Klasifikasi *Sentiment Analysis* pada *Review* Film Berbahasa Inggris dengan Menggunakan Metode Doc2Vec dan *Support Vector Machine* (SVM)

Sentiment Analysis Classification of Movie Review in English Language using Doc2Vec and Support Vector Machine (SVM)

Winda Christina Widyaningtyas [1], Adiwijaya [2], Said Al Faraby[3]

[1,2,3]Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom

[1] windachristina99@gmail.com, [2] adiwijaya@telkomuniversity.ac.id, [3] saidalfaraby@telkomuniversity.ac.id

Abstrak

Sentimen merupakan sebuah penilaian dari seseorang berupa pendapat atau komentar terhadap suatu topik atau produk tertentu. Analisis sentiment berfungsi untuk melihat pendapat dan komentar terhadap suatu masalah atau topic tertentu cenderung positif atau negatif. Penelitian Tugas Akhir ini menjelaskan klasifikasi sentiment pada dokumen *review* film untuk mempermudah orang lain dalam mengetahui kualitas sebuah film. Dengan kemajuan di bidang teknologi banyak informasi yang tersedia di internet, salah satunya *review* film. *Review* film berisikan pendapat orang lain mengenai ulasan film. Jika informasi tersebut diolah dengan baik, maka akan diperoleh informasi mengenai kualitas film. Metode yang digunakan dalam penelitian ini yaitu *Doc2Vec* untuk mengekstraksi data menjadi vektor. *Doc2Vec* dipilih karena metode ini membantu komputer untuk mengidentifikasi kombinasi kata yang akan diklasifikasi. Metode klasifikasi yang digunakan dalam penelitian ini yaitu *Support Vector Machine* (SVM) karena SVM mampu mengklasifikasikan data berdimensi tinggi. Proses klasifikasi dilakukan dengan melatih data yang telah ditentukan sehingga akan menghasilkan sebuah model yang akan diujukan pada data testing. Dari uji skenario yang dilakukan, algoritma Doc2Vec dan SVM yang digunakan pada kasus *review* film memiliki nilai *F1-Measure* sebesar 54.1872%.

Kata kunci: *Sentiment Analysis, Doc2Vec, Support Vector Machine (SVM), review film*

1. Pendahuluan

Peranan teknologi sangat besar dalam proses pencarian informasi. Internet merupakan salah satu wadah dalam proses persebaran informasi. Pengguna internet dapat memperoleh informasi teks melalui media internet yaitu melalui website. Informasi teks yang terdapat pada web dapat dibagi menjadi dua kategori yaitu fakta atau opini. Fakta merupakan informasi yang berdasarkan objektifitas sementara opini merupakan ekspresi *sentiment* dari penulisnya [4].

Pengguna internet dapat mengemukakan pendapatnya pada website. Akibatnya banyak opini yang muncul dari pengguna internet mengenai suatu hal yang spesifik. Salah satunya yaitu informasi mengenai film. Pengguna internet dapat mengetahui informasi mengenai film melalui pencarian website yang berisikan *review* film yang diinginkan berdasarkan pendapat penulis website. Namun tidak dapat dipungkiri bahwa pendapat setiap penulis tidak selalu sama antara satu dengan yang lain, dengan banyaknya pendapat muncul pada website maka akan semakin sulit untuk menemukan informasi penting yang sesuai kebutuhan pengguna. Namun jika data *review film* diolah dengan baik akan didapatkan informasi mengenai kualitas film. Proses klasifikasi informasi *review* film akan mempermudah pengguna dalam menarik kesimpulan berdasarkan opini orang lain.

