

Analisis Sentimen Terhadap Ulasan Film Menggunakan Algoritma *Random Forest*

Muhammad Asjad Adna Jihad¹, Adiwijaya², Widi Astuti³

^{1,2,3} Universitas Telkom, Bandung

¹adnajihad@students.telkomuniversity.ac.id, ²adiwijaya@telkomuniversity.ac.id,

³astutiwidi@telkomuniversity.ac.id

Abstrak

Meningkatnya antusiasme masyarakat dalam menonton film membuat semakin banyak bermunculan forum-forum berisikan diskusi maupun komentar/review para penontonnya terhadap film itu. Hasil review yang ada pada forum-forum tersebut dikumpulkan menjadi sebuah informasi penting yang dapat terjadi kesalahan konteks dalam penyampaian apabila masyarakat salah dalam mengartikan dan memahaminya. Teknik analisis sentimen digunakan untuk mencegah hal tersebut, dimana analisis sentimen akan melakukan analisa pada tiap teks untuk mengetahui aspek dan tingkatan nilai sentimen (positif atau negatif). Sebelum memulai pengolahan hasil *review* yang telah dikumpulkan, terdapat tahap yang akan dilakukan yaitu tahap *preprocessing* agar data dapat lebih mudah untuk melanjutkan ke tahap selanjutnya. Kemudian dilakukan *feature extraction* menggunakan *Word2Vec*, serta menggunakan metode *Random Forest* untuk melakukan pengklasifikasian sentimen. Dari penelitian yang telah dilakukan, didapatkan penelitian yang menggunakan *skip-gram Word2Vec* 300 dimensi, kemudian menerapkan *Adaptive Boosting* kepada *base model*, memiliki nilai akurasi terbaik sebesar 75.76%.

Kata kunci : *review imdb, analisis sentimen, preprocessing, Word2Vec, random forest*

Abstract

The more public interest in watching the film increases, the more forums that contain discussions and comments/reviews of the audience on the film appeared. The results of the reviews in these forums became important information that could be at risk if the public happens to misinterpret during understanding it. Sentiment analysis techniques are used to prevent this, where sentiment analysis will analyze each text to determine the aspect and level of sentiment (positive or negative). Before starting the processing of the review results that have been collected, there's a step that will be carried out first which is the preprocessing stage so that the data could be easier to proceed to the next stage. Then, feature extraction is performed using Word2Vec, and the Random Forest method is used to classify sentiments. From the research that has been done, it was found that a particular research scenario using the 300-dimensional Word2Vec skip-gram, and then applying Adaptive Boosting to the base model, had the best accuracy value of 75.76%.

Keywords: *review imdb, sentiment analysis, preprocessing, Word2Vec, random forest*

1. Pendahuluan

Media komunikasi baik media cetak, media elektronik, maupun media internet sering menyajikan hasil *review* dari sebuah film [2]. Hasil *review* yang ada pada berbagai platform tersebut berisi diskusi maupun komentar para penontonnya terhadap film-film tersebut. Seperti yang kita tahu, semakin besar dan semakin banyak jumlah informasi tersebut maka akan semakin berpengaruh informasi. Apabila masyarakat melakukan kesalahan dalam mengartikan dan memahami kumpulan informasi tersebut, kesalahan konteks dalam penyampaian akan memiliki potensi yang besar untuk terjadi. Teknik analisis sentimen digunakan untuk mencegah hal tersebut, dimana analisis sentimen akan melakukan analisa pada tiap teks untuk mengetahui aspek dan tingkatan nilai sentimen (positif atau negatif) seperti pada jurnal [3].

Penelitian mengenai analisis sentimen terhadap ulasan film berbahasa inggris sudah pernah dilakukan dengan berbagai metode seperti pada penelitian [1] dimana berfokus akan pengaruh penerapan tahap *Preprocessing* pada *dataset* menggunakan metode *Naïve Bayes* dan *LSTM (Long Short-Term Memory)* yang menghasilkan nilai akhir rata-rata *F1-Score* tertinggi sebesar 91.41% untuk *dataset* yang menggunakan proses *preprocessing*. Ada pula penelitian selanjutnya [4] dimana menggunakan *dataset* ulasan untuk film '*Quantum of Solace*' dan '*Star Trek*' dengan fokus untuk mengetahui subjektifitas dan polaritas kalimat memakai beberapa algoritma klasifikasi *supervised* dan *unsupervised*, untuk film '*Quantum of Solace*' didapatkan hasil akhir nilai akurasi yang berada pada lingkup 70% keatas dengan algoritma klasifikasi *supervised* serta akurasi yang berada di sekitar 60% kebawah untuk algoritma klasifikasi *unsupervised*. Bagi film '*Star Trek*' sendiri, didapatkan hasil akhir yang berada di sekitar nilai 60% untuk semua metode yang digunakan. Penelitian analisis sentimen pada *movie review* juga dilakukan oleh salah satu peneliti lainnya pada jurnal [27] dimana lebih berfokus pada teknik klasifikasi *Multinomial Naïve Bayes* dengan memakai *Gini Index Text Feature Selection* kemudian menggunakan *dataset* yang berasal dari *IMDB*, didapatkan hasil performansi terbaiknya sebesar 59.54% untuk

dataset yang menggunakan *Multinomial Naïve Bayes* serta menerapkan *Gini Index Text Feature Selection*. Penelitian terakhir yang digunakan sebagai rujukan merupakan penelitian pada jurnal [28] dengan fokus utama melihat performansi KNN dengan bantuan *Information Gain*, didapatkan hasil akhir akurasi sebesar 60% untuk penerapan KNN tanpa bantuan IG, sedangkan untuk penerapan KNN dengan bantuan IG didapatkan hasil akurasi sebesar 96.8%.

Berdasarkan hasil-hasil penelitian diatas, penelitian ini dilakukan untuk mengetahui dan menganalisa nilai sentimen dari sebuah kalimat dalam hasil *review* film berbahasa inggris dari *website* IMDB menggunakan algoritma yang dapat memproses & memetakan informasi menjadi sebuah vektor, yaitu menggunakan metode *Word2Vec*. Penggunaan metode *Random Forest* juga diterapkan dalam penelitian ini untuk menganalisa dan melakukan klasifikasi terhadap nilai sentimen hasil *review* dalam forum *website* IMDB.

Topik yang akan dibahas pada penelitian kali ini yaitu mengetahui pengaruh beberapa faktor terhadap hasil akhir penelitian seperti penerapan *stemming* terhadap *dataset*, dan jumlah dimensi *skip-gram Word2Vec* yang digunakan. Untuk batasan pada penelitian ini diantaranya *dataset* yang digunakan diambil dari *website* IMDB, *dataset* berjumlah 50.000 *review* dengan kelas positif dan negatif, bahasa *dataset* yang digunakan adalah bahasa inggris, kemudian pada penelitian ini tidak berfokus pada pencarian nilai n-tree yang optimal pada tahap klasifikasi.

Tujuan dari penelitian tugas akhir ini berdasarkan latar belakang dan perumusan masalah yang telah dijabarkan sebelumnya yaitu melakukan analisis sistem pada *review* film berbahasa inggris menggunakan algoritma *Random Forest* dengan *Word2Vec* untuk metode ekstraksi fitur, kemudian mengukur hasil performansi dari sistem yang dibangun.

Bagian selanjutnya yang akan dibahas adalah pada bab 2 membahas studi literatur yang terkait dengan penelitian ini. Pada bab 3 membahas teori dan rancangan sistem penelitian. Hasil dan analisis dibahas pada bab 4. Kesimpulan dari penelitian yang sudah dilakukan dibahas pada bab 5.

