

# PENGARUH STEMMER BAHASA INDONESIA TERHADAP PEFORMA ANALISIS SENTIMEN TERJEMAHAN ULASAN FILM

I Made Artha Agastya<sup>1)</sup>

<sup>1)</sup>Informatika, Universitas AMIKOM Yogyakarta

Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta, DIY

Email : artha.agastya@amikom.ac.id<sup>1)</sup>

## Abstrak

Bahasa Indonesia memiliki banyak variasi akhiran, awalan, dan sisipan. *Stemming* adalah bagian dari prapengolahan dari analisis sentimen yang mendeteksi dan menghilangkan imbuhan tersebut. Pengaruh dari *stemming* pada analisis sentimen masih belum jelas karena dataset yang digunakan tidak terdistribusi secara bebas. Untuk mendapatkan pengaruh dari *stemming* terhadap analisis sentimen maka dilakukan percobaan dengan dataset ulasan film yang sudah diterjemahkan ke Bahasa Indonesia. Stemmer Sastrawi sebagai algoritma *stemming* terbaru digunakan pada penelitian ini. Dataset dibagi menjadi 5 (lima) kategori yang mana 100 data, 250 data, 500 data, 750 data, dan 1000 data. Hasil yang diperoleh menunjukkan bahwa stemmer tidak memberikan peningkatan akurasi yang stabil. Bahkan waktu yang diperlukan untuk menyelesaikan analisis sentimen memerlukan waktu meningkat hingga 310 kali lipat. Kenyataan ini sangat buruk karena *stemming* dapat mengurangi efisiensi dari analisis sentimen.

**Kata kunci:** Analisis Sentimen, Stemming, Sastrawi, Ulasan Film, Bahasa Indonesia.

## 1. Pendahuluan

*Text Mining* (Liu, 2012) telah menjadi riset topik yang luas dan populer. Pendekatan Text Mining terbagi menjadi dua bagian yaitu dengan pendekatan lexical dan pendekatan pembelajaran mesin. Pada pendekatan lexical yang menjadi acuan adalah kamus dan corpus. Sedangkan pada pendekatan pembelajaran mesin (Haykin, 2009) terbagi empat jenis, yaitu pembelajaran terbimbing (*supervised learning*), pembelajaran tidak terbimbing (*unsupervised learning*), pembelajaran semi terbimbing (*semi supervised learning*), pembelajaran dengan penguatan (*reinforcement learning*). Namun dari empat tersebut yang paling sering digunakan adalah *supervised* dan *unsupervised learning*. Machine learning lebih favorit digunakan karena tidak diperlukan pemahaman mendalam mengenai struktur bahasa dari kata atau kalimat. Pada pendekatan *unsupervised learning* dapat dilakukan metode clustering (Agastya, Adji and Setiawan, 2017)(Chifu, Letia and Chifu, 2015)(Suresh and S., 2016) untuk mengelompokkan dokumen secara efektif seperti dengan K-Means dan Fuzzy C Means (FCM). Sedangkan pendekatan *supervised learning* (Kaur, Mangat and Nidhi, 2017) menggunakan berbagai macam metode yang terbagi menjadi empat yaitu *linear*, *decision tree*, *ruled based*, dan *probabilistic*.

Analisis sentimen merupakan bagian dari *text mining* yang merupakan metode untuk mengambil opini dari suatu kalimat atau dokumen. Saat ini analisis sentimen dengan menggunakan Bahasa Indonesia menjadi sangat populer di kalangan peneliti maupun bagi kalangan praktisi Industri. Terlihat dari banyaknya paper – paper yang dihasilkan dan perusahaan yang melakukan analisis sentimen untuk komersial. Beberapa contoh analisis sentimen yang sudah dilakukan seperti prediksi hasil pemilihan presiden atau gubernur (Aliandu, 2013)(Buntoro, 2017), sentimen masyarakat terhadap pemimpin publik atau pemerintahan (Faradhillah, Kusumawardani and Hafidz, 2016)(Aditya, 2016), dan sentimen masyarakat terhadap perubahan harga bahan pokok (UN Global Pulse, 2014)(Surjandari, Naffisah and Prawiradinata, 2015).

