ANALISIS SENTIMEN PADA ULASAN BUKU BERBAHASA INGGRIS MENGGUNAKAN *INFORMATION GAIN* DAN *SUPPORT VECTOR MACHINE*

SENTIMENT ANALYSIS ON THE ENGLISH BOOK REVIEWS USING INFORMATION GAIN AND SUPPORT VECTOR MACHINE

Muhammad Hilman Aprilian Nurjaman[1], Mohamad Syahrul Mubarok[2], Adiwijaya [3]

[1,2,3]Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
[1]aprilianhilman@gmail.com, [2]msyahrulmubarok@gmail.com, [3]kang.adiwijaya@gmail.com

Abstrak

Informasi yang terdapat di Internet sangat bermacam-macam, salah satunya adalah informasi mengenai buku. Jika informasi tersebut diolah dengan baik maka akan diperoleh kualitas buku dari informasi tersebut. Dengan membaca ulasannya, maka kita akan mengetahui kualitas dan juga menganalisis sentimen positif dan juga sentimen negatif dari buku tersebut. Namun, begitu banyaknya opini akan mempersulit pengguna lain untuk memperoleh kualitas dari informasi tersebut. Analisis sentimen merupakan penilaian seseorang tentang topik yang dibahas baik itu sentimen positif ataupun sentimen negatif. Untuk mempercepat dalam menganalisis banyaknya sentimen yang ada, digunakanlah metode klasifikasi yaitu *Support Vector Machine* . Kelebihan dari SVM ini yaitu untuk menentukan *hyperplane* yang dapat menghasilkan margin yang maksimal antara kelas yang satu dengan kelas yang lainnya. Tetapi SVM mempunyai kelemahan terhadap pilihan fitur atau parameter yang dapat mempengaruhi akurasi. Maka dari itu, pada penelitian ini mengunakan metode *Information Gain* agar dapat meningkatkan akurasi dengan mengurangi jumlah fitur yang akan dianalisis dan *Support Vector Machine* sebagai metode klasifikasi untuk menangani permasalahan ini dan hasil dari penelitian ini menghasilkan nilai rata-rata *F1-score* sebesar 82.35%

Kata Kunci: *Review* Buku, Klasifikasi, *Support Vector Machine* (SVM), *Information Gain*

1 Pendahuluan

Buku yang ada saat ini sangat bermacam-macam, baik dari jenisnya, maupun isi cerita dari buku tersebut, tetapi tidak semua buku mempunyai kualitas yang sama. Sebelum memutuskan untuk membeli buku, sebaiknya mengetahui terlebih dahulu informasi buku itu berdasarkan *review* atau opini dari buku yang telah dibeli dan dibaca oleh konsumen tersebut. Dari *review* tersebut dapat membantu konsumen untuk mengetahui kualitas buku itu. Apabila *review* tersebut dibaca secara keseluruhan pasti akan memakan waktu yang lama, tetapi apabila hanya membaca sedikit *review*, informasi yang didapat akan menjadi bias. Untuk mengatasi masalah tersebut, digunakan klasifikasi sentimen yang berfungsi untuk mengelompokkan *review* menjadi opini positif atau negatif secara otomatis [1]. Dengan klasifikasi *review* tersebut, opini dari konsumen lain tentang buku tersebut dapat diketahui secara tepat dan cepat. Dalam klasifikasi sentimen, terdapat masalah *uncertainty reasoning* atau ketidakpastian pada suatu teks. Contohnya jika terdapat dua buah kalimat yang mempunyai fitur atau kata yang sama, maka sistem akan mengalami kesulitan dalam proses klasifikasi [2].

Terdapat beberapa metode klasifikasi yang umumnya digunakan untuk analisis sentimen pada *review* ulasan antara lain *Support Vector Machine* (SVM) [3] [4] [5], *Naïve Bayes* [4], *Character Based N-gram Model* [4], *Artificial Neural Network* [5], *Novel Modified Binary Differential Evolution* (NMBDE) yang merupakan pengembangan dari *Differential Evolution* [6] [7]. Dari teknik-teknik klasifikasi tersebut, SVM merupakan teknik yang paling sering digunakan dalam teknik klasifikasi. SVM adalah metode *supervised learning* yang mengklasifikasikan data secara linier. Kelebihan dari SVM yaitu mampu menentukan *hyperplane* yang dapat menghasilkan margin maksimal antara kelas yang satu dengan kelas yang lainnya [8].