2. Studi Terkait

Kata *Sentiment analysis* pertama kali muncul pada tahun 2003, berdasarkan jurnal [7]. *Sentiment analysis* sendiri adalah salah satu cabang penelitian dari *text mining* yang berkolaborasi dengan *Natural Language Processing* (NLP) yang berfokus untuk mengekstraksi pola dan menganalisis informasi berupa opini dari suatu dokumen teks [1]. *Sentiment analysis* mulai muncul sebagai topik yang penting seiring dengan semakin bertambahnya interaksi menggunakan *social media*, penggunaan berbagai forum dan blog, serta komentar dan penilaian melalui berbagai *website E-commerce*[8].

Penelitian analisis sentimen terhadap ulasan film berbahasa inggris sudah pernah dilakukan dengan berbagai metode seperti pada penelitian [1] yang dilakukan oleh Rani Sari Murti. Untuk penelitian ini, sang peneliti menggunakan pendekatan *deep learning* yaitu metode RNN (*Recurrent Neural Network*) yang dimodifikasi dengan arsitektur LSTM (*Long Short-Term Memory*), metode tersebut dipilih karena menurut peneliti, LSTM sangat cocok untuk menyelesaikan masalah yang bersifat klasifikasi. Penelitian ini lebih berfokus akan pengaruh penerapan tahap *Preprocessing* pada *dataset*. Hasil akhir akan penelitian ini adalah didapatkan bahwa *preprocessing* dapat meningkatkan performa sistem, dibuktikan dengan rata-rata *F1-Score* tertinggi sebesar 91.41% untuk *dataset* yang menggunakan *preprocessing* sedangkan untuk *dataset* yang menggunakan *tokenization* memiliki rata-rata *F1-Score* tertinggi sebesar 90.88%. Hal tersebut terjadi karena *preprocessing* menghilangkan kata yang tidak dibutuhkan serta membuat kata pada model lebih terstruktur sehingga sistem menjadi lebih efisien dan performa semakin meningkat.

Penelitian selanjutnya [4] yang dilakukan oleh Kuat Yessenov dimana peneliti menggunakan *dataset* ulasan berbahasa inggris untuk film '*Quantum of Solace*' dan '*Star Trek*' dengan tiga nilai sentimen (positif, negatif, dan netral) yang didapatkan dari *website* Digg. Penelitian kali ini memakai tiga algoritma klasifikasi *supervised* diantaranya, *Naïve Bayes*, *Maximum Entropy* serta *Decision Tree*, dan juga satu algoritma klasifikasi *unsupervised* yaitu *K-Means Clustering* dengan menggunakan model *bag-of-words* untuk melakukan ekstraksi fitur, dengan fokus untuk mengetahui subjektifitas dan polaritas kalimat. Peneliti memilih menggunakan metode klasifikasi *supervised* dan *unsupervised* sendiri dengan alasan, dikutip langsung dari jurnal, pertama untuk klasifikasi *supervised* memiliki *classifier* yang dilatih pada contoh-contoh *label* yang mirip dengan data uji. Sebaliknya, klasifikasi *unsupervised* memasang *label* berdasar perbedaan internal (jarak) antara *data point*. Hasil akhir dari penelitian ini adalah didapatkan nilai akurasi untuk film '*Star Trek*', berada di sekitar nilai 60% bagi semua metode yang digunakan. Untuk film '*Quantum of Solace*' didapatkan nilai akurasi yang berada pada lingkup 70% keatas untuk metode *Decision Tree*, *Naïve Bayes*, dan *Max-Entropy*, sedangkan untuk metode *K-Means Clustering* berada di sekitar 60% kebawah. Kesimpulan dari penelitian yang dilakukan oleh peneliti adalah mengetahui penggunaan model *bag-of-words* sederhana dapat memiliki performa yang cukup baik dan dapat ditingkatkan lebih baik dengan pilihan fitur yang didasari oleh informasi sintaks dan semantik. Penggunaan *corpus* yang mirip dengan *corpus* lainnya belum berarti memiliki hasil performansi yang sama untuk klasifikasi meskipun mengandung berbagai polaritas kata yang berasal dari banyak ulasan film.

Kemudian ada penelitian yang dilakukan oleh Riko Bintang Purnomoputra seperti pada jurnal [27] menggunakan *dataset* yang berasal dari IMDB sebanyak 4000 ulasan terdiri dari masing-masing 2000 label negatif dan positif. Fokus utama pada penelitian ini adalah penggunaan teknik klasifikasi *Multinomial Naïve Bayes* (MNNB) dan *Gini Index Text (GIT) Feature Selection*, dimana kedua teknik tersebut dipilih setelah peneliti melakukan studi terkait akan kasus serupa. Terdapat dua skenario diantaranya pertama, pengujian tanpa fitur seleksi, dan kedua, pengujian dengan fitur seleksi, untuk dapat mengetahui pengaruh penggunaan MNNB dan GIT. Untuk skenario pertama, dilakukan 3 pembagian data untuk pengujian yaitu 2000, 3000, dan 4000 data, didapatkan nilai akurasi terbaik sebesar 62.5% dan *F1-Score* sebesar 64.11% untuk pengujian sebanyak 2000 data ulasan lalu menerapkan MNNB. Kemudian untuk skenario kedua, masih sama dengan sebelumnya dimana dilakukan 3 pembagian data dengan penerapan *GIT Feature Extraction*, didapatkan hasil terbaik akurasi sebesar 62.5% untuk data sebanyak 2000 data kemudian menerapkan MNNB dan GIT. Kesimpulan yang ditarik oleh sang peneliti ialah penggunaan metode *Multinomial Naïve Bayes* dengan *Gini Index Text Feature Selection* dapat meningkatkan performansi.

Penelitian terakhir dilakukan oleh Novelty Octaviani Faomasi Daeli [28], dimana sang peneliti memakai *polarity v.2.0* dari *Cornell review dataset* sebanyak 2000 ulasan dengan masing-masing label sebanyak 1000 untuk positif dan negatif. Sang peneliti berfokus pada hasil performansi penggunaan KNN dengan bantuan *Information Gain* kemudian membandingkannya dengan metode lain seperti SVM, *Naïve Bayes*, serta *Random Forest*. Terdapat 2 skenario dalam penelitian ini, pertama, program hanya menerapkan KNN tanpa memakai *Information Gain* lalu membandingkannya dengan hasil metode lain, kedua, *Threshold* untuk *Information Gain* dibagi 5 yaitu lebih dari 0.1, 0.2, 0.3, 0.4, dan 0.5 lalu seperti skenario sebelumnya menerapkan KNN namun kali ini beserta bantuan IG kemudian membandingkannya dengan hasil metode lainnya. Didapatkan hasil akhir yaitu performansi KNN tanpa bantuan IG ialah akurasi sebesar 60% dengan $K=3$, sedangkan performansi KNN dengan bantuan IG mendapat nilai yang jauh lebih tinggi yaitu akurasi sebesar 96.8% untuk $K=3$. Sang peneliti menyimpulkan bahwa penggunaan IG dapat meningkatkan performansi karena IG dapat melakukan reduksi fitur yang sekiranya kurang relevan dengan *class* yang digunakan.

