BAB II
LANDASAN TEORI

2.1 Requirements Engineering

Requirements Engineeering (RE) adalah rangkaian kegiatan yang terkoordinasi untuk mengeksplorasi, mengevaluasi, mendokumentasikan, mengkonsolidasikan, merevisi dan mengadaptasi tujuan, kemampuan, kualitas, kendala dan asumsi bahwa sistem yang akan dibuat harus memenuhi berdasarkan pada masalah yang diangkat oleh sistem yang ada dan peluang yang disediakan oleh teknologi yang baru (Lamsweerde, 2009).

Menurut Lamsweerde (2009) disiplin *Requirements Engineering* (RE) memiliki interaksi utama dengan *Software Engineering* (SE). RE memberikan maanfaat bagi SE untuk merancang alat untuk mendukung kegiatannya seperti *smart editors, prototyping tools, analysers, documentation tools,* dan *configuration managers*.

Proses *requirements engineering* mencakup beberapa kegiatan yaitu, menilai apakah sistem tersebut berguna untuk bisnis (studi fisibilitas), menemukan *requirements* (pengumpulan dan analisis), mengubah *requirements* menjadi bentuk yang standar (spesifikasi), dan memeriksa bahwa *requirements* benar-benar menentukan sistem yang diinginkan *user* (validasi). Namun, di lapangan, *requirements engineering* merupakan proeses yang interatif dimana setiap aktifitas dapat saling disisipkan (*interleaved*) (Sommerville, 2010).

2.2 Analisis Sentimen

Sejak tahun 2003 analisis sentimen telah berkembang dan telah menjadi bagian dari *text mining*. Analisis sentimen adalah penelitian komputasional berdasarkan pada sentimen, emosi, pendapat, komentar dan setiap ekspresi yang diungkapkan oleh sebuah teks. Analisis sentimen bertujuan untuk mengekstraksi atribut dan komponen dari sebuah objek yang telah dikomentari dengan sentimen atau ekspresi dan untuk menentukan sentimen tersebut positf atau negatif (Liu, 2010).

Sentimen didefinisikan sebagai sebuah opini atau pandangan yang di utarakan. Sentimen juga didefinisikan sebagai emosi atau perasaan yang diutarakan melalui kata. Sentimen analisis berfokus pada klasifikasi ulasan berdasarkan polaritas. Berdasarkan klasifikasi, analisis sentimen dibagi menjadi dua kelompok utama. Kelompok pertama adalah klasifikasi ke dalam kelas opini atau fakta, atau yang dikenal dengan klasifikasi subjektivitas. Kelompok kedua adalah klasifikasi ke kelas positif atau negatif, atau yang dikenal sebagai analisis sentimen (Wijaka dkk, 2013).

2.3 Text Pre-processing

Text pre-processing adalah proses untuk mempersiapkan data mentah sebelum dilakukan proses selanjutnya. Pada umumnya, *Text pre-processing* dilakukan dengan cara mengeliminasi data yang tidak sesuai atau dengan mengubah data menjadi bentuk yang lebih mudah dipahami oleh sistem. Text pre-processing sangat penting karena dalam melakukan analisis sentimen, di dalam *post* media sosial, sebagian besar berisi kata-kata dan kalimat yang tidak bersifat formal, tidak terstruktur dan memiliki *noise* yang sangat besar (Mujilahwati, 2016).

Merujuk pada penelitian sebelumnya yang dilakukan oleh Brata dan Muslim (2018), maka pada penelitian kali ini akan dibahas beberapa tahapan *text pre-processing* yang akan digunakan antara lain *case folding, tokenizing, filtering dan stemming*. Selain itu akan ditambahkan juga satu proses dalam *text preprocessing* yaitu *stopword removal*. Definisi dari setiap tahapan *text pre-processing* yang disebutkan diatas adalah sebagai berikut.

1. *Stop word removal*, yaitu membuang kata yang tidak memiliki makna dan yang tidak berguna dalam hal pengkalsifikasian dokumen (Gurusamy & Kannan, 2014).
2. *Case folding,* pada setiap data twitter *post (tweet)* akan dilakukan proses perubahan dari huruf besar ke huruf kecil, dan menghilangkan seluruh tanda baca pada setiap kalimat.
3. *Filtering,* yaitu membuang kata-kata tidak penting dari hasil *data crawling*
4. *Stemming,* yaitu mengubah kata yang berimbuhan menjadi kata dasar pada setiap *tweet*.
5. *Tokenizing,* pada setiap data *tweet*, setiap kata akan dipisahkan berdasarkan spasi yang ditemukan.