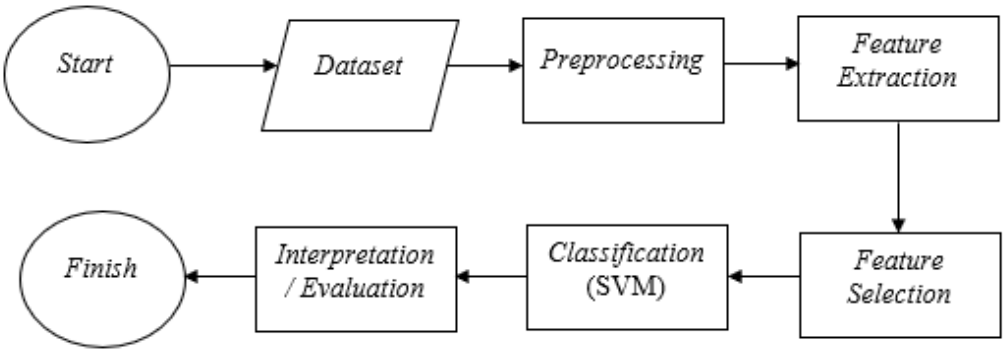
Pemilihan fitur data untuk klasifikasi SVM dapat mempengaruhi performa klasifikasi. Oleh sebab itu, pada penelitian ini *Information Gain* digunakan untuk seleksi fitur yang bertujuan untuk mengurangi fitur yang tidak relevan sehingga dimensi data berkurang dan diperoleh performa klasifikasi yang tinggi.

Analisis sentimen ini akan melihat pendapat atau opini terhadap suatu buku, apakah cenderung beropini positif atau negatif berdasarkan nilai dan titik-titik vektornya. Dengan demikian pengklasifikasian analisis sentiment ini diharapkan akan mempermudah pembaca mengetahui kualitas buku tersebut, sehingga mempermudah pembaca membeli sebuah buku berdasarkan kualitas buku tersebut.

2 Skema yang Diusulkan

2.1 Gambaran Umum Sistem

Sistem yang peneliti bangun merupakan sistem yang mampu mengklasifikasikan sentimen pada ulasan buku berbahasa Inggris secara otomatis menggunakan *Information Gain* dan *Support Vector Machine*. Proses gambaran sistem tersebut digambarkan pada Gambar 1.



Gambar 1 Gambaran umum sistem

2.2 Tahapan Tiap Proses Pembangunan Sistem

2.2.1 Pengumpulan *Dataset*

Dataset adalah sebuah himpunan yang didalamnya berisi data-data [9]. Data yang digunakan adalah data *review* buku dalam bahasa Inggris yang diambil dari situs *Goodreads* [10]. Situs tersebut adalah situs jejaring sosial yang mengkhususkan kategorisasi buku, dimana pengguna dapat *sharing* rekomendasi buku bacaan dengan memberikan opini mereka. Penelitian ini menggunakan 500 opini positif dan 500 opini negatif.

2.2.2 *Case Folding*

Sistem akan mengubah penggunaan huruf kapital menjadi bentuk standar yaitu dirubah menjadi huruf kecil atau *lower case* [11]. Ilustrasi proses ini dapat dilihat pada contoh berikut ini:

Sebelum *Case Folding* : A Sensational and enlightening book
Sesudah *Case Folding* : a sensational and enlightening book

2.2.3 *Remove Punctuation*

Sistem akan menghilangkan tanda baca yang ada di dalam *dataset*. Ilustrasi proses ini dapat dilihat pada contoh berikut ini:

Sebelum *Remove Punctuation* : a sensational and enlightening book…..!!!!
Sesudah *Remove Punctuation* : a sensational and enlightening book

2.2.4 *Tokenization*

Sistem akan mengubah teks yang panjang menjadi kata yang berdiri sendiri-sendiri atau disebut *token* [12]. Ilustrasi proses ini dapat dilihat pada contoh berikut ini:

Sebelum *Tokenization* : a sensational and enlightening book
Sesudah *Tokenization* : a
sensational
and
enlightening
book