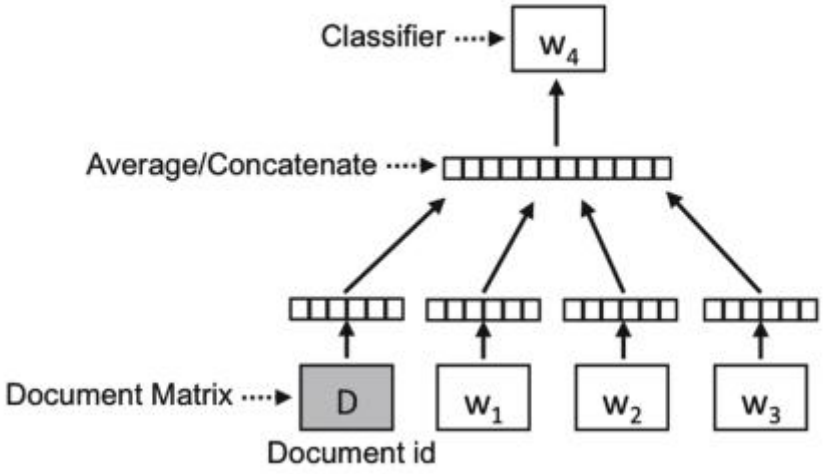
Pada penelitian ini, yang akan dihadapi yaitu penggunaan *Doc2Vec* dalam sistem klasifikasi sentiment analisis. Pada penelitian sebelumnya *Doc2Vec* banyak digunakan untuk melakukan klasifikasi berdasarkan vektor dari data teks [1], sementara pada penelitian ini *Doc2Vec* digunakan untuk melihat vektor dari sebuah paragraph atau dokumen yang kemudian akan diklasifikasi dengan menggunakan metode SVM. *Sentiment analysis* merupakan sebuah bidang ilmu yang menanalisis opini banyak orang, *sentiment*, evaluasi, penelitian, penilaian, sikap dan emosi terhadap sebuah entitas seperti produk, jasa, organisasi, individu, masalah, peristiwa, topik dan atribut lainnya (Bing Liu, 2012). Terdapat tiga level dalam *sentiment analysis* yaitu *coarse-grained sentiment analysis*, *fine-grained sentiment analysis* dan *adjective sentiment analysis* [7]. Pada tahap klasifikasi terdapat beberapa metode yang dapat digunakan, beberapa diantaranya yaitu *Decision Tree* [2], *K-Nearest Neighbour* (KNN)[2], *Support Vector Machine* (SVM) [11], *Random Forest* [2] dan Naïve Bayes [3]. Pada percobaan sebelumnya penggunaan Decision Tree, KNN dan Random Forest memiliki hasil akurasi yang rendah. Pada tugas akhir ini penulis mengambil topik *sentiment analysis* pada *review* film pada level *coarse-grained sentiment analysis* dengan

menggunakan metode word2vec pada proses ekstraksi fitur dan *Support Vector Machine* (SVM) pada proses klasifikasi. Pemilihan metode *Doc2Vec* karena metode ini membantu komputer untuk mengidentifikasi kombinasi kata yang akan diklasifikasi. Pada penelitian sebelumnya[1], penggunaan *Doc2Vec* memiliki hasil akurasi yang baik. Proses klasifikasi dilakukan dengan menggunakan metode *Support Vector Machine* (SVM) karena adanya penelitian sebelumnya dengan menggunakan metode SVM dengan hasil akurasi yang cukup baik yaitu 85.6%[11].

2. Dasar Teori

2.1 Doc2Vec

Doc2Vec merupakan pengembangan dari implementasi metode *word embedding Word2Vec* yang bertujuan untuk merepresentasikan dokumen ke dalam bentuk vector. *Doc2Vec* dapat melakukan ekstraksi fitur dengan menggunakan semua informasi atau kata yang ada pada dokumen, karena setiap kata yang ada pada dokumen digunakan untuk proses *learning*. *Doc2Vec* menghasilkan vektor dokumen dan vektor kata yang ada pada data *training*. Setiap dokumen yang ada pada data *training* akan direpesentasikan ke dalam *word set* dan *tag*. *Word set* adalah semua token yang ada pada masing-masing dokumen, dan *tag* adalah pengidentifikasi pada setiap dokumen.