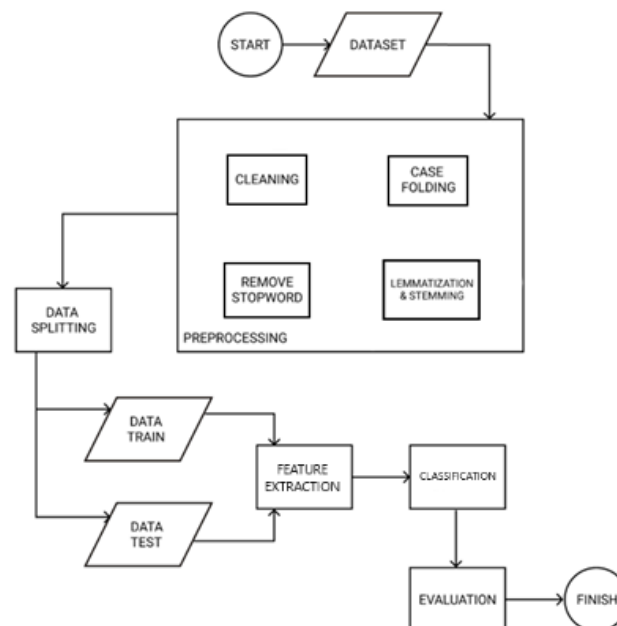
Berdasarkan dari hasil-hasil penelitian di atas, peneliti mengambil beberapa poin dan mengembangkannya diantaranya pertama, merujuk ke penelitian [27] dimana sang peneliti menggunakan *dataset* yang berasal dari IMDB sebanyak 4000 data, dapat dikembangkan lebih jauh lagi seperti yang dilakukan pada penelitian kali ini dengan menggunakan 50000 ulasan *dataset* IMDB. Kedua, disebutkan pada penelitian [1] bahwa *preprocessing* dapat meningkatkan performa sistem sehingga menjadi salah satu topik pada penelitian ini dimana mengetahui pengaruh penerapan *preprocessing* terhadap *dataset* terutama pada tahap *stemming*. Ketiga, penelitian [4] menyatakan bahwa penggunaan *bag-of-words* sederhana dapat memiliki performa yang cukup baik dan dapat ditingkatkan lebih baik lagi dengan pilihan fitur yang didasari oleh informasi sintaks dan semantik, hal tersebut dilakukan pada penelitian ini dimana peneliti mengimplementasikan fitur ekstraksi *Word2Vec* yang secara umum merupakan *bag-of-words* tingkat lanjut. Poin terakhir diambil dari penelitian [28] dimana disebutkan bahwa penggunaan *Information Gain* dapat meningkatkan performansi karena fungsi reduksi fiturnya, pernyataan tersebut diterapkan pada penelitian kali ini dalam bentuk parameter *Grid Search CV* untuk tahap *Hyperparameter Tuning* yaitu menggunakan *entropy* dimana *entropy* akan menghitung kualitas berdasar nilai *information gain*.

Sebelum pengklasifikasian data dapat dimulai, dibutuhkan suatu algoritma untuk merubah teks pada review menjadi data yang dapat diolah komputer. Kesimpulan pada penelitian [4] menyebutkan bahwa penggunaan model *bag-of-words* sederhana dapat memiliki performa yang cukup baik dan dapat ditingkatkan lebih baik dengan pilihan fitur yang didasari oleh informasi sintaks dan semantik, peneliti memutuskan untuk mengembangkan pernyataan tersebut dengan menggunakan *Word2Vec* yang dimana bukanlah sebuah algoritma melainkan satu kesatuan dari dua teknik, yaitu CBOW (*Continuous Bag-Of-Words*) serta *Skip-Gram*. Pada jurnal [6] disebutkan, vektor juga dibutuhkan karena dapat merepresentasikan setiap kata menjadi satu vektor agar lebih mudah dibaca oleh komputer. *Word2Vec* dipilih dari sekian metode ekstraksi fitur untuk penelitian kali ini karena *Word2Vec* menarik dan memahami konteks dari berbagai frase sehingga sangat cocok untuk menganalisa berbagai ulasan, komentar, serta umpan balik, seperti yang dilakukan pada penelitian kali ini.

Metode *Random Forest* digunakan sebagai metode klasifikasi terhadap hasil review. Metode ini dipilih karena beberapa alasan tertentu diantaranya, *Random Forest* sendiri merupakan gabungan prediksi dari banyak *decision tree* dalam satu model sehingga dapat membuat *Random Forest* memiliki kemungkinan lebih kecil untuk terjadinya *overfitting* atau suatu situasi dimana model sangat bergantung pada satu dataset sehingga apabila digunakan pada dataset lain akan didapat hasil yang jauh berbeda. Selain itu, *Random Forest* memiliki beberapa set *Hyperparameter* yang memerlukan penyetulan manual yang dimana dapat membantu performansi sistem pada penelitian kali ini.

3. Sistem yang Dibangun

Tahap ini menjelaskan mengenai sistem yang akan dibangun terkait dengan penelitian ini. Berikut merupakan *flowchart* dari perancangan analisis sistem sentimen *review* film :



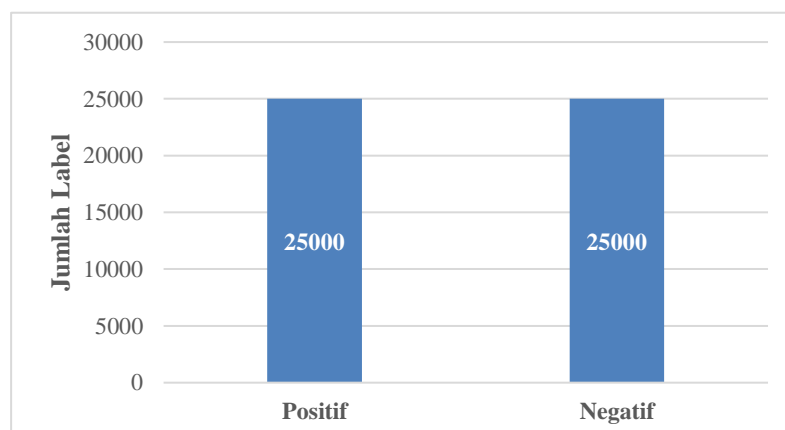
Gambar 1. Gambaran Umum Sistem

Gambar diatas menjelaskan bahwa dalam perancangan analisis sistem sentimen di penelitian ini, akan melewati beberapa tahapan diantaranya yaitu pembacaan *dataset*, tahap *Preprocessing*, dimana terdiri atas *data cleaning*, *case-folding*, *stopword removal*, dan sub-tahapan terakhir yaitu *Lemmatization* serta *Stemming*, kemudian data dibagi menjadi dua pada tahapan *data splitting*, dimana data menjadi *data train* dan *data test* dengan rasio 8:2. Kedua data tersebut akan digunakan pada tahapan selanjutnya yaitu, *Word2Vec feature extraction*. Data yang telah diolah akan masuk ke tahap pengklasifikasian menggunakan algoritma *Random Forest*. Evaluasi dilakukan sebagai tahapan terakhir untuk penganalisan hasil penelitian lebih lanjut.

3.1 Dataset

Data yang akan digunakan pada penelitian ini adalah merupakan *dataset* yang dikumpulkan dari penelitian terdahulu dengan hasil *review* yang didapat dari *website* forum *review* film internasional yaitu *imdb.com*. Dari situs tersebut dikumpulkan sekitar 50.000 data hasil *review* film, dimana data tersebut akan melewati tahap klasifikasi yang sebelumnya telah dilakukan *cleaning*, serta *feature extraction*.