Bagian yang sangat penting pada analisis sentimen adalah prapengolahan. Prapengolahan (Liu and Zhang, 2012) merupakan proses untuk memperbaiki inputan data yang tidak teratur. Hal ini untuk menanggulangi salah mengambil ciri atau atribut. Karena kesalahan dalam menggunakan ciri atau atribut dapat menurunkan performa analisis secara signifikan. Beberapa diantaranya yaitu tokenisasi, *case folding*, penghapusan karakter spesial, penghapusan *stop word*, dan *stemming*. *Stemming* merupakan proses untuk mengembalikan kata kerja yang berimbuhan menjadi kata dasarnya. Menurut penelitian (Adriani *et al.*, 2007), *stemming* dapat meningkatkan *recall* namun dapat menurunkan *precision* karena berkurangnya informasi dari kata yang telah di-stem.

Saat ini masih diperdebatkan apakah *stemming* dapat membantu meningkat performa dari analisis sentimen. Saat ini penelitian analisis sentimen berbahasa Indonesia menjadi populer. Namun dataset yang dapat digunakan untuk melakukan analisis sentimen masih belum ada. Sehingga sulit untuk menyatakan bahwa hasil yang diperoleh oleh peneliti yang menggunakan data pribadi itu benar. Oleh karena dilakukan penelitian untuk mencari pengaruh dari *stemming* pada peforma analisis sentimen berbahasa Indonesia dengan dataset ulasan film yang sudah diterjemahkan ke Bahasa Indonesia. Kontribusi yang dilakukan pada penelitian ini adalah:

1. Menunjukkan bahwa pengaruh *stemming* tidak konsisten terhadap akurasi karena stemming dapat meningkatkan atau menurunkan akurasi bergantung dengan jumlah data yang digunakan.
2. Membandingkan waktu komputasi yang diperlukan pada sentimen analisis tanpa *stemming* dan sentimen analisis dengan *stemming*.

3. Memberikan akses pada peneliti lain untuk menggunakan dataset analisis yang tersedia secara publik.

## 2. Pembahasan

### 2.1 Tinjauan Pustaka

*Stemming* yang dibuat oleh Nazief dan Adriani (NA) (Adriani *et al.*, 2007) lebih tepat untuk disebut dengan *lemmatization* karena melakukan proses yang dilakukan adalah mengembalikan kata yang berimbuhan menjadi kata dasar. Seperti pada kata berikut ini:

1. belajar → ajar
2. membaca → baca

Stemmer NA sudah diuji presisinya (Agusta, 2009) dengan stemmer porter Bahasa dan diperoleh kesimpulan bahwa presisi stemmer NA lebih tinggi dari stemmer porter sekitar 5 – 10 %. Namun jika dilihat dari rata-rata waktu komputasi, stemmer NA bekerja lima puluh kali lebih lama dari stemmer porter. Pada Penelitian (Wira and Indra, 2010), Wira dan Indra menguji pengaruh *stemming* dan *stop word* bahasa pada algoritma *Naïve Bayes* (NB). *Stemming* atau *stopword* tidak memberikan pengaruh yang besar karena *stemming* hanya memberikan peningkatan akurasi sebesar 0.12 %. Sedangkan penghapusan *stop word* memberikan peningkatan 0.38 %. Namun gabungan keduanya meningkatkan akurasi sebesar 0.92%. Penelitian terhadap pengaruh *stemming* terhadap akurasi analisis sentimen (Hidayatullah, Ratnasari and Wisnugroho, 2016) sudah dilakukan pada dua buah metode, yaitu *Naïve Bayes* (NB) dan *Support Vector Machine* (SVM). Dari penelitian tersebut diperoleh kesimpulan bahwa akurasi NB dan SVM menurun ketika digunakan *stemming*. Dataset yang digunakan adalah data twitter yang sudah diolah hingga diperoleh 2000 data. Namun data tersebut tidak tersedia secara publik sehingga untuk melanjutkan penelitian tersebut menjadi sulit dilakukan. Analisis pengaruh *stemming* dan *stopword* terhadap deteksi kemiripan dari abstrak juga sudah dilakukan. Hasil yang diperoleh adalah *stemming* tidak memberikan dampak signifikan karena hanya sekitar 1% untuk presisinya. Namun jika digabungkan dengan penghapusan *stopword* maka peningkatan presisinya menjadi sekitar 10%. Namun penelitian ini tidak menunjukkan pengaruh yang dihasilkan dari penghapusan *stopword*. Dataset abstrak yang digunakan juga tidak tersedia secara publik sehingga peneliti yang tertarik pada penelitian tersebut tidak dapat menggunakan data yang sama untuk mencari pengetahuan baru atau mengembangkan metode baru. Semua penelitian yang sudah disebutkan masih menggunakan algoritma NA yang mana memiliki kelemahan pada jumlah term atau kata yang dihasilkan cukup banyak dan menyebabkan waktu komputasinya tinggi. Oleh karena itu pada penelitian ini (Arifin, Mahendra and Ciptaningtyas, 2009) dibuat metode *Enhanced Confix Stripping Stemmer* (ECS) yang dapat mengurangi ukuran term sebesar 32.66% Kemudian stemmer tersebut diperbaiki