2.4 Text Mining

Text mining adalah proses mengekstraksi pola yang menarik dan signifikan untuk mengeksplorasi pengetahuan dari sumber data yang berupa tekstual. Selain itu, *text mining* (Fan dkk, 2006). Teknik *text mining* terus sampai saat ini diterapkan dalam industri, akademisi, aplikasi web, internet dan bidang lainnya. Aplikasi seperti *Search Engine, Customer Relationship Management System,* filter untuk *email, product suggestion analysis*, pendeteksi penipuan, dan *social media analytic*,

semua aplikasi tersebut menggunakan *text mining* untuk melakukan *opinion mining*, *feature extraction*, sentimen, prediktif, dan trend analisis (He, 2013).

2.5 Data Crawling

Crawling data merupakan tahap dalam penelitian yang bertujuan untuk mengumpulkan atau mengunduh data dari suatu database (George dkk, 2014). *Crawling data* pada penelitian ini dilakukan dengan mengambil data dengan bantuan *Application Programming Interface (*API) yang disediakan oleh twitter. Untuk mengambil data dari media sosial twitter, cara mendapatkan data pada penelitian kali ini dalah dengan membuat program yang akan mengeluarkan hasil berupa twitter *post* berdasarkan kata kunci yang kita masukkan, misalkan @gojekindonesia. Apabila kata kunci sudah dimasukkan, akan keluar *post* dari *user* yang terdapat tulisan @gojekindonesia pada *post* nya.

2.6 Twitter API

Twitter API atau *Aplication Programming Interface* adalah suatu fitur yang disediakan oleh twitter untuk mempermudah seorang *developer* perangkat lunak untuk mengakses informasi yang berada di dalam twitter. Untuk dapat mengakses informasi yang disediakan oleh twitter, *developer* harus mendaftar melalui laman *https://developer.twitter.com* untuk mendapatkan consumer key, consumer access, access token dan access token secret yang akan digunakan untuk melakukan otentifikasi dari sebuah aplikasi yang akan kita bangun. Tujuan dilakukannya otentifikasi adalah untuk mendapatkan hak akses sebagai *developer* yang nanti dapat mengunduh data yang ada di twitter (Eka dkk, 2016).

2.7 Naïve Bayes Classifiers

Algoritma pengklasifikasian Naïve Bayes merupakan salah satu algoritma yang terdapat dalam teknik klasifikasi. Naïve Bayes merupakan pengklasifikasian dengan menggunakan metode probabilitas dan statistik yang ditemukan oleh ilmuwan asal Inggris Thomas Bayes, yaitu dengan cara memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya atau bisa dikenal dengan Teorema Bayes. Teorema tersebut dikombinasikan dengan asumsi *naïve*, yaitu diasumsikan bahwa kondisi antar atribut saling bebas. Persamaan dari teorema bayes adalah sebagai berikut.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \qquad \ldots (2.1)$$

B adalah data dengan kelas yang belum diketahui dan A adalah hipotesis data B yang merupakan suatu kelas yang spesifik. $P(A|B)$ merupakan probabilitas hipotesis A berdasarkan kondisi B . Lalu, $P(B|A)$ adalah probabilitas A berdasarkan kondisi pada hipotesis B. $P(A)$ merupakan probabilitas hipotesis A dan $P(B)$adalah probabilitas B (Bustami, 2014).

Terdapat dua model dalam Naïve Bayes Classifier, yaitu Multivariate Bernoulli Naïve Bayes, dan Multinomial Naïve Bayes. Model Multivariate Bernoulli Naïve Bayes adalah model berdasarkan pada binary data. Setiap token di vector fitur pada sebuah dokumen terasosiai dengan nilai 1 atau 0 (Raschka, 2014). Pada model Multinomial Naïve Bayes merupakan model berbasis frekuensi, dimana suatu dokumen direpresentasikan oleh kumpulan kata yang muncul pada dokumen tersebut (Song dkk., 2017).

2.8 Multinomial Naïve Bayes Classifiers

Multinomial Naïve Bayes merupakan turunan yang spesifik dari pengklasifikasian Naïve Bayes yang menggunakan distribusi multinomial untuk setiap fitur dari pada merujuk pada independensi bersyarat dari masing-masing fitur dalam model. Dalam metode pengklasifikasian ini, pendistribusian diperkirakan dengan menggunakan prinsip Naïve Bayes, yang mengasumsikan bahwa fitur didistribusikan secara multinomial untuk menghitung probabilitas dokumen untuk setiap label dan menjaga label memaksimalkan probabilitas (Lohar dkk, 2017).

Menurut Shimodaira (2015), tahapan pengklasifikasian text multinomial menggunakan algoritma Naïve Bayes adalah sebagai berikut.