Untuk keterangan *label* dari *dataset* yang digunakan, dapat dilihat pada gambar yang disajikan dibawah ini dimana terlihat seimbang jumlah data yang memiliki *label* 0 (negatif) dan data yang memiliki *label* 1 (positif) yaitu masing-masing sebanyak 25.000 sehingga merupakan *dataset balanced*. Hal ini dilakukan pada penelitian terdahulu tanpa ada alasan tertentu, peneliti sebelumnya lebih mempertimbangkan kekurangan yang dapat dicapai apabila penelitian menggunakan *imbalanced dataset* diantaranya dapat menyebabkan beberapa problema dalam *training* diakibatkan karena *class/label* yang lebih banyak dapat mendominasi *class/label* yang lebih sedikit.



Gambar 2. Keterangan Label Dataset

3.2 Data Preprocessing

Pada jurnal [9] dijelaskan bahwa proses *Preprocessing* adalah proses mengubah *form* menjadi data terstruktur sesuai dengan kebutuhannya untuk proses *data mining*, biasanya numerik.. Tahapan *preprocessing* yang dilakukan pada penelitian ini adalah *data cleaning*, *case-folding*, *stopword removal*, dan sub-tahapan terakhir yaitu *Lemmatization* serta *Stemming*.

A. Data Cleaning

Data Cleaning adalah proses pendeteksian dan perbaikan (atau hapus) data rusak atau salah, tabel dan database. Istilah tersebut mengacu pada mengidentifikasi data yang tidak lengkap, tidak akurat, tidak akurat dan tidak relevan, dan kemudian mengganti, mengubah atau menghapus data dengan data kotor. [17]. Pada penelitian ini, *Data Cleaning* yang dilakukan terhadap *dataset* terkait diantaranya adalah melakukan pengecekan *dataset*, memastikan tidak adanya data kosong (*null*) serta *break lines* (</br>).

B. Case-Folding

Case folding adalah salah satu bentuk *pra-pemrosesan teks* yang paling sederhana dan paling efisien, tetapi sering diabaikan. Proses *case folding* adalah proses yang mengubah semua huruf menjadi huruf kecil, yang mengubah karakter "A" dan "Z" dalam data menjadi karakter "a" dan "z" [18]. Beberapa hal dilakukan pada tahap *Case-Folding* ini terhadap *dataset*, diantaranya mengubah data text menjadi huruf kecil (*lowercase*), menghapus angka, menghapus tanda baca, serta menghapus spasi (*whitespace*).

C. Stopword Removal

Proses selanjutnya adalah proses *Stopword Removal*, dimana proses *Stopword Removal* sendiri merupakan proses penyaringan, memilih kata-kata penting dari hasil yang ditandai, yaitu kata-kata mana yang digunakan untuk merepresentasikan dokumen. [14].

Pada jurnal terkait [14] juga dijelaskan bahwa *Stopword* adalah kata umum yang tidak memberikan informasi penting (biasanya diabaikan atau dibuang, misalnya saat membuat indeks atau daftar kata). Dari dua pernyataan diatas, dapat disimpulkan bahwa proses *Stopword Removal* merupakan proses penghilangan kata yang tidak mengandung makna apa apa atau dapat disebut dengan kata umum.

D. Lemmatization & Stemming

Lemmatization & Stemming merupakan proses terakhir pada tahap *data preprocessing* dimana pada proses ini, seperti namanya, terdiri atas dua sub-proses yaitu proses *Lemmatization* dan proses *Stemming*. *Lemmatization* adalah sebuah proses untuk menemukan bentuk dasar dari sebuah kata[15]. Proses *Stemming* adalah proses pemotongan imbuhan kata yang meliputi awalan, akhiran, maupun kombinasi keduanya, dengan menjalankan algoritma tertentu.

Pada jurnal [29] didapatkan pernyataan bahwa penggunaan *Stemming* sangat berpengaruh pada jumlah kalimat yang dideteksi untuk digunakan pada penelitian, dibuktikan dengan hasil eksperimen yang dilakukan oleh Alhamda Adisoka Bimantara dimana hanya ditemukan 8 komentar yang terdeteksi sebagai *cyber-bullying*, setelah diterapkan *Stemming*, ditemukan terdapat 12 komentar yang dapat dikategorikan sebagai *cyber-bullying*.

Tabel 1. Dataset melewati tahap-tahap Preprocessing

<i>Raw Data</i>	<i>One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked.</i>
<i>Basic Cleaning</i>	<i>one of the other reviewers has mentioned that after watching just oz episode youll be hooked</i>
<i>Stopword</i>	<i>one reviewers mentioned watching oz episode youll hooked</i>
<i>Lemmatization & Stemming</i>	<i>one reviewer mention watch oz episode youll hook</i>

3.3 Data Splitting

Seperti namanya merujuk, tahap *Data Splitting* akan membagi *dataset* yang telah melalui *preprocessing & cleaning* menjadi dua bagian yaitu *data train* serta *data test* dengan proporsi tertentu. *Data train* akan digunakan untuk mencari nilai *fitness* dari model sedangkan *data test* digunakan untuk mengevaluasi hasil nilai *fitness* model tersebut.

Seperti yang umumnya diketahui, *train/test split* membagi *dataset* menjadi *train set* dan *test set*, dimana setelah dilakukan pelatihan menggunakan *train set*, model hasil pembelajaran akan dievaluasi dengan *test set* [20]. Pada penelitian yang dilakukan, peneliti menetapkan rasio 8:2 sebagai rasio pembagian dalam data splitting dengan data latih sebanyak 40.000 dan data uji sebanyak 10.000, didasari dengan *Pareto Principle*, dimana dikutip dalam aturan tersebut, “..., that 20% of the factors in most situations account for 80% of what happens.” [21].

A. K-Fold Cross Validation

Pada penelitian kali ini digunakan *K-Fold Cross Validation* untuk tahap ini karena setelah dilakukan penelitian lebih jauh terkait *data splitting*, penggunaan “*train_test_split*” dapat menimbulkan *overfitting* terhadap *dataset* karena memiliki hasil yang konstan, sehingga peneliti memutuskan untuk menggunakan *K-Fold Cross Validation* untuk menghindari alasan tersebut serta memiliki hasil performansi yang berbeda setiap kali penelitian dilakukan.

K-Fold Cross Validation sendiri merupakan salah satu metode *Cross-Validarion* yang sering digunakan oleh para peneliti lainnya dengan melipat data sebanyak *k* kemudian melakukan iterasi eksperimen sebanyak *k* [9]. Pada jurnal [1] disebutkan bahwa *K-Fold Cross Validation* terdiri atas dua tahapan, tahap pertama membagi data menjadi *K*-buah partisi dalam ukuran yang sama, tahap kedua menjadikan partisi *i* sesuai dengan iterasi ke-*i* sebagai *data test* sedangkan sisanya akan menjadi *data train*. Untuk jumlah *K* yang digunakan pada penelitian ini yaitu menggunakan *K = 5* dengan alasan *K = 5* memiliki hasil pembagian yang kurang lebih sama dengan rasio 80:20.

Tabel 2. Perbandingan Hasil Split Data

<i>train_test_split</i> (rasio 80:20)	<i>Data Train</i>		<i>Data Test</i>	
	Jumlah Data	40000	Jumlah Data	10000
	Label 0	20039	Label 0	4961
	Label 1	19961	Label 1	5039
<i>K-Fold Cross Validation</i> (K=5)	<i>Data Train</i>		<i>Data Test</i>	
	Jumlah Data	40000	Jumlah Data	10000
	Label 0	20007	Label 0	4993
	Label 1	19993	Label 1	5007

3.4 Feature Extraction

Pada langkah selanjutnya dilakukan *Feature Extraction*, merupakan tahap yang dilakukan untuk mengambil ciri/*feature* dari suatu bentuk kemudian dianalisa untuk proses selanjutnya. [16]. Untuk penelitian ini, metode *Word2Vec* dipilih untuk tahap *Feature Extraction*. *Word2Vec* sendiri merupakan algoritma *word embedding* yang memetakan setiap kata dari teks ke vektor [5]. Hasil review yang terdapat di IMDB akan dikonversi menjadi vektor.