lagi untuk menyelesaikan masalah *over stemming* dan *under stemming* pada penelitian (Tahitoe and Purwitasari, 2010). Namun pada penelitian tersebut tidak dibandingkan berdasarkan performa pengenalan kata dasar namun dibandingkan berdasarkan implementasi pencarian dokumen. Tidak diketahui keunggulan dari perbaikan metode *stemming* yang diajukan. Kemudian ketiga penelitian tersebut dibuat sebuah program stemmer yang berbasis PHP dengan nama Sastrawi (Librian, 2017). Stemmer Sastrawi sudah mengalami perbaikan dan merupakan stemmer terbaru saat ini. Oleh karena itu, stemmer Sastrawi menjadi stemmer yang diuji pengaruhnya terhadap performa analisis sentimen.

### 2.2 Alat dan Bahan

Pada penelitian ini menggunakan perangkat keras dan lunak sebagai berikut:

Perangkat keras:

1. CPU 2.5 GHz Intel Core i5-3210M
2. Memori DDR3 8 GB
3. Sistem Operasi GNU/Linux Ubuntu versi 16.04 LTS atau Long Term Support
4. Penyimpanan berupa Hard Disk dengan kapasitas 150 GB

Perangkat lunak:

1. Anaconda versi 5.1
2. Spyder versi 3.4.2
3. Python versi 2.7.14
4. Pustaka NLTK, Pandas, dan Numpy

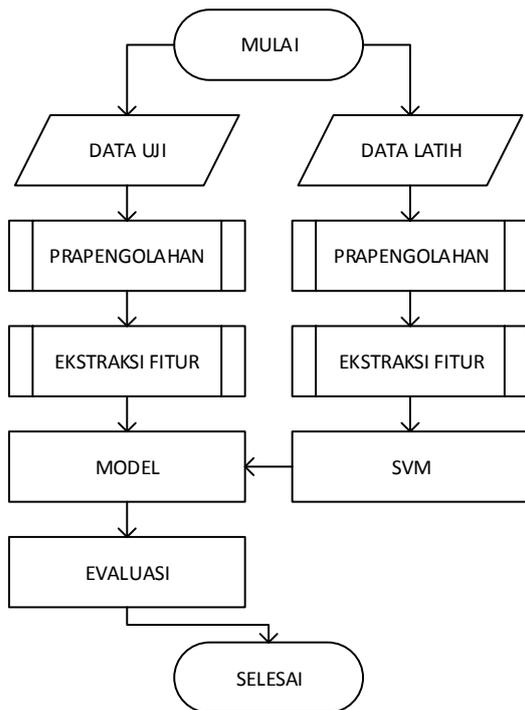
Karena dataset dengan Bahasa Indonesia yang dapat digunakan untuk melakukan analisis sentimen sulit diperoleh maka dibuat dataset yang merupakan modifikasi dari *Large Movie Review Dataset* (Maas *et al.*, 2011) tersebut terdiri dari 50.000 komentar yang terbagi menjadi dua kelas yaitu 25.000 positif dan 25.000 negatif. Kemudian dipilih 1000 komentar pertama dan diterjemahkan dengan menggunakan *Google Translate*. Dari 1000 komentar tersebut ada 500 komentar positif dan 500 komentar negatif. Dataset dapat diperoleh dari alamat berikut:

<https://github.com/ArthaAgastya/dataset/tree/master/MovieReviewID>

### 2.3 Analisis Sentimen

Analisis sentimen disebut juga dengan *opinion mining* (penambangan opini) merupakan proses untuk mengekstrak suatu opini atau pendapat dari dokumen untuk topik tertentu (Kaur, Mangat and Nidhi, 2017). Seperti pada contoh ulasan film dibawah ini memiliki sentimen positif atau negative.