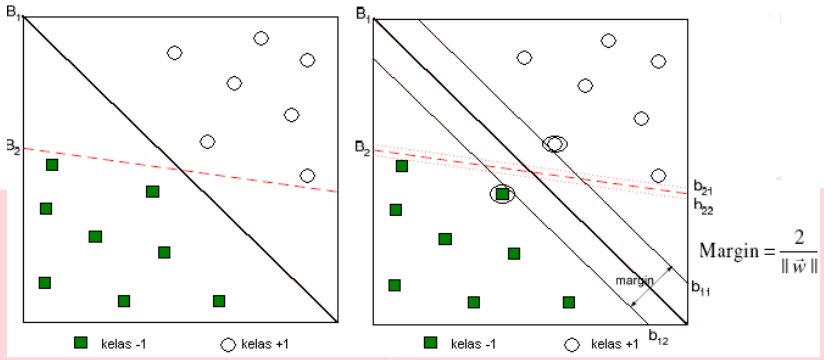
Gambar 1 Struktur *Doc2Vec*

Pada gambar 1 dijelaskan Document Matrix D adalah matriks vektor untuk semua *tag* dalam data *training*. Setiap vektor dokumen direpresentasikan oleh kolom dari matriks D. Word vector matrix W merupakan vektor matriks untuk semua token yang ada dalam data *training*. Setiap vektor kata direpresentasikan dalam matriks W. Vektor dokumen dihasilkan dari konteks dokumen dan keseluruhan yang ada dalam dokumen. Model *Doc2Vec* dilatih dengan data *training* yang kemudian model akan menghasilkan dokumen vektor untuk setiap dokumen. Ukuran matriks D akan sama dengan jumlah dokumen. Hasil vektor yang ada pada data train akan digunakan untuk memprediksi vektor pada data tes. Pada data tes vektor dan bobot dari kata yang ada pada data tes bernilai tetap.

2.2 Support Vector Machine

Support Vector Machine (SVM) merupakan *supervised learning*. SVM pertama kali diperkenalkan oleh Vapnik. SVM memiliki tujuan untuk mencari *hyperplane* dengan margin terbesar antara *hyperplane* dengan *support vector* (jarak terdekat dengan hyperplane)[10].

Pada gambar 2 menunjukkan struktur SVM terdiri dari dua kelas yaitu kelas -1 dan kelas +1. Kedua kelas data tersebut dipisahkan oleh *hyperplane*. Data yang terletak paling dekat dengan *hyperplane* merupakan *support vector* dengan margin sebagai nilai jarak antara *hyperplane* dengan *support vector* [10].



Gambar 2 Pembentukan Hyperplane pada SVM

Jika data dinotasikan sebagai $x \in \Re$ dan label dari masing-masing kelas dinotasikan dengan $y_i \in \{-1, +1\}$ untuk i $= 1,2, \ldots, l$, dimana l adalah banyaknya jumlah data. Dengan asumsi bahwa kedua kelas dapat terpisah secara sempurna oleh *hyperplane* dengan dimensi d maka *hyperplane* dapat dinotasikan sebagai berikut [8]:

$$w.x + b = 0 \quad (1)$$

Karena *hyperplane* membagi dua lokasi berdasarkan kelas masing-masing, maka untuk sampel x_i yang termasuk kelas -1 dan kelas +1 masing-masing memenuhi pertidaksamaan [7]:

$$w.x_1 + b \leq -1 \text{ untuk } Y_i = -1 \quad (2)$$

$$w.x_2 + b \leq +1 \text{ untuk } Y_i = +1 \quad (3)$$

Sehingga kedua margin yang sama tersebut dapat dihitung dengan cara mengurangi persamaa (2) dan (3) didapatlah persamaan berikut [7]:

$$w.(x_1 - x_2) = 2 \quad (4)$$

$$\left[\frac{\mathrm{W}}{\|\mathrm{w}\|}(\mathrm{x}_1 - \mathrm{x}_2)\right] = \frac{2}{\|\mathrm{w}\|} \quad (5)$$

Untuk mencari margin terbesar dapat dilakukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, dan diperoleh formula $\frac{2}{\|\mathrm{w}\|}$.

Hal ini dapat dirumuskan sebagai *quadratic programming problem* (QP), yaitu mencari titik minimal dengan persamaan:

$$\min_w \tau(w) = \frac{1}{2}\|w\|^2 \quad (6)$$

Dengan *constraint* yang harus dipenuhi sesuai dengan pertidaksamaan berikut:

$$y_i(x_i.w + b) - 1 \leq 0, \forall i \quad (7)$$

Untuk menyelesaikan permasalahan (4) dan (5) diatas dapat menggunakan *lagrange multiplier* dengan formula [8]:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=l}^{l} \alpha_i \left(y_i\big((x_i.w + b) - 1\big)\right); (i = 1,2,..l) \quad (8)$$

αi merupakan Lagrange multiplier, yang nilainya lebih besar dari 0, $\alpha i \leq 0$.