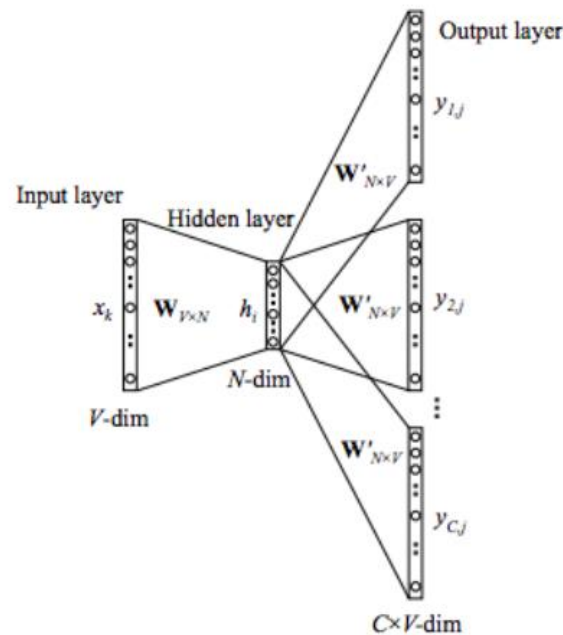
Saat menyajikan sebuah kata, *Word2Vec* mengimplementasikan *neural network* untuk menghitung konteks dan kesamaan semantik dari setiap kata (input) dalam bentuk *one-hot encoded vectors* [12]. *Word2Vec* memiliki atribut lebih yang tidak terdapat pada klasifikasi teks yang lain, hal ini yang membuat metode *Word2Vec* sangat bermanfaat[6].

A. Skip-Gram

Terdapat dua model yang dimiliki metode *Word2Vec*, *Continuous-Bag-of-Word* (CBOW) dan *Skip-Gram*. Model *Skip-Gram* adalah model yang digunakan pada penelitian kali ini, pertama kali dikenalkan oleh Mikolov et al[22]. Dari segi arsitektural, *Skip-Gram* menerima *input* yaitu *current word* untuk bisa melakukan prediksi konteks kemudian akan dipelajari distribusi dari probabilitas kata-kata dalam konteks dengan *windows* yang sudah ditentukan. *Skip-gram* merupakan metode yang efisien untuk mempelajari representasi vektor kata dalam jumlah besar pada teks tidak terstruktur[23].

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t) \quad (1)$$

Dengan w_t merupakan kata center, w_{t+j} merupakan kata setelah *center*, dan *c* adalah ukuran *training context*.

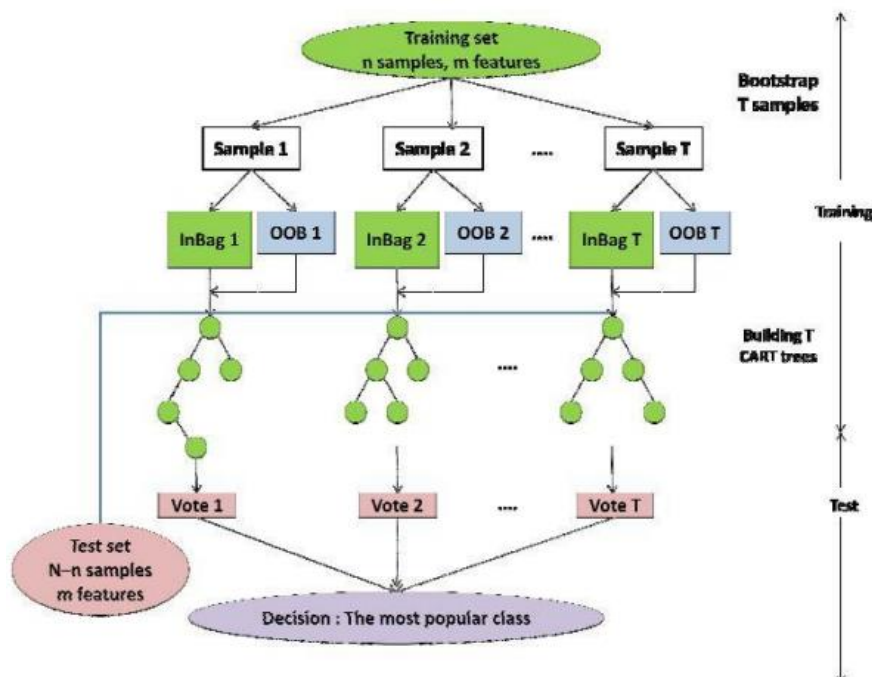


Gambar 3. Ilustrasi feeding forward Skip-Gram dengan windows = 2
 Sumber : [5]

Sebagai fokus utama di penelitian ini, model Word2Vec yang digunakan mengaplikasikan *Skip-Gram Negative Sampling* (SGNS) 100 dimensi dan 300 dimensi *corpus* yang berasal dari *corpus Wikipedia*[24]. SGNS sendiri merupakan model yang berguna untuk meningkatkan kecepatan serta kualitas dari komputasi *Word2Vec*[23]. Pemilihan penggunaan dua dimensi tersebut didasari dari penelitian yang dilakukan oleh Mikolov [12] dimana secara garis besar disebutkan bahwa penggunaan dimensi *skip-gram* sebesar 100 dimensi dan 300 dimensi merupakan jumlah paling optimal, Mikolov tidak menjamin semakin besar jumlah dimensi memiliki performa yang terbaik.

3.5 Classification

Tahap klasifikasi merupakan langkah selanjutnya setelah melewati tahapan *Data Preprocessing*. Metode yang digunakan pada penelitian ini adalah metode *Random Forest*. Dalam jurnal[3] dijelaskan bahwa *Random Forest* pada konsepnya menggabungkan beberapa *Decision Tree* untuk melakukan klasifikasi, dimana setiap *tree* akan diberikan data secara acak dari sebagian *dataset* training.



Gambar 4. Ilustrasi cara kerja Random Forest Classifier
 Sumber : [11]

Penggunaan algoritma *Random Forest* untuk proses klasifikasi data utama adalah untuk mengumpulkan pohon melalui pelatihan dalam pembentukan sampel data, di mana lebih banyak pohon dalam kasus ini mempengaruhi akurasi untuk mendapatkan yang terbaik [9]. *Random Forest* merupakan tipe klasifikasi yang tidak rentan terhadap noise serta outlier karena sifat random yang dimilikinya[10].

A. Hyperparameter

Pada jurnal [9] dijelaskan bahwa, proses klasifikasi pada *Random Forest* akan berjalan setelah semua pohon terbentuk. Setelah proses selesai, maka akan diinisialisasi dengan data sebanyak mungkin sesuai dengan nilai akurasi. Keuntungan menggunakan *Random Forest* adalah dapat mengklasifikasikan data dengan atribut yang tidak lengkap, metode ini juga sangat berguna untuk klasifikasi data dan dapat digunakan untuk mengolah data sampel yang besar. Dengan kata lain, menggunakan *Random Forest* untuk klasifikasi memberikan hasil yang baik.. *Random Forest* sendiri menggunakan konsep *Bagging* serta konsep *Bootstrapping*[13].