1. Film avenger sangat bagus karena banyak aksinya. (positif)
2. Film dilan jelek banget dan pemainnya kurang cocok tapi karena dipaksa pacar nonton ya jadi nonton. (negatif)
3. Saya kecewa sekali dengan cerita pada naruto karena tokoh utamanya jadi lemah. (negatif)
4. Teman saya bilang kalau dia menonton avatar sampai menangis bahagia. (positif)



Gambar 1. Diagram Alir Analisis Sentimen

Analisis sentimen bertugas untuk memprediksi sentimen yang ada dari ulasan film tersebut.

Langkah – langkah analisis sentimen ditunjukkan pada Gambar 1. Data terbagi menjadi dua yaitu data latih dan data uji. Proporsi yang digunakan adalah 70% data latih dan 30% data uji yang sesuai dengan penelitian sebelumnya (Franky and Manurung, 2008). Kemudian dilakukan prapengolahan, ekstraksi fitur, pemodelan dengan SVM, evaluasi dengan *confusion matrix*. Hasil dari *confusion matrix* tersebut menjadi parameter kinerja dari analisis sentimen.

**2.4 Prapengolahan**

Prapengolahan merupakan proses untuk menormalkan kata – kata dari kalimat atau frase yang akan dimodelkan dengan metode SVM. Hal ini dilakukan untuk mendapatkan data latih yang baik dan fitur – fitur yang diekstrak nantinya sesuai dengan yang diinginkan. Pada Prapengolahan terdapat 5 proses yang terdiri dari:

1. *Tokenization* atau tokenisasi

Proses penghilangan tanda baca pada kalimat yang ada dalam dokumen sehingga menghasilkan kata - kata yang berdiri sendiri - sendiri.

2. *Case Folding*

Merubah semua kata menjadi lowercase (huruf kecil) atau uppercase (huruf besar)

3. *Stemming*

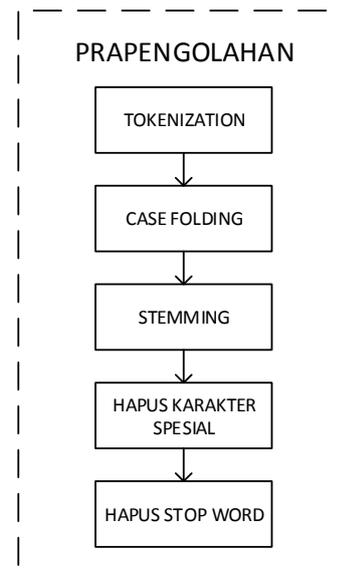
Proses penghilangan imbuhan, akhiran, dan sisipan.

4. Hapus karakter spesial

Menghapus karakter yang khusus seperti ?, \$, &, \*, %, @, (, ), dan ~.

5. Hapus stopwords

Menghapus stopwords seperti di, ke, dari, dan, atau, berikan, kalau, akan, dan lain – lain. Stopword yang digunakan berdasarkan penelitian ini (Tala, 2003).



Gambar 2. Diagram Alir Prapengolahan

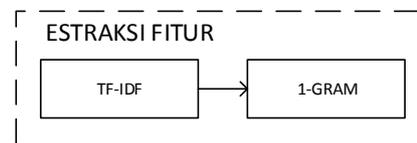
**2.5 Ekstraksi Fitur**

Setelah dilakukan prapengolahan maka masuk ke proses ekstraksi fitur yang menggunakan dua metode yaitu 1-gram dan Term Frequency Inverse Document Frequency (TF-IDF). Dengan menggunakan 1-gram maka term atau kata dapat langsung diberikan bobot tanpa ada kombinasi penggabungan dua kata atau lebih.

$$IDF_i = \log \left( \frac{D}{df_i} \right) \tag{1}$$

$$W_{ij} = TF_{ij} \times IDF_i \tag{2}$$

TF-IDF (Agastya, Adji and Setiawan, 2017) merupakan metode untuk memberikan bobot pada suatu kata atau term sehingga kata – kata yang sering muncul dan unik terhadap jumlah dokumen memiliki bobot yang besar. TF-IDF digunakan untuk mendapatkan kata – kata yang menjadi utama di dokumen tersebut. Jika  $TF_{ij}$  menyatakan frekuensi dari *term i* yang muncul pada dokumen *j*,  $IDF_i$  menyatakan *inverse document frequency* dari suatu *term i* terhadap keseluruhan dokumen, *D* menyatakan jumlah keseluruhan dokumen, dan  $df_j$  menyatakan jumlah dokumen yang mengandung term, untuk menghitung bobot dari term  $W_{ij}$  digunakan Persamaan (1) dan (2).