Dengan memperhatikan sifat bahwa titik *optimal gradient* L = 0, persamaan (8) dapat dimodifikasi sebagai maksimalisasi problem yang hanya terdiri dari αi, persamaannya sebagai berikut [8]:

Memaksimalkan

$$\sum_{i=l}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=l}^{l} \alpha_i \alpha_j y_i y_j x_i x_j \quad (9)$$

Memenuhi Persamaan

$$\alpha_i \leq 0 \ (i = 1,2, \ldots, l) \qquad \sum_{i=l}^{l} \alpha_i y_i = 0 \quad (10)$$

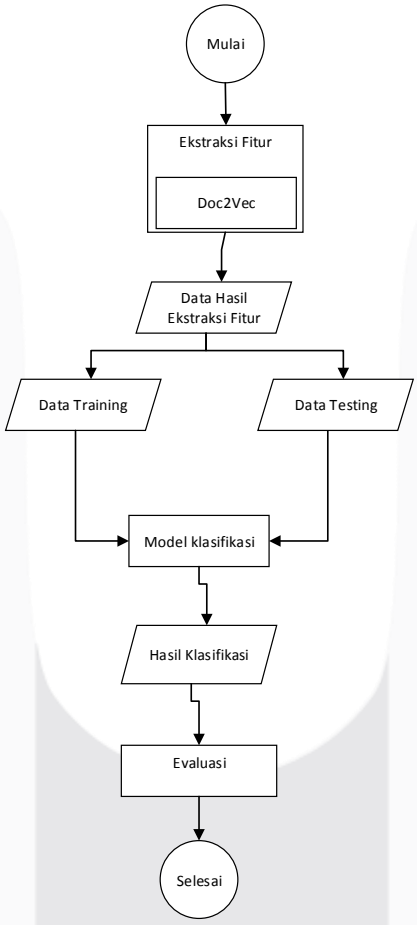
Hasil perhitungan diatas nantinya akan menghasilkan *Lagrange Multiplier* (α_i) yang positif, nantinya data yang berkolerasi dengan α_i yang disebut dengan *support vector* [8].

Tabel 1 *Kernel* SVM

Jenis *kernel*	Definisi
Polynomial	$\mathrm{K}(x_i, x_j) = (x_i, x_j + 1)^p$
Gaussian RBF	$\mathrm{K}(x_i, x_j) = exp\left[-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right]$ $\mathrm{K}(x_i, x_j) = exp\left[-\gamma\|x_i - x_j\|^2, \gamma = \frac{1}{2\sigma^2}\right]$
Sigmoid	$\mathrm{K}(x_i, x_j) = \tan(\alpha x_i, x_j + \beta)$

3. Pembahasan

Dalam sistem ini terdapat empat bagian besar yaitu (1) Dataset, (2) Ekstraksi fitur, (3) Klasifikasi, (4) Evaluasi. Gambaran umum yang dilakuakn pada sistem ada pada Gambar 3.



Gambar 3 Perancangan Sistem

3.1 Dataset

Dataset yang digunakan merupakan data yang telah diklasifikasi menjadi dua kelas yaitu kelas positif dan negatif. Dataset yang digunakan pada penelitian ini adalah dataset *review* film yang diambil dari www.cs.cornell.edu/People/pabo/movie-*review*-data/. Dataset sudah dibagi menjadi 2 kelompok yaitu 1000 dokumen yang berlabel positif dan 1000 dokumen yang berlabel negatif. Dataset ini dikumpulkan dari IMDb (*Internet Movie Database*). Berikut merupakan contoh data set yang akan digunakan :

Contoh kelas positif :

though it is by no means his best work, laissez-passer is a distinguished and distinctive effort by a bona-fide master , fascinating film replete with reward to be had by all willing to make the effort to reap them

Contoh kelas negatif :

a visually flashy but narratively opaque and emotionallu vapid exercise in style and mystification . the story is also as unoriginal as they come , already having been recycled more times than i'd care to count .