Seperti yang telah tercantum pada bab sebelumnya, *Random Forest* memiliki beberapa set *Hyperparameter* yang perlu dilakukan penyetelan manual untuk dapat membantu performansi sistem pada penelitian kali ini. Salah satu *Hyperparameter* tersebut adalah *criterion* atau kriteria untuk mengukur kualitas tiap split. Terdapat dua pilihan dalam *hyperparameter criterion*, yaitu *gini* serta *entropy*, dimana *gini* menggunakan nilai impuritas *gini* sedangkan *entropy* menghitung kualitas berdasar nilai *information gain*. Dalam jurnal[3] terdapat beberapa formula dalam menjalankan metode *Random Forest*, pertama adalah formula untuk menentukan *root node term* dengan menghitung Entropi(S):

$$Entropi(S) = \sum_{j=1}^k - P_j \log_2 P_j \quad (2)$$

S merupakan himpunan *dataset* kasus, sedangkan untuk k adalah banyaknya partisi s serta P_j adalah total kelas j dibagi total kasus. Formula selanjutnya untuk menghitung *Gain(A)* dimana nilai terbesar menjadi *node* tiap cabang *tree* kemudian dilakukan berulang hingga mencapai data terakhir nilai *Bagging(k)*, berikut ini adalah formula tersebut:

$$Gain(A) = Entropi(S) - \sum_{i=1}^k \frac{|S_i|}{|S|} \quad (3)$$

Dimana A merupakan atribut, S adalah data sampel yang digunakan untuk training, $|S_i|$ adalah jumlah sampel nilai v , $|S|$ adalah jumlah seluruh sampel data, serta *Entropi(s_i)* merupakan entropi untuk sampel yang bernilai i .

B. Hyperparameter Tuning

Untuk mencapai hasil yang optimal, *hyperparameter tuning* merupakan langkah selanjutnya yang akan dilakukan. *Hyperparameter tuning* merupakan langkah terbaik yang bisa ditempuh dalam menentukan pengaturan parameter dengan mencoba banyak kemungkinan kombinasi untuk mengevaluasi hasil performansi tiap modelnya. Terdapat banyak *hyperparameter* pada algoritma *Random Forest* yang dapat dilakukan *tuning*, pada penelitian kali ini lebih berfokus kepada beberapa *hyperparameter* diantaranya "min_samples_split" (penentu berapa kali *node splitting* dilakukan), "max_depth" (kedalaman maksimum yang dimiliki oleh *tree*), "min_samples_leaf" (jumlah minimum *leaf node*), dan "max_leaf_nodes" (jumlah maksimum *tree* yang akan dibuat dengan dasar *best-first*).

Beberapa teknik yang dapat dilakukan saat *hyperparameter tuning* diantaranya yaitu *Cross Validation* (CV), *Random Search Cross Validation*, dan *Grid Search with Cross Validation*. Pemanfaatan *Grid Search with Cross Validation* (*GridSearchCV*) dilakukan pada penelitian kali ini untuk meminimalisir terjadinya *overfitting* saat melakukan *hyperparameter tuning*, dimana *GridSearchCV* melakukan evaluasi semua kombinasi yang ada.

C. Adaptive Boosting

Metode *AdaBoost* atau *Adaptive Boosting* merupakan salah satu algoritma *supervised* dalam data mining dan banyak digunakan untuk membuat model klasifikasi [25]. Metode *AdaBoost* pertama kali dikembangkan oleh Yoav Freund dan Robert Schapire [26], merupakan salah satu teknik terintegrasi untuk meningkatkan akurasi prediksi, sehingga sangat berguna untuk meningkatkan kinerja sistem dengan *decision tree* seperti pada penelitian ini.

3.6 Evaluation

Tahap *Evaluation* merupakan tahapan terakhir setelah melewati serangkaian tahap dari perancangan sistem. Evaluasi sistem dilakukan untuk mengetahui seberapa baik performansi sistem yang dihasilkan [19]. Untuk tahap ini, *Confusion Matrix* digunakan untuk mencari nilai dari beberapa point yang diperlukan untuk tahap ini seperti *accuracy*, *precision*, *recall*, dan *F1-Score*, berikut adalah formula dari point-point yang dibutuhkan:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (4)$$

$$precision = \frac{TP}{TP + FP} \quad (5)$$

$$recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1\ Score = \frac{2 \times (recall \times precision)}{(recall + precision)} \quad (7)$$

TP adalah *True Positive*, merupakan jumlah positif yang diklasifikasikan oleh sistem dengan benar. TN adalah *True Negative*, yaitu jumlah data negatif yang diklasifikasikan dengan benar oleh sistem. FN adalah *False Negative*, dengan kata lain datanya negatif tetapi salah diklasifikasikan oleh sistem. FP adalah *False Positive*, yaitu jumlah data positif yang dianggap salah oleh sistem. *Confusion matrix* membutuhkan nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) untuk menghitung poin yang tercantum dalam langkah evaluasi.

4. Evaluasi

Untuk penelitian yang dilakukan kali ini, digunakan *dataset* review film dari *imdb* sebanyak 12.000 data kemudian dibagi menjadi dua bagian dengan rasio 80:20, dengan keterangan untuk *data train* memiliki 3010 *label* negatif dan 2990 *label* positif kemudian *data test* 3049 *label* negatif serta 2951 *label* positif. Terdapat beberapa skenario pengujian yang dilakukan yang dijabarkan seperti berikut:

- i. Skenario 1 : Melihat pengaruh penggunaan jumlah dimensi *skip-gram* yang digunakan.
- ii. Skenario 2 : Melihat pengaruh penggunaan proses *stemming* pada *dataset*.

Adapun sub-skenario yang dilakukan pada tahap *Classification*, dimana sub-skenario ini dilakukan untuk mengetahui pengaplikasi metode apakah yang dapat meningkatkan hasil performansi lebih jauh, diantaranya:

- a. Sub-skenario 1 : hasil klasifikasi tidak mendapatkan perubahan apa apa (*Base Model*)
- b. Sub-skenario 2 : hasil klasifikasi dipengaruhi oleh parameter yang dirubah (*Changed Parameters*)
- c. Sub-skenario 3 : hasil klasifikasi ditingkatkan nilai akurasinya menggunakan metode *Adaptive Boosting* (*Boosted Model*).

4.1 Evaluasi Skenario 1

Pengujian skenario 1 dilakukan untuk melihat pengaruh penggunaan jumlah dimensi *skip-gram* yang digunakan, dimana pada penelitian ini digunakan dua macam dimensi *skip-gram* yaitu *skip-gram* 100 dimensi dan *skip-gram* 300 dimensi, digunakan *dataset* yang telah melewati proses *stemming* tanpa ada alasan tertentu.