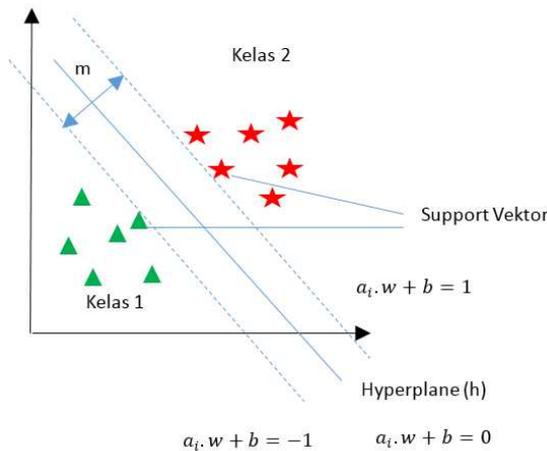


Gambar 3. Diagram Alir Ekstraksi Fitur

**2.6 Support Vector Machine (SVM)**

SVM menjadi algoritma yang paling sering digunakan pada awal tahun 90’an hingga awal 2000’an. Hal ini dikarenakan SVM memiliki kemampuan untuk menemukan solusi global yang optimal. Selain itu SVM mampu memberikan hasil yang optimal untuk data latih yang kecil dan tahan terhadap outlier. SVM mencari hyperplane dengan margin paling besar yang mungkin

antara dua kategori. Dengan margin yang besar tersebut maka kemungkinan model yang dihasilkan overfit menjadi kecil.



Gambar 4. Support Vector Machine (SVM)

Langkah – langkah yang dilakukan dengan SVM adalah sebagai berikut:

- 1) Dataset = {a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, ..., a<sub>n</sub>} dengan label atau kelas dari a adalah y<sub>i</sub> ∈ {+1,-1}.
- 2) Untuk mendapatkan persamaan hyperplane yang optimal maka perlu untuk meminimalkan nilai empirical risk (R) dengan persamaan (3).

$$R(f) = \min \frac{1}{n} \sum_{i=1}^n L(f(a_i), y_i) \quad (3)$$

- 3) Dengan meminimalkan nilai R maka diperoleh hyperplane dengan nilai m sebagai margin yang maksimal seperti pada Gambar 1. Hyperplane dituliskan dengan persamaan (4).

$$a_i \cdot w + b = 0 \quad (4)$$

- 4) Kemudian data diprediksi dengan persamaan (5) dan (6).

$$a_i \cdot w + b \leq -1 \text{ untuk } e_i = -1 \quad (5)$$

$$a_i \cdot w + b \geq +1 \text{ untuk } e_i = +1 \quad (6)$$

### 2.6.1 Confusion matrix

Evaluasi hasil klasifikasi merupakan hal yang wajib dilakukan untuk mengetahui performa dari sistem klasifikasi. *Confusion matrix* merupakan alat ukur yang standar digunakan untuk mengetahui seberapa akurat hasil perkiraan dari sistem klasifikasi. Untuk kasus kelas yang biner atau dua kelas maka terdapat empat istilah yang menjadi acuan untuk mendapatkan keakuratan dari sistem klasifikasi. Istilah yang digunakan adalah sebagai berikut:

1. *True Positif* (TP)

*True Positif* adalah merupakan data yang klasifikasi riilnya positif dan diprediksi positif.

2. *True Negative* (TN)

*True Negative* adalah merupakan data yang klasifikasi riilnya negatif dan diprediksi negatif.

3. *False Positif* (FP)

*False Positif* adalah merupakan data yang klasifikasi riilnya negatif dan diprediksi positif.

4. *False Negative* (FN).

*False Negative* adalah merupakan data yang klasifikasi riilnya positif dan diprediksi negatif.

*Confusion matrix* dibentuk dari ke empat informasi tersebut seperti ditunjukkan pada Tabel 1.

Tabel 1. Confusion matrix

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Riil Positif	True Positif (TP)	False Negatif (FN)
Riil Negatif	False Positif (FP)	True Negative (TN)

Berdasarkan ke empat nilai tersebut dapat diperoleh turunannya, yaitu:

1. Akurasi

Akurasi merupakan seberapa akurat sistem dapat mengklasifikasikan secara benar.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

2. Presisi

Presisi merupakan seberapa tepat sistem dapat memprediksi target positif terhadap jumlah data yang diprediksi positif.

$$Presisi = \frac{TP}{TP + FP} \quad (8)$$

3. Recall

Recall merupakan seberapa tepat sistem memprediksi target positif terhadap jumlah data yang riil positif.