3.2 Ekstraksi Fitur

Sebelum melakukan klasifikasi, data akan diekstraksi. Pada tahap ekstraksi data pada tugas akhir ini akan menggunakan metode *Doc2Vec. Training* data dilakukan untuk melihat vektor dari sebuah dokumen. Ekstraksi fitur dilakukan dengan mencari vektor pada setiap kata. Hasil ekstraksi fitur akan digunakan sebagai input pada proses klasifikasi.

Pada tahap pertama pada *Doc2Vec*, data yang ada diubah dari data teks menjadi data numeric untuk melihat bobot dari sebuah kata.

Contoh :

Table 2 Contoh Dokumen

Dokumen 1	The brown cat can not swim
Dokumen 2	My boss ask me can I swim

Pembentukan vocab dengan window = 2 dari dokumen 1

The brown cat can not swim
The brown cat can not swim
The brown cat can not swim
The brown cat can not swim

Komponen dari dokumen 1 dan 2

Table 3 Contoh komponen dokumen

	Ask	Boss	Brown	Can	Cat	I	Me	My	Not	Swim	The
Ask	0	1	0	1	0	0	1	1	0	0	0
Boss	1	0	0	0	0	0	1	1	0	0	0
Brown	0	0	0	1	1	0	0	0	0	0	1
Can	0	0	1	0	1	0	0	0	1	2	0
Cat	0	0	1	1	0	0	0	0	1	0	1
I	0	0	0	1	0	0	1	0	0	1	0
Me	1	1	0	1	0	1	0	0	0	0	0
My	1	1	0	0	0	0	0	0	0	0	0
Not	0	0	0	1	1	0	0	0	0	1	0
Swim	0	0	0	2	0	1	0	0	1	0	0
The	0	0	1	0	1	0	0	0	0	0	0

Semakin banyak jumlah kata yang berkaitan, maka nilai kata utama terhadap kata yang terkait akan semakin tinggi.
Contoh pencarian kata "can" :

Vektor "can" *Weight* dimensi 11 (vektor kata "can")

$$[00010000000] \quad x \quad \begin{bmatrix} 1\,2\,3\,4\,1\,2\,1\,2\,4\,3\,1 \\ 3\,4\,5\,6\,1\,5\,2\,4\,3\,6\,2 \\ 1\,1\,2\,5\,7\,8\,9\,6\,5\,4\,1 \\ 1\,1\,1\,1\,2\,4\,5\,6\,3\,4\,5 \\ 1\,4\,5\,2\,6\,3\,4\,5\,2\,4\,1 \\ \vdots \\ 1\,5\,4\,6\,4\,5\,2\,4\,6\,3\,2 \end{bmatrix} \rightarrow [1\,1\,1\,1\,2\,4\,5\,6\,3\,4\,5]$$

Vektor dokumen merupakan rata-rata dari setiap kata yang ada pada dokumen.
Contoh hasil vektor dokumen 1 :
Dokumen 1 [0.00012, 0.00025, 0.00011, 0.00015, . . , 0.00012]
Dokumen 2 [0.00021, 0.00017, 0.00029, 0.00013, . . , 0.00019]

3.3 Klasifikasi

Proses klasifikasi pada penelitian Tugas Akhir ini menggunakan *Support Vector Machine* (SVM). SVM bertujuan untuk menemukan garis yang terbaik untuk membagi dua kelas dan mengklasifikasikan data test.

Data input SVM adalah hasil vektor pada proses ekstraksi fitur *Doc2Vec*. Data hasil *Doc2Vec* akan di*training* untuk menghasilkan pola yang akan digunakan untuk proses testing SVM, untuk memprediksi kelas dari dokumen.

