



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang Masalah**

Dalam pengembangan piranti lunak, ada suatu istilah yang dikenal dengan *requirements engineering*. *Requirements Engineering* ini sendiri merupakan proses yang mana kebutuhan untuk produk piranti lunak dikumpulkan, didokumentasikan, dan dikelola sepanjang proses rekayasa piranti lunak (Van Lamsweerde, 2009). Menafsirkan dan mengerti tujuan, kebutuhan dan kepercayaan *stakeholders* adalah fokus dari *requirements engineering* (Van Lamsweerde, 2009).

Akhir-akhir ini, *software developer* mulai sering mengumpulkan *user feedback* dari media sosial, forum, dan *software marketplace*. Rupanya pengembang sekarang ini lebih sering melibatkan pengguna dalam proses rekayasa kebutuhan baru (Morales-Ramirez, 2017). *User feedback* dapat membantu pengembang agar dapat memahami kebutuhan, preferensi, dan masalah yang dialami dari sisi pengguna. *Software developer* harus memahami *issues* atau masalah apa saja yang timbul dari aplikasi yang dikembangkannya, contohnya bug, fitur yang tidak memuaskan, dan penambahan fitur baru secara akurat dan tepat waktu untuk pengembangan di kemudian waktu (Gao dkk., 2018). *User* yang mengalami *issue* tertentu saat menggunakan suatu aplikasi dapat membuat sebuah ulasan yang merefleksikan pengalaman user tersebut saat menggunakan aplikasi yang dibuat, kemudian *review* tersebut dapat dijadikan bahan evaluasi oleh pengembang untuk memperbaiki atau meningkatkan kualitas dari aplikasinya (Gao dkk., 2018).

Ketika lingkup aplikasi semakin besar, mengidentifikasi *issues* dalam *user feedback* merupakan sebuah tantangan yang rumit (Palomba, 2015). Melakukan analisis secara manual akan memakan banyak waktu. Satu aplikasi dapat menerima ratusan hingga ribuan *user feedback* dalam satu hari. Tentunya akan sangat melelahkan apabila dilakukan pengecekan secara manual terhadap *user feedback* tersebut. Oleh karena itu, suatu sistem otomatis analisa sentimen *user feedback* sangat dibutuhkan. Untuk menganalisa dan melakukan generalisasi opini dari *user feedback*, dapat digunakan teknik analisis sentimen.

Analisis sentimen adalah proses mempelajari opini, sentimen, emosi, penilaian, dan sikap seseorang terhadap entitas tertentu (Zhang dkk., 2018). Terdapat banyak aktivitas yang berhubungan dengan proses analisis sentimen dan bahkan sulit untuk dipisahkan karena berbagai banyak aspek, salah satunya adalah klasifikasi sentimen. Klasifikasi sentimen didasarkan pada gagasan bahwa teks mengekspresikan opini seseorang terhadap suatu entitas dan mencoba untuk memperkirakan sentimen orang tersebut terhadap entitas tersebut (Serrano-Guerrero dkk., 2015). *Machine learning* telah umum digunakan untuk klasifikasi sentimen dan memberikan hasil akurasi yang baik. Algoritma *Random Forest* cukup banyak digunakan untuk klasifikasi karena dinilai memberikan hasil akurasi yang baik.

Gupte, dkk. (2014) melakukan studi perbandingan beberapa algoritma klasifikasi dalam *machine learning*. Beberapa algoritma yang dibandingkan adalah Naive Bayes, Max Entropy, Boosted Trees, dan Random Forest. Hasil analisa dari penelitian tersebut menyatakan bahwa Random Forest memiliki performa yang

terbaik dengan tingkat simplinitas yang cukup tinggi walaupun membutuhkan spesifikasi yang sedikit lebih besar.

Vora dkk. (2017) menyatakan bahwa sekarang ini banyak orang mengekspresikan pendapat mereka dengan bahasa yang cenderung ambigu dan pilihan-pilihan kata yang rumit. Seringkali ada banyak kata yang memiliki hubungan dengan kata-kata lainnya, bahkan memiliki makna yang mirip. Untuk menemukan kemiripan makna tersebut dapat digunakan metode *word embedding*. *Word embedding* adalah jenis representasi kata yang memungkinkan kata-kata yang memiliki kemiripan arti dapat dipahami oleh algoritma *machine learning* (Spirling, 2019). Secara teknis, kata-kata yang dimasukkan akan dipetakan menjadi vektor angka menggunakan *neural network*, model probabilitas, atau reduksi dimensi pada *word co-occurrence matrix*. Menurut Khrisna (2017), ada beberapa model *word embedding* yang umum digunakan, yaitu Word2Vec (Google), Glove (Stanford), dan FastText (Facebook).

Mikolov (2013) menyatakan bahwa *word embedding* dinilai mampu mempelajari *word vector* dengan kualitas tinggi dari *dataset* yang sangat besar. Selain itu, dengan adanya *vocabulary* yang dibentuk dari *word embedding* hasil *pre-trained* model dari *word embedding* dinilai mampu mendeteksi *word similarity* baik secara semantik ataupun sintatik. Hal ini memungkinkan komputer dapat membaca kemiripan makna yang terdapat dalam kumpulan kata yang ada dalam *dataset*. Karena hal itulah *word embedding* tidak lagi memerlukan proses penghapusan imbuhan (*stemming*) dan *stop word removal* sehingga dinilai mampu

meningkatkan performa apabila dikombinasikan dengan algoritma *machine learning* lainnya.