2.2.5 *Stop Word Removal*

Sistem akan menjadikan teks menjadi huruf kecil, lalu melakukan proses *stop word removal* atau penghilangan kata yang sering muncul atau disebut *stop word* [13]. Ilustrasi proses ini dapat dilihat pada contoh berikut ini:

Sebelum *Stopword Removal* : a
sensational
and
enlightening
book
Sesudah *Stopword Removal* : sensational
enlightening
book

2.2.6 *Lemmatization*

Dengan teknik ini sistem akan mengubah kata berimbuhan menjadi bentuk kata dasarnya [14]. Ilustrasi proses ini dapat dilihat pada contoh berikut ini:

Sebelum *Lemmatization* : perform evaluation **using** goodreads dataset
Sesudah *Lemmatization* : perform evaluation **use** goodreads dataset

2.2.7 *Feature Extraction*

Pada tahap ini, sistem akan menghitung jumlah kemunculan *term*, *inverse document frecuency* (idf) dan juga *term weighting* pada dokumen *dataset* menggunakan proses TF-IDF. Contoh Representasi penghitungan *term weighting* dari data yang ditampilkan ditujukan pada Tabel 2-2.

Tabel 2-2 Tabel Representasi Perhitungan Term Weighting

	Tf								idf	Wdt = tf.idf						
Term	D1	D2	D3	D4	D5	D6	D7	df	log(n/df)	D1	D2	D3	D4	D5	D6	D7
Junk	1							1	0.85	0.85	0	0	0	0	0	0
Awful			1					1	0.85	0	0	0.85	0	0	0	0
book					1	1	1	3	0.37	0	0	0	0	0.37	0.37	0.37
long		1						1	0.85	0	0.85	0	0	0	0	0
waste		1						1	0.85	0	0.85	0	0	0	0	0
time				1				1	0.85	0	0	0	0.85	0	0	0
worthy					1			1	0.85	0	0	0	0	0.85	0	0
clancy	1							1	0.85	0.85	0	0	0	0	0	0
great			2				1	3	0.37	0	0	0.74	0	0	0	0.37
Funny					1			1	0.85	0	0	0	0	0.85	0	0
person		1						1	0.85	0	0.85	0	0	0	0	0

2.2.8 *Feature Selection*

Penerapan metode *feature selection* digunakan untuk mengurangi dimensi dari set fitur dengan menghapus fitur yang tidak relevan [15]. Fitur seleksi mempunyai keunggulan seperti ukuran *dataset* yang lebih kecil, menyusutkan ruang pencarian, dan kebutuhan komputasi yang rendah. Tujuannya yaitu mengurangi ukuran dimensi untuk menghasilkan meningkatnya akurasi klasifikasi. Metode fitur seleksi ini digunakan untuk klasifikasi dokumen teks menggunakan fungsi evaluasi yang dipakai untuk satu kata. Fitur individu terbaik dapat dilakukan menggunakan beberapa metode, salah satunya *Information Gain* (IG).

2.2.9 Klasifikasi Menggunakan *Support Vector Machine*

Proses klasifikasi pada penelitian ini menggunakan *Support Vector Machine.* Dalam proses klasifikasi ini, untuk pengubahan data latih ke dimensi yang baru perlu menggunakan metode nonlinier. Pada dimensi yang lebih tinggi ini, SVM ini akan mencari *hyperplane* sebagai pemisah tupel dari dua kelas yang berbeda [3]. Dengan metode pemetaan nonlinier yang sesuai, *hyperplane* dapat memisahkan data dari dua kelas yang berbeda tersebut. *Support vector* (tupel *training*) dan *margins* ini digunakan untuk menemukan *hyperplane* pada metode klasifikasi SVM.

2.2.10 Evaluasi dan Validasi

Setelah semua proses dilakukan, yaitu dari proses *pre-processing* hingga proses klasifikasi menggunakan *Support Vector Machine*, selanjutnya dilakukan evaluasi dan validasi dari hasil kinerja klasifikasi. Pada penelitian ini, evaluasi untuk mengetahui performa dari hasil kinerja klasifikasi menggunakan *confusion matrix* dan validasi menggunakan *5-fold cross validation*. Pada saat melakukan validasi, urutan dari kumpulan dokumen yang ada akan diacak. Hal ini bertujuan untuk menghindari adanya pengelompokkan dokumen yang berasal dari kategori tertentu.