3.4 Evaluasi

Tahap yang terakhir yaitu tahap evaluasi, evaluasi manggunakan *f1-measure* untuk menguji hasil dari klasifikasi dengan mengukur nilai performansi dari sistem yang telah dibuat. Sebelum menghitung nilai dari *f1-measure* terlebih dahulu mencari nilai dari *recall* dan *precision*. *Recall* dihitung untuk mengevaluasi seberapa besar sample positif sebenarnya dan benar diprediksi positif. *Precision* adalah jumlah sample yang diprediksi positif dan terbukti positif, *precision* dihitung untuk mengevaluasi seberapa baik prediksi dari suatu model klasifikasi.

4. Hasil Pengujian dan Analisis

4.1 Analisis Komposisi Data *Training* dan Data Testing

Pengujian perbandingan komposisi data *training* dan data testing dilakukan untuk melihat pengaruh pada model yang dibuat oleh sistem klasifikasi. Pengujian ini dilakukan dengan mengubah komposisi dari data *training* kemudian dibandingkan dan dianalisa hasil dari masing-masing komposisi. Mengubah komposisi data *training* berguna agar model yang didapatkan memiliki kemampuan dalam hal generalisasi untuk melakukan klasifikasi data.

Tabel *4* Pengaruh Jumlah data Training terhadap Akurasi

Jumlah Data	Akurasi
200	48.98%
500	46.31%
800	51.12%

Dari hasil yang dilakukan pada Tabel 4.1, dapat dilihat bahwa sistem dengan data *training* 800 memiliki akurasi paling tinggi. Hal tersebut bisa disebabkan karena dengan banyaknya variasi data yang ada sehingga sistem mampu mengklasifikasi dengan lebih baik ketika melakukan testing. Hasil pengujian dengan komposisi data *training* 800 memiliki akurasi paling tinggi yaitu 51.12%. Hal ini bisa disebabkan ketika jumlah data train yang digunakan lebih banyak, maka model yang dibentuk juga akan menghasilkan model yang baik juga sehingga hasil yang didapatkan lebih tinggi.

4.2 Analisis perbandingan *size* (dimensi) pada *Doc2Vec*

Pengujian perbandingan dimensi dilakukan untuk melihat pengaruh dimensi terhadap model yang dilakukan oleh sistem. Pengujian ini dilakukan dengan mengubah dimensi dari data *training* kemudian dibandingkan dan dianalisa hasil akurasi dari masing-masing dimensi. Mengubah dimensi berguna agar model memiliki variasi *vocabulary* yang lebih tinggi. Selain itu dengan meningkatkan dimensi berguna untuk menghindari *vocabulary* yang hilang akibat jumlah dimensi terlalu kecil.

Tabel 5 Hasil perbandingan pengaruh dimensi terhadap akurasi

Dimensi	Akurasi
300	45.33%
500	51.83%
1000	51.11%

Berdasarkan penelitian yang telah dilakukan dapat diketahui jumlah dimensi dapat mempengaruhi akurasi. Pada Tabel 4.2 dapat diketahui *Doc2Vec* dengan dimensi 500 memiliki akurasi paling tinggi yaitu 51.83%. Hal ini dapat disebabkan karena data yang diklasifikasi dengan dimensi 500 memiliki jenis *vocabulary* yang sesuai atau vektor yang dekat dengan data *testing*, sehingga *vocabulary* dengan bobot yang rendah akan dieliminasi. Sementara pada dimensi 1000 *vocabulary* dengan bobot yang rendah akan masuk dalam perhitungan, sehingga dapat mempengaruhi akurasi.

4.3 Analisis fungsi *Kernel* 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 *F1-Measure* dari setiap fungsi *kernel* untuk masalah klasifikasi sentiment analisis *review* film.

4.3.1 *Kernel* Linear

Kernel linear adalah *kernel* yang paling sederhana dari semua fungsi *kernel*. Pada percobaan ini, system diuji dengan menggunakan nilai konstanta C yang berbeda yaitu dari rentang $0.01 \leq C \leq 1000$. Kinerja SVM tergantung pada pilihan *kernel* dan penentuan konstanta C. Konstanta C akan berpengaruh pada *trade off* antara *margin* dan *error*. Dengan adanya pemilihan nilai konstanta C yang tepat maka kinerja SVM akan lebih optimal.