Tabel 3. Hasil performansi menggunakan skip-gram 100 dimensi

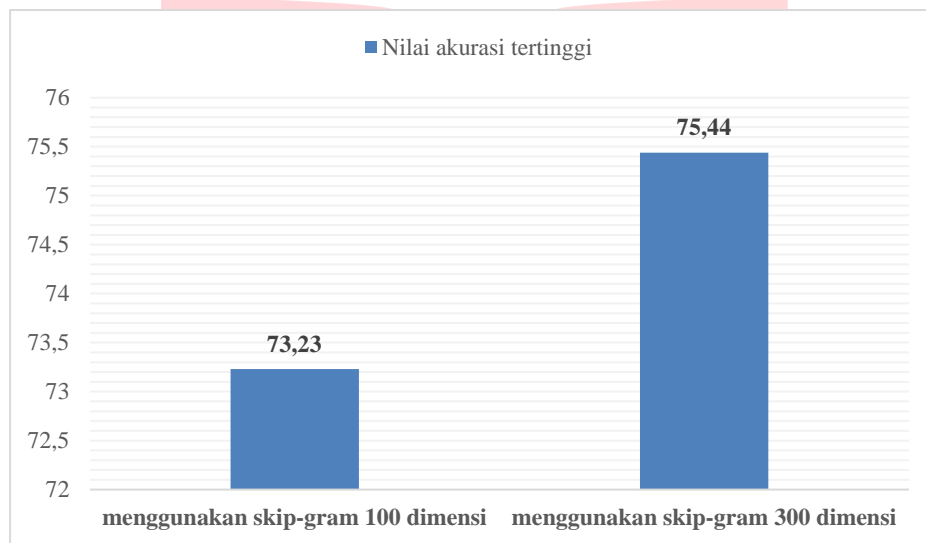
RFC	Accuracy	Precision (AVG)	Recall (AVG)	F1-Score (AVG)	Max Accuracy
Base Model	73.10%	73.11%	73.10%	73.10%	73.23%
Changed Parameters	72.31%	72.32%	72.31%	72.31%	
Boosted Model	73.23%	73.24%	73.23%	73.23%	

Berdasarkan hasil performansi diatas, didapatkan nilai akurasi tertinggi sebesar 73.23% untuk sub-skenario 3 yaitu *Boosted Model*, dimana nilai akurasi dari klasifikasi ditingkatkan lebih lanjut menggunakan metode *Adaptive Boosting* dari 73.10% pada *Base Model* menjadi 73.23%.

Tabel 4. Hasil performansi menggunakan skip-gram 300 dimensi

RFC	Accuracy	Precision (AVG)	Recall (AVG)	F1-Score (AVG)	Max Accuracy
Base Model	74.23%	74.23%	74.23%	74.23%	75.44%
Changed Parameters	74.28%	74.30%	74.28%	74.28%	
Boosted Model	75.44%	75.46%	75.44%	75.44%	

Dapat dilihat dari hasil performansi pada tabel 4, didapatkan nilai akurasi tertinggi sebesar 75.44% untuk sub-skenario 3 yaitu *Boosted Model*, dimana hasil akurasi dari klasifikasi ditingkatkan lebih lanjut menggunakan metode *Adaptive Boosting*, yang awalnya senilai 74.23% pada *Base Model* berubah menjadi 75.44%.

**Gambar 5. Diagram analisis hasil skenario 1**

Hasil evaluasi skenario 1 membuktikan bahwa penggunaan skip-gram dimensi 300 dimensi dapat mendapatkan nilai performansi yang lebih baik, namun hal tersebut tidak berarti semakin besar jumlah dimensi yang digunakan maka semakin baik hasil performansi yang didapatkan seperti pernyataan pada sub-bab 3.4 - point A - paragraf kedua di kalimat terakhir, "Pemilihan penggunaan dua dimensi tersebut didasari dari penelitian yang dilakukan oleh Mikolov [12] dimana secara garis besar disebutkan bahwa penggunaan dimensi *skip-gram* sebesar 100 dimensi dan 300 dimensi merupakan jumlah paling optimal, Mikolov tidak menjamin semakin besar jumlah dimensi memiliki performa yang terbaik."

4.2 Evaluasi Skenario 2

Pengujian skenario 2 dilakukan untuk melihat pengaruh penerapan proses *stemming* pada *dataset* yang digunakan, dimana pada penelitian ini digunakan *skip-gram* 300 dimensi demi mencapai hasil performansi terbaik.

Tabel 5. Hasil performansi dari dataset yang diterapkan stemming

RFC	Accuracy	Precision (AVG)	Recall (AVG)	F1-Score (AVG)	AVG F1-Score Max
Boosted Model	75.76%	75.78%	75.76%	75.76%	75.76%

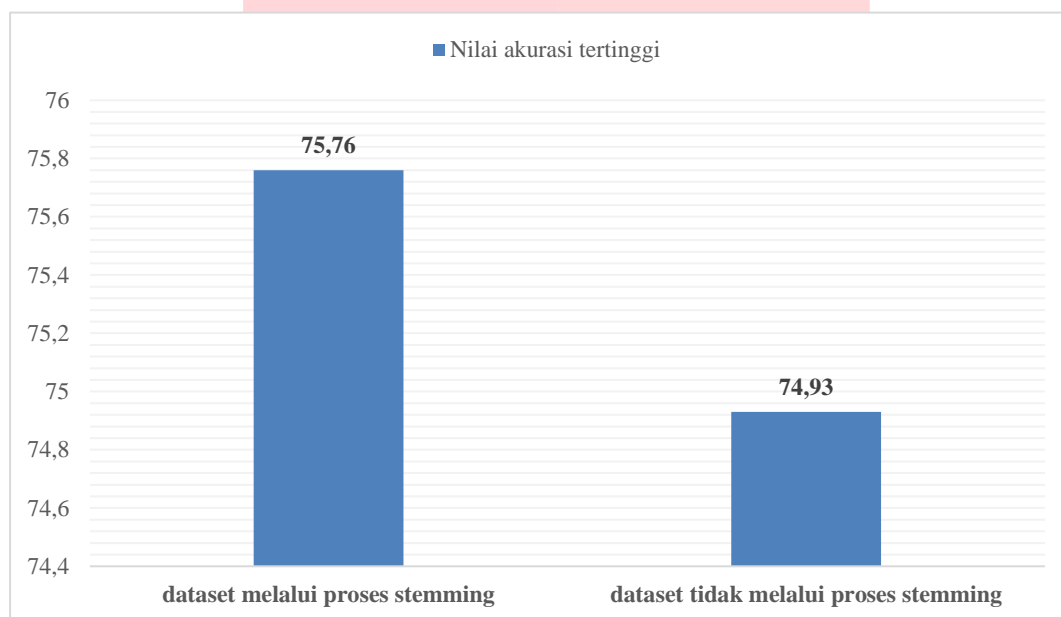
Berdasarkan hasil skenario tabel 5 diatas, dapat dilihat bahwa hasil yang digunakan berasal dari model yang telah melewati proses *boosting* / *Boosted Model*, dimana memiliki nilai akurasi tertinggi sebesar 75.76%.

Kemudian apabila diperhatikan dengan seksama, meskipun sama-sama menggunakan *skip-gram* 300 dimensi dan melalui proses *stemming*, hasil performansi berbeda dengan hasil di skenario 1 yang memiliki parameter sama. Hal ini terjadi karena proses *K-Fold Cross Validation* dimana hasil akhir performansi akan selalu berbeda setiap kali program dijalankan ulang.

Tabel 6. Hasil performansi dari dataset yang tidak diterapkan stemming

RFC	Accuracy	Precision (AVG)	Recall (AVG)	F1-Score (AVG)	AVG F1-Score Max
Boosted Model	74.93%	74.95%	74.93%	74.93%	74.93%

Tabel 6 menyajikan hasil performansi dari dataset yang tidak melalui proses *stemming*. Nilai akurasi tertinggi adalah 74.93% yang dimana apabila dibandingkan dengan nilai akurasi tertinggi dari dataset yang melalui proses *stemming* merupakan nilai yang lebih rendah. Hal tersebut membuktikan bahwa penggunaan/penerapan metode *stemming* terhadap dataset dapat meningkatkan lebih jauh hasil akhir performansi.