3 Analisis Hasil Pengujian

3.1 Analisis Pengaruh *Pre-Processing* terhadap Proses Klasifikasi

Pengujian ini dilakukan untuk melihat pengaruh dari penggunaan proses *pre-processing* terhadap klasifikasi. Hasil dari pengujian akan dievaluasi menggunakan *F1-score* yang kemudian hasil tersebut dianalisis. Jadi, semakin tinggi nilai rata-rata dari *F1-score* maka sistem tersebut semakin baik. Hasil dari pengujian ini ditujukan pada Tabel 3-1.

Tabel 3-1 Hasil Pengujian Pengaruh Lemmatization

Fold	Pengujian	
	Tanpa Lemmatization	Lemmatization
1	81.41	79.59
2	82.35	81.65
3	85.71	82.99
4	86.67	83.69
5	87.50	83.85
Rata-Rata	**84.73**	82.35

Berdasarkan Tabel 3-1, hasil uji *pre-processing* yang menggunakan *lemmatization* memiliki nilai yang lebih rendah dibanding hasil uji tanpa *lemmatization* dengan memiliki nilai rata-rata *F1-score* 82,35. Dari hasil pengujian ini dapat diketahui bahwa pengujian yang menggunakan *lemmatization* menghasilkan klasifikasi yang rendah. Hal tersebut disebabkan *lemmatization* ini mengubah setiap kata menjadi kata dasar sehingga pada *dataset* yang ada antara kata-kata di *class* positif dan *class* negatif memiliki sedikit kemiripan dan akhirnya pada saat proses klasifikasi tersebut menghasilkan hasil yang rendah.

3.2 Analisis Pengaruh Fitur Seleksi *Information Gain*

Pengujian kedua ini, *Information Gain* akan memilih fitur-fitur yang memenuhi *threshold*. Parameter *threshold* yang digunakan akan mempengaruhi kinerja dari *Support Vector Machine*. Hasil evaluasi dari pengujian parameter threshold ditujukan pada Tabel 3-2.

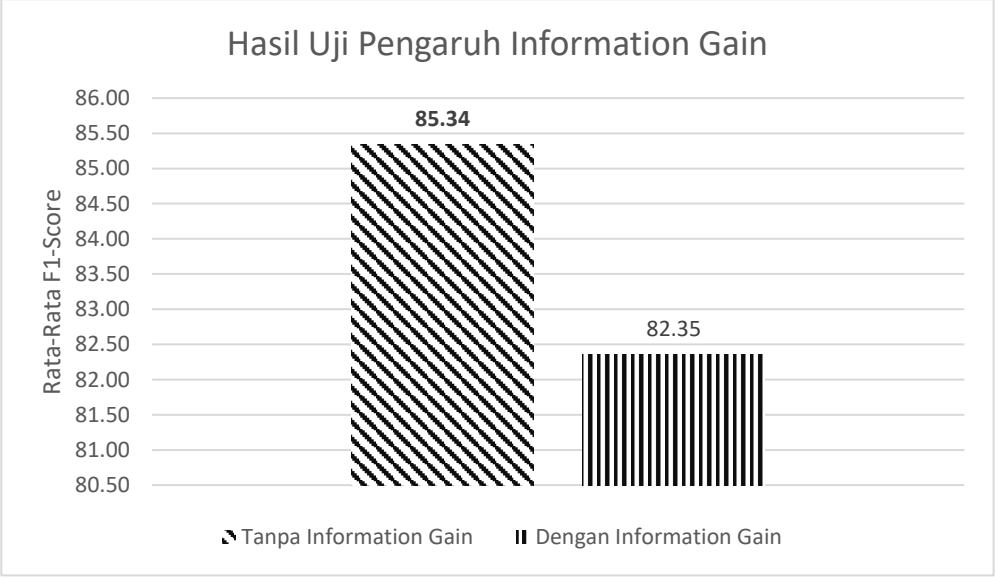
Tabel 3-2 Hasil Uji Pengaruh Treshold

Treshold	Rata-Rata F1-Score
0.5	**82.35**
0.7	**82.35**
0.9	**82.35**
0.96	82.04
0.97	81.47
0.98	80.76

Berdasarkan Tabel 3-2, hasil terbaik dari pengujian threshold yaitu threshold 0.5, 0.7, 0.9 dengan nilai sebesar 82.35. Kemudian threshold tersebut digunakan untuk pengujian pengaruh *Information Gain* terhadap klasifikasi *Support Vector Machine*. Selanjutnya pengujian yang dilakukan yaitu dengan membandingkan hasil rata-rata *F1-score* menggunakan *Information Gain* dan tanpa *Information Gain*. Hasil pengujian pengaruh *Information Gain* ini ditujukan pada tabel 3-3.

