pada pengujian ini parameter C mempengaruhi ketepatan klasifikasi data *testing*. Semakin besar nilai parameter C maka semakin menurun nilai akurasi yang dihasilkan. Hal ini bisa disebabkan karena *trade off* antara *margin* dan *error* semakin besar. Nilai pada hasil tersebut merupakan nilai yang didapat dari rata-rata keseluruhan semua fold.

b. *Kernel* RBF

Tabel 4- 3 Hasil akurasi klasifikasi Kernel RBF

C	γ					
	0.01	0.1	1	10	100	1000
0.01	32,4%	32,2%	32,1%	32,8%	32,4%	32,6%
0.1	32,6%	32,5%	32,4%	32,6%	32,8%	32,2%
1	32,5%	79,7%	83,6%	33,5%	33,4%	32,6%
10	79,3%	83,8%	83,9%	32,7%	32,8%	32,7%
100	83,3%	83,8%	83,8%	32,7%	32,7%	32,7%
1000	83,6%	83,5%	**84,3%**	33,4%	32,9%	32,2%

Berdasarkan hasil analisis skenario yang telah dilakukan, dapat dilihat nilai *F1-Score* dari klasifikasi *Kernel* RBF dengan nilai konstanta C dan nilai γ. Nilai konstanta C merupakan yang paling umum untuk semua *kernel* SVM. Fungsi dari parameter gamma adalah menentukan level kedekatan antar dua titik sehingga lebih memudahkan untuk menemukan pemisah *hyperplane* yang konsisten dengan data. Sesuai Tabel 4-3, dapat diketahui bahwa hasil klasifikasi *Kernel* RBF yang terbaik adalah dengan nilai C = 1000 dan γ = 1.0 yakni sebesar 84.3%. Pengujian menggunakan nilai C dan γ dapat mempengaruhi ketepatan klasifikasi data *testing*. Berdasarkan hasil yang diperoleh diatas, dapat dikatakan semakin kecil nilai gamma (γ) maka semakin kecil pula nilai *F1-Score* yang diperoleh sehingga dibutuhkan parameter C yang lebih besar.

c. *Kernel* Polynomial

Tabel 4- 4 Hasil akurasi klasifikasi Kernel Polynomial

d	C					
	0.01	0.1	1	10	100	1000
1	32,5%	**32,8%**	32,7%	32,7%	32,1%	31,8%
2	32,2%	32,3%	32,4%	32,7%	32,7%	32,6%
3	32,2%	32,2%	32,7%	32,1%	32%	31,4%

Berdasarkan hasil analisis skenario yang telah dilakukan, dapat dilihat nilai *F1-Score* dari klasifikasi *polynomial kernel* dengan nilai konstanta C dan d (*degree*). Fungsi dari parameter d (*degree*) yaitu membantu memetakan data dari *input space* ke dimensi *space* yang lebih tinggi pada *feature space*, sehingga dalam dimensi yang baru tersebut bisa ditemukan *hyperplane* yang konsisten. Sesuai Tabel 4-4 dapat diketahui bahwa hasil klasifikasi *polynomial kernel* yang terbaik adalah dengan nilai d=1 dan nilai konstanta C = 0.1 yakni mencapai 32.8%.

4.4.3 Analisis Pengaruh Negation Handling

Tabel 4- 5 Evaluasi pengaruh negation handling

Fold	Tanpa Negation Handling			Dengan Negation Handling		
	Precision	Recall	F-1 Score	Precision	Recall	F-1 Score
1.	79,4%	81,1%	80,3%	86,5%	78,1%	82,1%
2.	85,2%	79%	82,4%	84,8%	89,6%	87,2%
3.	85,4%	83,2%	84,3%	87,5%	85,5%	86,5%
4.	85,1%	87,3%	83,4%	85,4%	86,6%	86%
5.	83,6%	79,8%	81,7%	80,6%	84%	82,2%
Rata-rata			82,4%			84,8%

Berdasarkan Tabel 4-5, ditunjukkan bahwa nilai *F1-Score* terbaik pada percobaan dengan menggunakan *negation handling* dan nilai *F1-Score* yang diperoleh mencapai 84.8%, dimana jika tanpa menggunakan *negation handling* nilai *F1-Score* hanya 82.4%. Perbedaan nilai *F1-Score* yang dihasilkan terhadap metode yang digunakan yaitu *Support Vector Machine* antara dengan *negation handling* dan tanpa *negation handling* yaitu sebesar 0.024. Penanganan negasi bisa dikatakan mampu meningkatkan kinerja analisis sentimen dalam kalimat. Meningkatnya nilai performansi bisa disebabkan karena suatu kata pada konteks kalimat yang berbeda dapat mendeteksi negasi sintetik dimana hal tersebut dapat berpengaruh terhadap hasil prediksi suatu kelas pada *review* film yang digunakan.