Menurut Rehurek (2019) yang menulis dokumentasi resmi Gensim (Generate Similar), Word2Vec memberikan performa yang lebih baik untuk *semantic task*, sedangkan FastText unggul di sisi *syntactic task* apabila terdapat N-Gram pada FastText. Selain itu, FastText dapat memproses *input* kata yang tidak terdapat pada *vocabulary (Out of Vocabulary)*. Hal ini disebabkan karena FastText menjumlahkan vektor-vektor yang terdapat di yang terdapat pada N-Gram tiap karakternya (Radim Rehurek, 2019).

Vora, dkk. (2017) melakukan eksperimen melakukan klasifikasi sentimen *tweet* berbahasa Inggris menggunakan algoritma Random Forest yang dikombinasikan dengan beberapa metode *word embedding*. Metode *word embedding* yang digunakan antara lain GloVe, Word2Vec, dan FastText. Presisi tertinggi yang diperoleh adalah 91% dari lebih dari 100.000 *tweets* yang dianalisa. Hasil presisi tersebut diperoleh ketika menggunakan FastText dengan ukuran dimensi 300 yang dikombinasikan dengan algoritma Random Forest.

Ali Akbar (2017) telah mencoba memadukan *word embedding* FastText dengan beberapa algoritma klasifikasi seperti Naive Bayes, SVM, dan XGBoost untuk mendeteksi spam pada *post* Instagram berbahasa Indonesia. Dari hasil penelitian yang telah dilakukan, FastText cukup membantu meningkatkan akurasi dari masing-masing algoritma klasifikasi. Hasilnya, semua *classifier* yang dikombinasikan dengan FastText memiliki akurasi di atas 80%.

Palomba (2015) melakukan analisa terhadap *app review* untuk menyelidiki seberapa besar pengaruh *app review* terhadap pengembangan aplikasi yang dilakukan oleh *developer*. Metode yang digunakan bernama CRISTAL yang melacak dampak yang diakibatkan ulasan informatif terhadap perubahan atau pengembangan *source code* pada program. Hasil penelitian yang dilakukan menyatakan bahwa 49% *developer* mempertimbangkan ulasan dari pengguna yang informatif untuk pengembangan aplikasi lebih lanjut. Selain itu, *user review* dinilai sangat berpengaruh terhadap kesuksesan sebuah aplikasi, karena peningkatan *rating* dan komentar positif dari *user* berbanding lurus dengan pemenuhan *requirement* berdasarkan *user reviews*. Palomba (2015) menyarankan perlu dibuat sistem rekomendasi atau klasifikasi untuk menganalisa *user feedback* untuk membantu *developer* dalam pengembangan aplikasinya.

Berdasarkan penelitian-penelitian yang telah dilakukan yang telah disebutkan, penelitian ini akan mengadopsi metode versi Parth Vora, dkk. dan Ali Akbar yaitu menggunakan *word embedding* sebagai *pre-trained model* atau *embedding dataset* yang kemudian dikombinasikan dengan *classifier* lain yang dalam hal ini Random Forest Classifier. Selain itu, metode tersebut akan diterapkan dalam *dataset user feedback* yang didapatkan dari ulasan aplikasi berbahasa Indonesia untuk mengklasifikasi sentimen yang terdapat dalam *user feedback* tersebut. Dengan itu, diharapkan dapat membantu para *developer* untuk mengembangkan aplikasi atau fitur yang dibuatnya berdasarkan sentimen *user* yang diberikan lewat ulasan aplikasi dengan metode klasifikasi yang lebih modern.

Saat ini terdapat dua *platform* besar untuk memperoleh ulasan *user* aplikasi, yaitu Google PlayStore dan Apple AppStore. Dataset *user feedback* dari ulasan *user* ini dipilih untuk penelitian ini karena *user* yang memberikan ulasan merupakan pemakai aplikasi yang bersangkutan. Ketika seseorang ingin memberikan ulasan terhadap suatu aplikasi di Google Playstore maupun Apple AppStore, maka orang tersebut haruslah memasang serta membuat akun untuk aplikasi tersebut di dalam *smartphone* miliknya. Selain itu, beberapa *developer* menyetel aplikasi yang dibuatnya agar ketika *user* menggunakan aplikasi dalam jangka waktu tertentu *user* bisa memberikan *feedback* terhadap aplikasi atau fitur yang dikembangkan dalam aplikasi tersebut sehingga dapat dipastikan *user* yang memberikan ulasan adalah *user* yang sempat menggunakan aplikasi tersebut.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, rumusan masalah dalam penelitian ini adalah sebagai berikut.

1. Bagaimana menerapkan algoritma klasifikasi Random Forest yang menggunakan model *word embedding* untuk klasifikasi sentimen *user feedback*?
2. Bagaimana mengukur performa klasifikasi sentimen yang menggunakan algoritma Random Forest yang menggunakan model *word embedding* pada *user feedback*?

### 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini dapat dijabarkan menjadi beberapa poin sebagai berikut.

1. Lingkup bahasa adalah bahasa Indonesia.
2. *Dataset* yang digunakan untuk klasifikasi adalah *dataset app review* berbahasa Indonesia.

### 1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menerapkan algoritma klasifikasi Random Forest yang menggunakan model *word embedding* untuk klasifikasi sentimen *user feedback* dan mengukur performa klasifikasi sentimen yang menggunakan algoritma Random Forest yang menggunakan model *word embedding* pada *user feedback*.

### 1.5 Manfaat Penelitian

Manfaat dari penelitian ini dapat dijabarkan sebagai berikut.

1. Membantu melakukan generalisasi polaritas opini pada *user feedback* suatu aplikasi.
2. Membantu pengembang piranti lunak untuk memahami pandangan pengguna terhadap aplikasi atau fitur yang dibangun.
3. Mempercepat proses klasifikasi *user feedback* dan dilakukan secara otomatis.