Tabel 3-3 Hasil Uji Pengaruh Information Gain

Fold	F1-Score	
	Tanpa Information Gain	Dengan Information Gain
1	82.00	79.59
2	82.72	81.65
3	86.01	82.99
4	87.38	83.69
5	88.57	83.85
Rata-Rata	**85.34**	82.35



Gambar 3-3 Pengaruh Information Gain Terhadap F1-Score

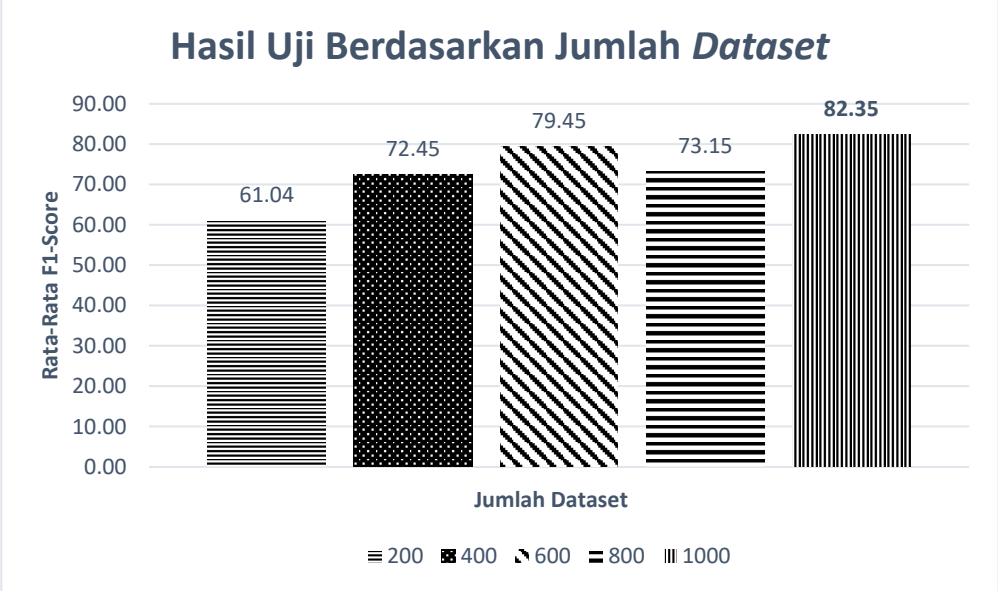
Berdasarkan hasil uji pengaruh *Information Gain* pada Tabel 3-3 dan Gambar 3-3, Performa setelah menggunakan *Information Gain* dengan *threshold* 0.5, 0.7, 0.9 adalah 82.35. Performa *Support Vector Machine* mengalami penurunan dengan menggunakan *Information Gain* sebagai fitur seleksi. Hal ini disebabkan adanya kemiripan kata yang ada di *class* positif dan *class* negatif sehingga pada saat proses seleksi fitur banyak kata yang dianggap tidak penting pada setiap *class* tersebut dibuang. Faktor lainnya yaitu nilai gain pada setiap kata memiliki nilai *gain* dengan rata-rata 0.985 sehingga nilai *F1-score* pada hasil uji 0.5, 0.7, 0.9 nilainya sama dan performa klasifikasi tidak efektif. Pada penelitian ini, jika ingin mengambil *threshold* < 0.5 pun sebenarnya tidak berpengaruh pasti hasil *F1-Score* tersebut sama dengan *threshold* 0.5, 0.7, dan 0.9 sehingga pada pengujian ini hanya mengambil nilai *F1-Score* tertinggi pada *threshold* yang ada pada Tabel 4-2. Terkecuali nilai gain pada penelitian ini bermacam-macam, yaitu berada diantara *threshold* 0.1 hingga 1, pasti hanya satu *threshold* yang memiliki nilai *F1-Score* tertinggi. Faktor terakhir yaitu dengan dilakukannya proses *Information Gain*, adanya satu data yang hilang semua kata sehingga mengurangi keseluruhan dari data sebelumnya dan konsep dari TF-IDF ini harus mempertahankan suatu data tidak boleh mengurangi data yang sebelumnya. Oleh karena itulah pada penelitian ini, hasil performa klasifikasi *Support Vector Machine* dengan menggunakan *Information Gain* menurun.