4.4.4 Analisis Pengaruh *Preprocessing*

Tabel 4- 6 Perhitungan nilai proses lemmatization

	Lemmatization				
	Fold				
	1	2	3	4	5
F1-Score	85,8%	84,8%	84,1%	83,6%	86,2%
Rata-rata	84,9%				

Tabel 4- 7 Perhitungan nilai proses stemming

	Stemming				
	Fold				
	1	2	3	4	5
F1-Score	84,1%	83,9%	84,8%	83,8%	83,1%
Rata-rata	83,9%				

Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa pada hasil evaluasi, nilai *lemmatization* lebih tinggi dari *stemming*. Pada proses *lemmatization* didapatkan nilai yang lebih baik karena pada proses ini setiap kata dirubah menjadi kata dasar dan membuat peningkatan fitur benar pada setiap dokumen yang mengakibatkan peningkatan pengambilan fitur yang terekstrak. Percobaan ini menyatakan bahwa *lemmatization* memiliki kecenderungan untuk menghasilkan performa yang baik untuk klasifikasi.

Selain itu hal yang menyebabkan proses *stemming* memiliki fitur yang lebih sedikit daripada

lemmatization adalah karena adanya perbedaan proses pemotongan kata. *Stemming* bekerja dengan cara menghilangkan semua imbuhan baik awalan, sisipan, akhiran, maupun gabungan awalan dan akhiran. Menurunnya hasil evaluasi dari proses *stemming* bisa disebabkan karena hilangnya informasi dari kata yang di-stem. Sedangkan *lemmatization* mengubah setiap token ke bentuk dasar umum dari kata tersebut.

4. Kesimpulan dan Saran

Berdasarkan analisis terhadap hasil pengujian yang telah dilakukan pada Tugas Akhir ini, dapat disimpulkan bahwa semakin banyak data yang digunakan pada proses *training*, maka semakin tinggi nilai *F1-Score* yang dihasilkan oleh sistem dalam melakukan klasifikasi. Dalam kasus ini, nilai *F1-Score* yang paling baik pada pembagian data 90% data *training* dan 10% data *testing* dengan hasil 85.6%. Pada pengujian menggunakan linear *separable* dan non-linear *separable* didapatkan nilai *F1-Score* yang baik sebesar 84.9% pada kasus linear *separable*. Fungsi *kernel* yang ada pada SVM dapat digunakan pada kasus *review* film, yaitu *kernel* linear, *kernel* RBF, dan *kernel polynomial*. Dari ketiga *kernel* tersebut dapat diketahui bahwa *kernel* linear memiliki nilai *F1-Score* paling tinggi. Penggunaan *negation handling* memberikan pengaruh dalam analisis sentimen kasus *review* film. Hasil yang paling optimal dihasilkan saat menggunakan *negation handling* dilihat dari perbandingan kalimat yang menggunakan *negation* dan tidak. Proses *stemming* dan *lemmatization* yang diuji menunjukkan bahwa proses *lemmatization* menghasilkan nilai *F1-Score* yang baik daripada *stemming* karena adanya perbedaan dasar pada proses pemotongan kata.

Saran yang diperlukan dari Tugas Akhir ini untuk pengembangan sistem yang lebih lanjut adalah sebagai berikut:

1. Menggunakan metode seleksi fitur karena seleksi fitur mampu memilih fitur yang penting dalam dokumen.
2. Memahami karakteristik *dataset* untuk melakukan eksplorasi yang lebih dalam melakukan klasifikasi sentimen.

Daftar Pustaka

[1] B. Liu, "Sentiment Analysis: A Multi-Faceted Problem," *IEEE Intelligent Systems,* 2010.