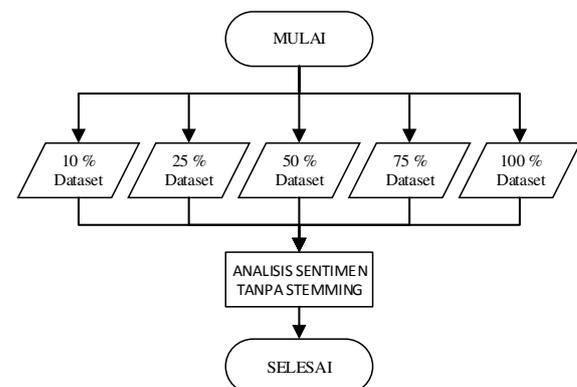
$$Recall = \frac{TP}{TP + FN} \quad (9)$$

4. F1-Score

F1-Score merupakan alternatif dari akurasi karena akurasi tidak cocok untuk mengukur performa klasifikasi dari data yang tidak seimbang seperti data kelas yang diprediksi hanya 10 dari 10.000 data. Nilai F1-Score diperoleh dari kelipatan dua dari perkalian nilai presisi dan recall dibagi dengan jumlah presisi dan recall seperti ditunjukkan pada persamaan (10).

$$F1 = 2 \cdot \frac{Presisi \cdot Recall}{Presisi + Recall} \quad (10)$$

### 2.7 Skenario Pengujian



Gambar 5. Diagram Alir Pengujian Tanpa Stemming

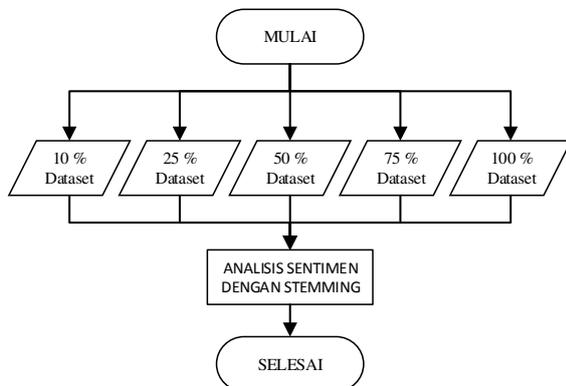
Pengujian dilakukan dengan dua skenario, yaitu:

1. Analisis sentimen tanpa melakukan *stemming* pada proses prapengolahan seperti ditunjukkan pada Gambar 5.
2. Analisis sentimen dengan melakukan *stemming* pada proses prapengolahan seperti ditunjukkan pada Gambar 6.

Kedua skenario tersebut menggunakan dataset yang bertingkat jumlah datanya. Terdapat lima model dataset, yaitu:

1. Jumlah data yang digunakan 100 atau 10% dataset. Terdapat 50 komentar positif dan 50 komentar negatif.
2. Jumlah data yang digunakan 250 atau 25% dataset. Terdapat 125 komentar positif dan 125 komentar negatif.
3. Jumlah data yang digunakan 500 atau 50% dataset. Terdapat 250 komentar positif dan 250 komentar negatif.
4. Jumlah data yang digunakan 750 atau 75% dataset. Terdapat 375 komentar positif dan 375 komentar negatif.
5. Jumlah data yang digunakan 1000 atau 100% dataset. Terdapat 500 komentar positif dan 500 komentar negatif.

Evaluasi dilakukan dengan *confusion matrix* dan menghitung waktu yang diperlukan untuk menyelesaikan analisis sentimen.



Gambar 6. Diagram Alir Pengujian dengan *Stemming*

### 2.8 Analisis Hasil Pengujian

Tabel 2. Hasil Evaluasi Pengujian Tanpa *Stemming*

Dataset	Akurasi	Precision	Recall	F1 Score
100	<b>0.77</b>	<b>0.7</b>	<b>0.64</b>	<b>0.67</b>
250	0.75	0.83	0.63	0.72
500	<b>0.71</b>	<b>0.73</b>	0.64	0.69
750	0.8	0.82	0.8	0.81
1000	<b>0.81</b>	<b>0.85</b>	<b>0.78</b>	<b>0.81</b>

Seperti yang ditunjukkan pada Table 2 dan Table 3, *Stemming* dapat meningkatkan akurasi pada dataset dengan jumlah data 250 dan 750. Namun pada dataset dengan jumlah data 100, 500, dan 1000 akurasinya menurun. Tidak ditemukan pola dari hasil yang diperoleh sehingga dapat disimpulkan *stemming* tidak

selalu meningkatkan performa analisis sentimen. Namun ada hal yang berbeda pada dataset 500 *stemming* unggul di recall dan F1 score. Ternyata hal ini sesuai dengan pernyataan yang disebutkan pada penelitian (Adriani *et al.*, 2007) yang mana *stemming* dapat meningkatkan recall dan menurunkan presisi.