3.3 Analisis Pengaruh Jumlah Dataset terhadap Model Klasifikasi

Pada pengujian ketiga ini dilakukan dengan membandingkan *F1-score* dari jumlah *dataset* yang ditentukan. Hasil pengujian pengaruh jumlah *dataset* terhadap model klasifikasi ditujukan pada Tabel 3-4.

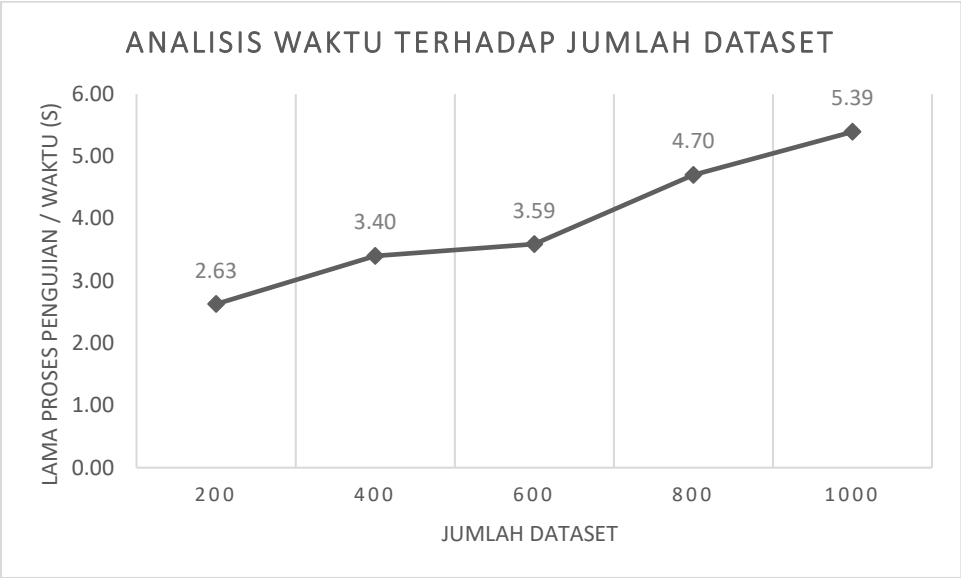
Tabel 3-4 Hasil Uji Pengaruh Jumlah Dataset

Jumlah Dataset	F1-Score					Rata-Rata
	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	
200	55.00	59.74	60.87	64.94	64.65	61.04
400	64.65	71.60	75.64	74.89	75.48	72.45
600	75.84	80.65	79.34	79.89	81.51	79.45
800	81.21	66.67	67.09	73.27	77.50	73.15
1000	79.59	81.65	82.99	83.69	83.85	**82.35**



Gambar 3-4 Pengaruh Jumlah Dataset Terhadap F1-Score

Berdasarkan Tabel 3-4 dan Gambar 3-4, dapat dilihat bahwa hasil pengujian ini adalah *dataset* yang berjumlah 1000 data dengan nilai rata-rata *F1-score* tertinggi sebesar 82,35. Hasil uji tersebut menunjukkan bahwa semakin banyak *dataset* yang digunakan, maka nilai rata-rata *F1-score* yang didapat juga semakin tinggi. Ini menyatakan bahwa banyak data yang dipelajari oleh sistem dan sistem yang dibangun semakin baik.



Gambar 3-5 Analisis Waktu Terhadap Jumlah Dataset

Dari hasil uji yang dilakukan berdasarkan Gambar 3-5, dapat dilihat bahwa dengan *dataset* sebanyak 200 memerlukan waktu komputasi 2.63 sekon, 400 *dataset* memerlukan waktu komputasi selama 3.40 sekon, 600 *dataset* memerlukan waktu komputasi selama 3.59 sekon, 800 dataset memerlukan waktu komputasi selama 4.70 sekon, dan 1000 dataset memerlukan waktu komputasi selama 5.39 sekon. Dengan demikian dapat disimpulkan bahwa semakin banyak *dataset* maka waktu komputasi yang dibutuhkan oleh sistem lebih lama.