[2] K. Yessenov and S. Misailovic, "Sentiment Analysis of Movie Review Comments," 2009.

[3] C. L. Liu, W. Hsaio, C. H. Lee and E. Jou, "Movie Rating and Review Summarization in Mobile Environment," *IEEE Transactions on Systems Man, and Cybernetics, Part C (Applications and Reviews),* 2012.

[4] V. Chandani, R. S. Wahono and Purwanto, "Komparasi Algoritma Klasifikasi Machine Learning Dan Feature Selection pada Analisis Sentimen Review Film," *Journal of Intelligent Systems,* vol. 1, 2015.

[5] R. A. Aziz, M. S. Mubarok and Adiwijaya, "Klasifikasi Topik Pada Lirik Lagu Dengan Metode Multinomial Naive Bayes," *Indonesia Symposium on Computing (IndoSC),* 2016.

[6] A. H. R. Z.A, M. S. Mubarok and Adiwijaya, "Learning Struktur Bayesian Networks Menggunakan Novel Modified Binary Differential Evolution Pada Klasifikasi Data," *Indonesia Symposium on Computing (IndoSC),* 2016.

[7] I. N. Yulita, H. L. The and Adiwijaya, "Fuzzy Hidden Markov Models For Indonesian Speech Classification," *JACIII,* vol. 16(3), pp. 381-387, 2012.

[8] U. N.Wisesty, M. Mubarok and Adiwijaya, "A Classification Of Marked Hijaiyah Letters' Pronunciation Using Hidden Markov Model," *AIP Conference Proceedings 1867,* vol. 020036, 2017.

[9] E. Simoudis, "Reality Check for Data Mining," *IEEE Expert: Intelligent Systems and Their Application,* vol. 11, 1996.

[10] M. Hu and B. Liu, "Mining and Summarizing Customer Review," 2004.

[11] Irawati, "Optimisasi Parameter Support Vector Machine (SVM) Menggunakan Algoritme Genetika," 2010.

[12] B. Liu, "Sentiment Analysis And Opinion Mining," 2012.

[13] R. Feldman and J. Sanger, "The Text Mining HandBook: Advanced Approaches in Analyzing Unstructured Data," 2007.

[14] M. S. Mubarok, Adiwijaya and M. D. Aldhi, "Aspect-based Sentiment Analysis to Review Products Using Naive Bayes," *AIP Conference Proceedings 1867,* vol. 020060, 2017.

[15] C. C. Aggarwal and C. Zhai, "A Survey of Text Clustering," *IBM T.J. Watson Research Center,* pp. 78-128, 2012.

[16] Adiwijaya, Aplikasi Matriks dan Ruang Vektor, Yogyakarta: Graha Ilmu, 2014.

[17] Adiwijaya, Matematika Diskrit dan Aplikasinya, Bandung: Alfabeta, 2016.

[18] J. Han and M. Kamber, "Data Mining: Concepts and Techniques," 2006.

[19] O. Maimon and L. Rokach, " Data Mining and Knowledge Discovery Handbook," 2010.

[20] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, Boston: Pearson Addison Wesley, 2006.

[21] A. S. Nugroho, A. B. Witarto and D. Handoko, Support Vector Machine - Teori dan Aplikasinya dalam Bioinformatika, Penerbit IlmuKomputer.Com, 2003.

[22] D. Xhemali, C. J. Hinde and R. G. Stone, "Naive Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages," *International Journal of Computer Science,* no. 4(1), pp. 16-23, 2009.

[23] L. Hamel, The Encyclopedia of Data Warehousing and Mining, 2nd Edition, Idea Group Publishers, 2008.

[24] S. Bird, E. Klein and E. Loper, Natural Language Processing with Python, 2009.

[25] N. M. S. Hadna, P. I. Santosa and W. W. Winarno, "Studi Literatur Tentang Perbandingan Metode Untuk Proses Analisis Sentimen Di Twitter," *Seminar Nasional Teknologi Informasi dan Komunikasi 2016 (SENTIKA 2016),* 2016.

[26] P. Chaovalit and L. Zhou, "Movie Review Mining: a Comparison between Supervised and Unsupervised Classification Approaches," *Proceedings of the 38th Hawaii International Conference on System Sciences,* 2005.