1. Tentukan *vocabulary* (V), yaitu kumpulan kata yang menentukan dimensi dari vector fitur.
2. Hitung hal-hal berkut ini pada training set:
   a. N : yaitu jumlah seluruh dokumen.
   b. N_k : yaitu dokumen yang diklasifikasikan ke kelas k, untuk setiap kelas $k = 1, \ldots, K$.
   c. x_{it} : frekuensi kemunculan kata w_t pada dokumen D_i, untuk setiap kata pada V.
3. Hitung *likelihoods*, yaitu frekuensi kemunculan suatu kata w_t dalam semua dokumen yang termasuk dalam suatu kategori C_k dengan menggunakan *Lalpace's law of succession* untuk menghindari *zero probability problem* dimana perhitungan *likelihood* menggunakan *product* ($\prod$) dari probabilitas, jika salah satu bagian dari *product* berniai 0, maka seluruh *product* menjadi 0.

Untuk mendapatkan likelihood dengan menerapkan *Laplace's law of sucession* dapat dilihat dalam Perhitungan 2.1 atau Perhitungan 2.2.

$$P(W_t|C_k) = \frac{1+\sum_{i=1}^{N_k} x_{it}}{|V|+\sum_{S=1}^{|V|}\sum_{i=1}^{N_k} x_{it}} \quad \ldots (2.2)$$

$$P(W_t|C_k) = \frac{1+ n_k(w_t)}{|V|+\sum_{S=1}^{|V|} n_k(w_s)} \quad \ldots (2.3)$$

4. Hitung *priors*, yaitu probabilitas terklasifikasinya suatu dokumen ke dalam suatu kategori $P(C_k)$ dengan menggunakan Perhitungan 2.3.

$$P(C_k) = \frac{N_k}{N} \quad \ldots (2.4)$$

Tahap selanjutnya adalah untuk menerima dokumen yang belum diklasifikasi (D), lalu dokumen akan diklasifikasi ke kelas-kelas yang telah ditentukan sebelumnya. Dibawah ini adalah perhitungan unutuk menghitung probabilitas suatu dokumen untuk diklasifikasikan ke dalam suatu kelas, dimana x merupakan kemunculan suatu kata dalam dokumen D.

$$P(C_k|D) = a\ P(C_k) \prod_{J=1}^{|V|} P(W_t|C_k)^{x_t} \quad \ldots (2.5)$$

2.9 Confusion Matrix

Confusion matrix menurut Han dan Kamber (2011) adalah alat yang berguna untuk mengetahui seberapa baik *classifier* dapat mengenali tuple dari kelas yang berbeda. *Confusion Matrix* untuk du akelas dapat dilihat pada Tabel 2.1. Nilai *True-Positive* dan *True-Negative* memberikan informasi ketika *classifier* melakukan klasifikasi data bernilai benar, sedangkan *False-Positive* dan *False-Negative* memberikan informasi ketika *classifier* melakukan klasifikasi data bernilai salah.

		Predicted class	
		C_1	C_2
Actual class	C_1	true positives	false negatives
	C_2	false positives	true negatives

Tabel 2.1 *Confusion Matrix* Untuk Dua Kelas

Berdasrkan Tabel 2.1, dapat diketahui bahwa.

- True Positive (TP), adalah jumlah data dengan nilai sebenarnya positif dan nilai prediksi positif.
- False Positive (FP), adalah jumlah data dengan nilai sebenarnya negatif dan nilai prediksi positif.
- False Negative (FN), adalah jumlah data dengan nilai sebenarnya postif dan nilai prediksi negatif.
- True Negative (TN), adalah jumlah data dengan nilai sebenarnya negatif dan nilai prediksi negatif.

2.9.A Akurasi

Berdasarkan Brata dan Muslim (2018), Akurasi merupakan persentase dari suatu kelas terprediksi dengan benar oleh model yang sudah dibuat. Perhitungan akurasi ditunjukan pada persamaan 2.

$$Akurasi = \frac{TP + TN}{(TP + FP + FN + TN)} \times 100\% \quad \ldots (2.6)$$

2.9.B Presisi

Presisi adalah tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Perhitungan presisi ditunjukan pada persamaan 3.

$$Presisi = \frac{TP}{(TP + FP)} \times 100\% \quad \ldots (2.7)$$

2.9.C Recall

Recall adalah persentase sebuah program memprediksi sebuah data ke bukan kelas aktualnya. Perhitungan *recall* ditunjukan pada persamaan 4.

$$Recall = \frac{TP}{(TP + FN)} \times 100\% \quad \ldots (2.8)$$

2.9.D F-measure

F-measure merupakan salah satu perhitungan evaluasi dalam infomrasi yang menkombinasikan *recall* dan *precission*. Perhitungan *F-measure* ditunjukan pada persamaan 5.

$$F - measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} \times 100\% \quad \ldots (2.9)$$