Tabel 6 Hasil F1-Score klasifikasi SVM dengan *kernel* linear

C	F1-Measure Data Traing	*F1-Measure Data Testing*
0.01	75.51%	48.69%
0.1	75.51%	48.69%
1	75.51%	48.69%
10	75.51%	48.69%
100	75.51%	48.69%
1000	75.51%	48.69%

Berdasarkan hasil analisis skenario yang telah dilakukan, dapat dilihat bahwa hasil nilai *F1-Measure* pada SVM dengan *kernel* linear menggunakan nilai konstanta C sebagai pembandingnya. Sesuai dengan table 4-3, pada pengujian ini konstanta C tidak mempengaruhi ketepatan klasifikasi data testing.

4.3.2 *Kernel* RBF

Kernel Radial Basic Function (RBF) merupakan *kernel* yang paling banyak digunakan untuk menyelesaikan masalah klasifikasi untuk dataset yang tidak terpisah secara linear karena akurasi pelatihan dan akurasi prediksi yang sangat baik pada *kernel* ini. Pada percobaan ini, system diuji dengan menggunakan nilai konstanta C yang berbeda yakni dari rentang $0.01 \leq C \leq 1000$ dan parameter gamma (γ) dari rentang $0.01 \leq \gamma \leq 1000$.

Tabel 7 Hasil F1-Score klasifikasi SVM dengan *kernel* RBF

γ	C	F1-Measure Data *Training*	*F1-Measure Data Testing*
0,01	0,01	76,66%	51.22%
	0,1	76,66%	51.22%
	1	76,66%	51.22%
	10	76,66%	51.22%
	100	76,66%	51.22%
	1000	76,66%	51.22%
0,1	0,01	75,34%	47.92%
	0,1	75,34%	47.92%
	1	75,34%	47.92%
	10	75,34%	47.92%
	100	75,34%	47.92%
	1000	75,34%	47.92%
1	0,01	75,67%	48.19%
	0,1	75,67%	48.19%
	1	75,67%	48.19%
	10	75,67%	48.19%
	100	75,67%	48.19%
	1000	75,67%	48.19%
10	0,01	75,59%	48.19%
	0,1	75,59%	48.19%
	1	75,59%	48.19%
	10	75,59%	48.19%

	100	81,92%	48.19%
	1000	75,59%	48.98%
100	0,01	75,75%	48.19%
	0,1	75,75%	48.19%
	1	75,75%	48.19%
	10	75,75%	48.19%
	100	82,27%	48.98%
	1000	92,89%	**50.74%**
1000	0,01	77,91%	48.19%
	0,1	77,91%	48.19%
	1	77,91%	48.19%
	10	85,79%	49.49%
	100	99,81%	50.495%
	1000	1,00%	50.495%

Berdasarkan hasil analisis skenario yang telah dilakukan, dapat dilihat nilai *F1-Measure* dari klasifikasi *kernel* RBF dengan nilai konstanta C dan nilai γ. Nilai konstanta C merupakan yang paling umum untuk semua *kernel* SVM.

Fungsi parameter gamma adalah menentukan level kedekatan atara titik sehinggal lebih memudahkan untuk menemukan pemisah hyperplane yang konsisten dengan data. Sesuai Tabel 4.4, dapat diketahui bahwa hasil klasifikasi *kernel* RBF yang terbaik dengan nilai C=100 dan γ = 1000 yakni sebesar 50.74%. Pengujian menggunakan nilai C dan γ dapat mempengaruhi ketepatan klasifikasi data testing.

4.3.3 Kernel Polynomial

Kernel Polynomial adalah *kernel* yang cocok untuk mengatasi masalah klasifikasi, dimana data train sudah normal. Pada percobaan ini, system diuji dengan mengunakan nilai konstanta C yang berbeda yaitu dengan rentang $0.01 \leq C \leq 1000$ dan degree (d) yaitu $1 \leq d \leq 3$.