Tabel 3. Hasil Evaluasi Pengujian dengan *Stemming*

Dataset	Akurasi	Precision	Recall	F1 Score
100	0,7	0,6	0,55	0,57
250	<b>0,77</b>	<b>0,84</b>	<b>0,68</b>	<b>0,75</b>
500	0,7	0,68	<b>0,73</b>	<b>0,7</b>
750	<b>0,82</b>	<b>0,86</b>	<b>0,79</b>	<b>0,82</b>
1000	0,77	0,81	0,76	0,78

Ket. Nilai yang dicetak tebal menyatakan bahwa nilainya lebih besar dari pasangan nilai dari table lain.

Jika dilihat dari waktu komputasinya maka *stemming* meningkatkan waktu komputasi yang sangat signifikan. Peningkatan rata – ratanya adalah 23.093 % atau 231 kali lipat. Peningkatan waktu komputasi sangat dipengaruhi dengan algoritma *stemming* yang digunakan. Seperti dijelaskan pada penelitian (Agusta, 2009), Algoritma NA membutuhkan waktu 50 kali lebih lama dibandingkan dengan Algoritma Porter. Karena Algoritma Sastrawi dibangun dari algoritma NA maka waktu komputasinya tinggi dan mungkin lebih rendah dari NA original karena ada pengurangan ukuran term atau kata seperti pada penelitian (Arifin, Mahendra and Ciptaningtyas, 2009).

Tabel 4. Waktu komputasi Pengujian

Dataset	Waktu			
	Tanpa <i>Stemming</i> (detik)	<i>Stemming</i> (detik)	Selisih (detik)	Peningkatan (%)
100	2,2	533,9	531,7	24268
250	5,2	1217,2	1212	23407
500	10,9	2428,7	2417,8	22281
750	16,1	3703,7	3687,6	23004
1000	20,9	4703,4	4682,5	22504

### 3. Kesimpulan

Kesimpulan yang diperoleh dari melakukan pengujian adalah sebagai berikut:

1. *Stemming* dapat meningkatkan akurasi dan menurunkan akurasi bergantung jumlah data yang digunakan.
2. Waktu komputasi dari algoritma *stemming* sangat tinggi hingga meningkatkan waktu sentimen analisis hingga 231 kali lipat.
3. Pada pengujian dengan jumlah data 500 diperoleh hasil pengujian yang sesuai dengan algoritma NA yang mana *stemming* dapat meningkatkan recall namun menurunkan presisi.

Saran untuk penelitian selanjutnya adalah sebagai berikut:

1. Menggunakan dataset yang utuh yaitu 50.000 data yang sudah diterjemahkan ke Bahasa Indonesia.

2. Membandingkan Stemmer NA, Stemmer Sastrawi, dan Stemmer Porter.
3. Mengembangkan Stemmer Sastrawi menjadi lebih cepat dari versi aslinya.