4 Kesimpulan

Kesimpulan yang dihasilkan dari analisis dan pengujian yang telah dilakukan antara lain:

1. Sistem yang dibangun yang dibangun menggunakan *Information Gain* dan *Support Vector Machine* menghasilkan rata-rata *F1-Score* sebesar 82,35%
2. Secara umum penghapusan tanda baca pada fitur ekstraksi mampu meningkatkan hasil klasifikasi, begitu pula penghapusan *stop word.*
3. Untuk menghasilkan performa sistem yang baik, kombinasi teknik saat *preprocessing* perlu dilakukan.
4. Secara umum pada pengujian yang dilakukan dengan menggunakan *Information Gain* dan *lemmatization* ini mengalami penurunan dikarenakan faktor *dataset* yang kurang optimal serta faktor kemiripan kata saat diproses dengan menggunakan *Information Gain* dan *lemmatization* sehingga hasil dari pengujian tersebut menurun.

Daftar Pustaka

[1] Z. Zhang, Q. Ye, Z. Zhang and Y. & Li, "Sentiment Classification of Internet Restaurant Reviews Written in Cantonese," in *Expert Systems with Applications, 38(6)*, 2011, pp. 7674-7682. doi:10.1016/j.eswa.2010.12.147.

[2] M. S. Mubarok, Adiwijaya and M. D. Aldhi, "Aspect-based Sentiment Analysis to Review Products Using Naive Bayes," *Am. Inst. Phys.*

[3] A. S. H. Basari, B. Hussin, I. G. P. Ananta and J. & Zeniarja, "Opinion Mining of Movie Review using Hybrid Method of Support and Particle Swarm Optimization," in *Procedia Engineering, 53*, (2013), pp. 453-462. doi:10.1016/j.proeng.2013.02.059.

[4] Q. Ye, Z. Zhang and R. Law, "Sentiment Classification of Online Reviews to Travel Destinations By Supervised Machine Learning Approaches," in *Expert Systems with Applications, 36(3)*, (2009), pp. 6527-6535. doi:10.1016/j.eswa.2008.07.035.

[5] R. Moraes, J. F. Valiati and W. P. & Gaviao Neto, "Document-Level Sentiment Classification: An Empirical Comparison Between SVM and ANN," in *Expert Systems with Applications, 40(2)*, (2013), pp. 621-633. doi:10.1016/j.eswa.2012.07.059.

[6] A. Arifin, M. Mubarok and Adiwijaya, "Learning Struktur Bayesian Networks menggunakan Novel Modified Binary Differential Evolution pada Klasifikasi Data," in *Indonesia Symposium on Computing (IndoSC)*, 2016.

[7] L. Wang, X. Fu, Y. Mao, M. I. Menhas and M. Fei, "A novel modified binary differential evolution algorithm and its applications," *Bio-inspired computing and applications (LSMS-ICSEE ' 2010)*, vol. 98, pp. 55-75, 2012.

[8] J. S. Chou, M. Y. Cheng, Y. W. Wu and A. D. & Pham, "Optimizing Parameters of Support Vector Machine Using Fast Messy Genetic Algorithm For Dispute Classification," in *Expert Systems with Applications, 41(8)*, (2014), pp. 3955-3964. doi:10.1016/j.eswa.2013.12.035.

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

[10] "Goodreads," [Online]. Available: https://www.goodreads.com/review/recent_reviews.

[11] N. Eko, Perancangan Sistem Deteksi Plagiarime Dokumen Teks Dengan Menggunakan Algoritma Robin-Karp., Program Studi Ilmu Komputer, Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Brawijaya Malang., 2011.

[12] G. Grefenstette, Syntactic Wordclass Tagging, Springer Netherlands, 1999.

[13] A. Rajaraman, J. Leskovec dan J. D. Ullman, Mining of Massive Datasets, 2014.

[14] B. Jongejan dan H. Dalianis, “Automatic training of lemmatization rules that handle morphological changes in pre-, in- and suffixes alike,” dalam *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, Suntec, 2009.

[15] X. Zhang, H. Huang and K. Zhang, "KNN Text Categorization Algorithm Based on Semantic Centre," in *International Conference on Information Technology and Computer Science*, 2009.