Tabel 8 Hasil F1-Score klasifikasi SVM dengan *kernel* polynomial

d	**C**	**F1-Measure Data Traing**	***F1-Measure Data Testing***
1	0,01	74.21%	44.71%
	0,1	74.21%	44.71%
	1	74.21%	44.71%
	10	74.21%	44.71%
	100	74.21%	44.71%
	1000	74.21%	44.71%
2	0,01	1	51.23%
	0,1	1	51.23%
	1	1	36.01%
	10	1	40.80%
	100	1	40.80%
	1000	1	40.80%
3	0,01	1	54.19%
	0,1	1	54.19%
	1	1	54.19%
	10	1	54.19%
	100	1	54.19%
	1000	1	54.19%

Berdasarkan hasil analisis skenario yang telah dilakukan, dapat dilihat nilai *F1-Measure* dari klasifikasi *kernel* polynomial dengan nilai konstanta C dan degree (d) yaitu untuk 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 4.5 dapat diketahui bahwa hasil klasifikasi *kernel* polynomial yang terbaik adalah pada d=3 dengan nilai *F1-Measure* 54,1872%.

5 Kesimpulan

1. Komposisi data *training* dapat mempengaruhi akurasi. Hal ini disebabkan semakin tinggi komposisi data *training* maka jumlah variasi data yang di train akan semakin banyak sehingga, sistem dapat melakukan klasifikasi lebih baik. Pada sistem ini nilai F1-Measure terbaik pada komposisi data *training* 800 dan data testing 200.
2. Jumlah dimensi yang ada pada data train dapat mempengaruhi akurasi dari klasifikasi. Hal ini karena semakin tinggi dimensi, maka variasi dari *vocabulary* yang dibentuk oleh model *Doc2Vec* akan semakin tinggi variasinya, sehingga akurasi meningkat. Pada penelitian ini F1-Measure terbaik ada pada dimensi 500 yaitu 51.83%.
3. Fungsi *kernel* pada SVM yang dapat digunakan pada kasus *review* film yaitu *kernel* linear, *kernel* RBF dan *kernel* polynomial. Dengan hasil akurasi tertinggi pada dataset ini yaitu pada *kernel* polynomial dengan akurasi 54.19%

Daftar Pustaka

[1] Acosta, J., Lamaute, N., Luo, M., Finkelstein, E., & Andreea, C. (2017). *Sentiment analysis* of Twitter Messages Using Word2Vec. *CSIS*, 7.

[2] Aziz, R. A., Mubarok, M. S., & Adiwijaya. (2016). Klasfikasi Topik pada Lirik Lagu dengan Metode Multinomial Naive Bayes. *ISSN 2460-3295*, 10.

[3] Dhande, L. L., & Patnaik, G. K. (2014). Analyzing Sentiment of *Movie review* Data using Naive Bayes Neural Classifier . *IJETTCS*, 8.

[4] Naradhipa, A. R., & Purwarianti, A. (2011). Sentiment Classification for Indonesian Message in Social Media. *2011 International Conference on Electrical Engineering and Informatics*, 4.

[5] Nguyen, D. Q., Nguyen, D. Q., Vu, T., & Pham, S. B. (2014). Sentiment Classification on Polarity *Review*s: An Empirical Study Using Rating-based Features. *Association for Computational Linguistics*, 8.

[6] Setiawan, K. Y., Hidayati, H., & Akbar , A. (2014). Analisis User Opinion Twitter pada Level Fine-Grainde *Sentiment analysis* terhadap Tokoh Publik.

[7] Setyawan, D., & Winarko, E. (2016). Analisis Opini Terhadap Fitur Smartphone. *IJCCS, Vol.10, No.2, July 2016, pp. 183~194*, 12.

[8] Vijayarani, S., Ilamathi, J., & Nithya. (n.d.). Preprocessing Techniques for Text Mining - An Overview. *ISSN:2249-5789*, 10.

[9] Yessenov, K., & Misailovic, S. (2009). *Sentiment analysis* of *Movie review* Comments. *6.863 Spring 2009 Final Project*, 17.

[10] Yulietha, I. M., Faraby, S. A., & Adiwijaya. (2017). Klasifikasi Sentiment *Review* Film Menggunakan *Support Vector Machine*. 10.

[11] Le, Q. Mikolov, T. Distributed Representations of Sentences and Documents. Google Inc, 1600 Amphitheatre Parkway, Mountain View, CA 94043

[12] Priansya, S. (2017). Normalisasi Teks Media Sosial Menggunakan Word2Vec, Levenshtein Distance dan Jaro-Winkler Distance.

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