#### Daftar Pustaka

- Aditya, B. R. (2016) 'A Framework for Sentiment Analysis Implementation of Indonesian Language Tweet on Twitter', in *International Conference on Computing and Applied Informatics 2016*, pp. 1–6. doi: 10.1088/1742-6596/755/1/011001.
- Adriani, M. *et al.* (2007) 'Stemming Indonesian: A Confix-Stripping Approach', *ACM Transactions on Asian Language Information Processing (TALIP)*, 6(4), pp. 1–33. doi: 10.1145/1316457.1316459.
- Agastya, I. M. A., Adji, T. B. and Setiawan, N. A. (2017) 'Comparison of distributed K-means and distributed fuzzy C-means algorithms for text clustering', *Communication in Science and Technology*, 2(1), pp. 11–17. doi: <http://dx.doi.org/10.21924/cst.2.1.2017.46>.
- Agusta, L. (2009) 'Perbandingan Algoritma Stemming Porter Dengan Algoritma Nazief & Adriani Untuk Stemming Dokumen Teks Bahasa Indonesia', in *Konferensi Nasional Sistem dan Informatika 2009*, pp. 196–201.
- Aliandu, P. (2013) 'Twitter Used by Indonesian President: An Sentiment Analysis of Timeline', (December), pp. 2–4.
- Arifin, A. Z., Mahendra, I. P. A. K. and Ciptaningtyas, H. T. (2009) 'Enhanced Confix Stripping Stemmer and Ants Algorithm For Classifying News Document In Indonesian Language', in *Proceeding of International Conference on Information & Communication Technology and Systems (ICTS)*, pp. 149–158.
- Buntoro, G. A. (2017) 'Analisis Sentimen Calon Gubernur DKI Jakarta 2017 Di Twitter', *32 Integer Journal Maret*, 1(2016), pp. 32–41.
- Chifu, E. Ş., Letia, T. Ş. and Chifu, V. R. (2015) 'Unsupervised aspect level sentiment analysis using Ant Clustering and Self-organizing Maps', *2015 International Conference on Speech Technology and Human-Computer Dialogue, SpeD 2015*. doi: 10.1109/SPED.2015.7343075.
- Faradhillah, N. Y. A., Kusumawardani, R. P. and Hafidz, I. (2016) 'Eksperimen Sistem Klasifikasi Analisa Sentimen Twitter pada Akun Resmi Pemerintah Kota Surabaya Berbasis Pembelajaran Mesin', *Prosiding Seminar Nasional Sistem Informasi Indonesia 2016*, pp. 15–24.
- Franky and Manurung, R. (2008) 'Machine Learning-based Sentiment Analysis of Automatic Indonesian Translations of English Movie Reviews', *International Conference on Advanced Computational Intelligence and Its Application (ICACIA)*, 1.
- Haykin, S. (2009) *Neural Networks and Learning Machines*. 3rd edn, Prentice Hall. 3rd edn. New Jersey: Pearson Prentice Hall.
- Hidayatullah, A. F., Ratnasari, C. I. and Wisnugroho, S. (2016) 'The Influence of Stemming on Indonesian Tweet Sentiment Analysis', *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 14(June), pp. 665–673. doi: 10.12928/TELKOMNIKA.v14i1.3113.
- Kaur, H., Mangat, V. and Nidhi (2017) 'A Survey of Sentiment Analysis techniques', in *International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC 2017)*, pp. 921–925. doi: 10.5715/jnlp.13.3\_201.
- Librian, A. (2017) *High quality stemmer library for Indonesian Language (Bahasa)*, *GitHub*. Available at: <https://github.com/sastrawi> (Accessed: 20 February 2018).
- Liu, B. (2012) 'Sentiment Analysis and Opinion Mining', *Synthesis Lectures on Human Language Technologies*, 5(1), pp. 1–167. doi: 10.2200/S00416ED1V01Y201204HLT016.
- Liu, B. and Zhang, L. (2012) 'A survey of opinion mining and sentiment analysis', *Mining Text Data*, 9781461432, pp. 415–463. doi: 10.1007/978-1-4614-3223-4\_13.
- Maas, A. L. *et al.* (2011) 'Learning Word Vectors for Sentiment Analysis', *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. doi: 978-1-932432-87-9.
- Suresh, H. and S., G. R. (2016) 'An unsupervised fuzzy clustering method for twitter sentiment analysis', *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pp. 80–85. doi: 10.1109/CSITSS.2016.7779444.
- Surjandari, I., Naffisah, M. S. and Prawiradinata, M. I. (2015) 'Text Mining of Twitter Data for Public Sentiment Analysis of Staple Foods Price Changes', *Journal of Industrial and Intelligent Information*, 3(3), pp. 253–257. doi: 10.12720/jiii.3.3.253-257.
- Tahitoe, A. D. and Purwitasari, D. (2010) *Implementasi Modifikasi Enhanced Confix Stripping Stemmer Untuk Bahasa Indonesia dengan Metode Corpus Based Stemming*. Institut Teknologi Sepuluh November (ITS).
- Tala, F. Z. (2003) *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*.
- UN Global Pulse (2014) 'Mining Indonesian Tweets to Understand Food Price Crises', (February), pp. 1–18.
- Wira, P. and Indra, D. (2010) 'Pengklasifikasian Otomatis Berbasis Ontologi untuk Artikel Berita Berbahasa Indonesia', *MAKARA, TEKNOLOGI*, 14(1), pp. 29–35.