Gambar 6. Diagram analisis hasil skenario 2

5. Kesimpulan

Dari penelitian yang telah dilakukan, peneliti telah membangun program untuk menganalisa nilai sentimen hasil *review* film dari *website review* film IMDB menggunakan *word2vec feature extraction* dan *random forest classifier*. Terdapat dua skenario pengujian dalam penghitungan hasil performansi dimana, skenario pertama, melihat pengaruh penggunaan jumlah dimensi *skip-gram* yang digunakan, dan skenario kedua, melihat pengaruh penggunaan proses *stemming* pada dataset.

Peneliti menarik kesimpulan dari kedua skenario pengujian, pertama, banyak dimensi *skip-gram word2vec* dapat mempengaruhi hasil akhir performansi namun tidak berarti semakin besar jumlah dimensi yang digunakan maka semakin baik hasil performansi yang didapatkan. Kedua, terbukti bahwa penerapan metode *stemming* terhadap dataset dapat mempengaruhi hasil akhir performansi dimana pada kasus kali ini *stemming* meningkatkan performansi lebih jauh. Kesimpulan terakhir yang didapatkan adalah hasil akurasi tertinggi senilai 75.76% yang berasal dari skenario 2 untuk *dataset* yang menerapkan proses *stemming* serta menggunakan *skip-gram* 300 dimensi. Nilai ini menunjukkan keakurasian program dalam menentukan nilai sentimen dari *dataset* dimana pada penelitian kali ini adalah hasil *review* film dari situs IMDB.

Saran yang dapat dilakukan untuk penelitian selanjutnya adalah melakukan pengecekan terhadap list *stopword* yang digunakan untuk menghindari kata yang dapat mempengaruhi performansi, melakukan percobaan lebih dalam akan *hyperparameter tuning*, mencoba metode *boosting* lainnya, membangun sistem yang lebih efisien demi mengurangi *running time* program dan melakukan penelitian dengan metode *Modified Balanced Random Forest*.

Referensi

- [1] Murti, R. S., Adiwijaya, & Al-Faraby, S. (2019). Analisis Sentimen pada Ulasan Film Menggunakan *Word2Vec* dan *Long Short-Term Memory* (LSTM). Telkom University
- [2] Yosafat, A. R., & Kurnia, Y. (2019). Aplikasi Prediksi Rating Film dengan Perbandingan Metode *Naïve Bayes* dan KNN Berbasis *Website* Menggunakan *Framework Codeigniter*. Universitas Buddhi Dharma
- [3] Rohman, T. B., Purwanto, D. D., & Santoso, J. (2018). Sentiment Analysis Terhadap *Review* Rumah Makan di Surabaya Memanfaatkan Algoritma *Random Forest*. Sekolah Tinggi Teknik Surabaya
- [4] Yessenov, K., & Misailovic, S. (2009). *Sentiment Analysis of Movie Review Comments. Methodology*, Massachusetts Institute of Technology, 1– 17
- [5] Nurdin, A., Aji, B. A. S., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja *Word Embedding Word2Vec*, *GLOVE*, dan *Fasttext* Pada Klasifikasi Teks. Institut Teknologi Telkom Surabaya
- [6] Lilleberg, J., Zhu, Y., & Zhang, Y. (2015, July). *Support vector machines and Word2Vec for text classification with semantic features. In 2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC) (pp. 136-140)*. IEEE.
- [7] Nasukawa, T., & Yi, J. (2003). *Sentiment analysis: Capturing Favorability Using Natural Language Processing. In Proceedings of the KCAP-03, 2nd Intl. Conf. on Knowledge Capture*.
- [8] Pervan, N., & Keleş, H. Y. (2018). *Sentiment analysis using a random forest classifier on turkish web comments*. Ankara University
- [9] Mahendra, B. H., Adiwijaya, & Wisesty, U. N. (2019). Kategorisasi Berita Multi-Label Berbahasa Indonesia Menggunakan Algoritma *Random Forest*. Telkom University
- [10] Parmar, H., Bhandari, S., & Shah, G. (2014). *Sentiment Mining of Movie Reviews using Random Forest with Tuned Hyperparameters*. Marwadi Education Foundations Group of Institution
- [11] Maziida, S. R. (2018). Klasifikasi Penyakit *Diabetes Mellitus* dengan Menggunakan Perbandingan Algoritma J48 dan *Random Forest* (Studi Kasus : Rumah Sakit Muhammadiyah Lamongan). Universitas Muhammadiyah Malang
- [12] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*.
- [13] Breiman, L. (2001). *Random forests, Machine Learning, vol. 45. Issue 1, pp. 5-32*
- [14] Anwar, M. S., Subroto, I. M. I., & Mulyono, S. (2019). Sistem Pencarian *E-Journal* Menggunakan Metode *Stopword Removal* dan *Stemming* Berbasis *Android*. Universitas Islam Sultan Agung.
- [15] Septianto, H. (2016). Implementasi Metode *Vector Space Model* (VSM) Untuk Rekomendasi Nilai terhadap Jawaban *Essay*. Universitas Komputer Indonesia.
- [16] Religia, Y., Rusdi, A., Romli, I., & Mazid, A. (2019). Feature Extraction Untuk Klasifikasi Pengenalan Wajah Menggunakan Support Vector Machine dan K-Nearest Neighbor. Universitas Pelita Bangsa.
- [17] Riezka, A., Atastina, I., & Maulana, K. (2011). Analisis dan Implementasi *Data-Cleaning* Dengan Menggunakan Metode *Multi-Pass Neighborhood* (MPN). Telkom University.
- [18] Jumeilah, F. S. (2017). Penerapan *Support Vector Machine* (SVM) untuk Pengkategorian Penelitian. STMIK GI MDP
- [19] Yutika, C. H., Adiwijaya, & Al-Faraby, S. (2021). Analisis Sentimen Berbasis Aspek pada *Review Female Daily* Menggunakan TF-IDF dan *Naïve Bayes*. Telkom University
- [20] Putra, J. W. G. (2017). Pengenalan Pembelajaran Mesin dan *Deep Learning*. Tokyo Institute of Technology
- [21] Richard, K. (1997). *The 80/20 Principle, The Secret of Achieving More with Less: 285*
- [22] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems, pages 3111–3119*.
- [23] Rifda, A., Adiwijaya, & Dwifabri, M. (2021). Analisis Sentimen level Aspek pada Produk Perawatan Kulit dari Ulasan *Female Daily* berbahasa Indonesia menggunakan *Word2Vec* dan SVM. Telkom University
- [24] Purbalaksono, M.D., (2019). *Skip-Gram Negative Sample for Word Embedding in Indonesian Translation Text Classification. Doctoral dissertation*, Telkom University
- [25] Zulhanif, (2015). Algoritma AdaBoost dalam Pengklasifikasian. Universitas Muhammadiyah Surakarta
- [26] Freund, Y., & Schapire, R. E. (1997). *A decision-theoretic generalization of online learning and an application to boosting. Journal of Computer and System Sciences 55(1): 119–139*
- [27] Purnomoputra, R. B., Adiwijaya, & Wisesty, U. N. (2019). *Sentiment Analysis of Movie Reviews using Naïve Bayes Method with Gini Index Feature Selection. Journal of Data Science and its Applications 2(2)*. Telkom University
- [28] Daeli, N. O. F., & Adiwijaya. (2020). *Sentiment Analysis on Movie Reviews Using Information Gain and K-Nearest Neighbor. Journal of Data Science and its Applications 3(1)*. Telkom University

[29] Bimantara, A. A., Larasati, A., Risondang, E. M., Naf'an, M. Z., & Nugraha, N. A. S. (2019). *Sentiment Analysis of Cyberbullying on Instagram User Comments*. *Journal of Data Science and its Applications* 2(1). Telkom University

