BAB 2
LANDASAN TEORI

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan serta mendukung penelitian ini sehingga membuat semua proses dilakukan sistematis.

2.1 Text Mining

Text mining merupakan suatu metode untuk menggali informasi yang tidak terstruktur berbentuk teks yang biasanya berasal dari dokumen, lampiran, atau kalimat untuk mendapatkan kata dari dokumen tersebut untuk melakukan analisis yang berkaitan dengan dokumen tersebut [11], atau bisa didefinisikan sebagai suatu proses dimana informasi berkualitas dari sebuah teks digali atau bisa disebut juga sebagai *text data mining*. Informasi yang berkualitas didapatkan dari suatu pola atau teknik dengan cara yang berbeda beda yang menjadikannya sebagai suatu teknik pembelajaran yang cukup statistik. Pada *text mining*, proses dalam menata suatu *teks input* biasanya menggunakan suatu teknik parsing yang sejalan dengan penambahan fitur bahasa tertentu yang sudah dibangun dan lalu dimasukkan ke dalam database sebelum *output*-nya dievaluasi dan di interpretasikan.[12]. Ada banyak contoh metode atau teknik untuk melakukan *text mining* atau *text data mining* seperti *Scraping*, dsb.

2.1.1 Scrapping

Scrapping atau *Web Scraping* merupakan salah satu dari metode *text mining* atau *text data mining* dengan mengambil dokumen semi terstruktur dari internet yang biasanya terdiri dari halaman website atau web dalam bentuk XHTML atau HTML, lalu menganalisisnya untuk mengambil data yang dibutuhkan untuk kepentingan tertentu. Web Scraping hanya berfokus pada cara mendapatkan data dan mengekstraknya dalam ukuran apa pun [13].

2.2 Analisis Sentimen

Analisis sentimen atau *Opinion Mining*, adalah bidang studi yang menganalisis pendapat, sentimen, penilaian, sikap, dan emosi orang terhadap entitas dan atributnya yang diekspresikan dalam teks tertulis. Polaritas atau orientasi sentimen adalah kandungan dari suatu kalimat sentiment seperti positif, negative, dan netral [14].

2.2.1 Analisis Sentimen Berbasis Aspek

Analisis Sentimen Berbasis Aspek atau ABSA adalah teknik Analisis teks untuk diidentifikasi menjadi berbagai aspek dan menentukan sentimen yang sesuai untuk masing aspek didalam teks tersebut. Dengan kata lain, ABSA diperlukan untuk mengidentifikasi sentimen yang berbeda untuk beberapa aspek dalam ulasan yang sama. Misalnya pada ulasan teks terkadang memiliki 2 atau lebih aspek yang terkandung didalamnya. Analisis sentimen berdasarkan aspek bertujuan untuk mendeteksi polaritas dari teks berdasarkan aspek tertentu [15].

2.3 Text Pre-processing

Text Preprocessing merupakan sebuah proses dimana kumpulan data yang berasal dari proses *text mining* atau data mentah (*raw data*) akan diproses untuk mendapatkan suatu kalimat yang mengandung informasi-informasi penting yang bisa disebut sebagai *sample data*. Dimana, data ini tersusun atau terdiri dari kata-kata yang berkaitan dengan proses analisis yang kemudian akan digunakan untuk dilakukanya proses klasifikasi. *Text Preprocessing* diperlukan untuk mengoptimalkan kinerja dari algoritma saat melakukan proses klasifikasi [16]. Proses *Preprocessing* terdiri dari beberapa tahapan yaitu ada proses *Case Folding*, *Normalization*, *Stopword*, *Tokenizing*, dan *Stemming*.

2.3.1 Case Folding

Case folding adalah tahapan dimana raw data akan diubah semua huruf menjadi huruf kecil [11].

2.3.2 Cleansing

Cleansing adalah tahapan dimana char selain huruf seperti symbol, tanda baca, dan angka akan dibuang [17].

2.3.3 Normalization

Tahap *Normalization* atau Normalisasi, adalah tahap dimana pada proses tahapan ini, data sampel akan di sesuaikan kata katanya menjadi kata baku / dasar yang sesuai misalnya, merubah kata singkat menjadi kata utuh atau formal [18], misalnya “pdhl” menjadi “padahal”.

2.3.4 Stop Word

Tahap *Stopword*, adalah tahap dimana data sampel yang sudah dinormalisasi akan menghilangkan kata-kata yang tidak memiliki makna ataupun kata yang sering muncul dan menjadi tidak berharga atau tidak penting [11], misalnya “apa”, “kenapa”, “mungkin”, “bahwa”, dan lainnya. Ada juga kata yang memiliki makna namun sering di sebut oleh pengguna misalnya “game”, “cerita”, “grafis”, dll.

2.3.5 Tokenizing

Tahap *Tokenizing*, adalah tahap dimana data sampel yang sudah melewati proses stopword akan dipecah-pecah teks kalimatnya menjadi pecahan kata-kata.

2.3.6 Stemming

Tahap *Stemming* merupakan sebuah proses untuk menghilangkan imbuhan pada kata. proses *Stemming* menggunakan algoritma stemmer ECS (Enhanced Confix Stripping) yang dikembangkan oleh Putu Adhi Kerta Mahendra pada tahun 2008 [19] sebagai hasil evaluasi penelitian sebelumnya. ECS akan menghilangkan imbuhan yang terdapat pada suatu kata, misalnya kata “musiknya” akan dihilangkan imbuhan “-nya” dan menjadi kata “musik” dan membuang kata yang tidak memiliki kata dasar pada kamus.

2.4 Naïve Bayes

Naive Bayes yang memiliki learning efficiency yang sangat tinggi dan dapat memperkirakan semua kemungkinan hanya dengan memindai data latih. *Naive Bayesian Classifier* adalah classifier sederhana berdasarkan penerapan teorema Bayes dengan asumsi yang independensi [20] dan *Naive Bayesian Classifier* adalah algoritma klasifikasi yang mudah diimplementasikan [21][22], Serta memiliki proses klasifikasi yang cukup singkat [23]. Berikut ini adalah persamaan dari Algoritma *Naive Bayes* pada persamaan 1 dibawah ini [24].

$$P(\mathrm{D}|\mathrm{C}) = P(\mathrm{C}) \times P(\mathrm{x}_1|\mathrm{C}) \times \ldots \times P\,(\mathrm{x}_n|\mathrm{C})$$

Dimana:

$\mathrm{x}_{1\ldots}\mathrm{x}_n$: kata ke-1…kata ke-n dari data

$P(\mathrm{D}|\mathrm{C})$: Probabilitas dari dokumen D per Class

$P(\mathrm{C})$: Probabilitas dari masing-masing Class

$P(\mathrm{xn}|\mathrm{C})$: Probabilitas dari atribut ke-n per kelas pada dokumen D biasa ditulis dengan rumus = $\mathrm{Atribut}_n\,(\mathrm{C.pos}|\mathrm{C.neg})/\mathrm{Atribut}_n\,(\mathrm{C.pos}+\mathrm{C.neg})$

2.4.1 Naïve Bayes Multinomial

Multinomial Naïve Bayes atau MNB adalah varian algoritma dari Naïve Bayes yang merupakan metode supervised learning yang lebih difokuskan untuk klasifikasi teks [1]. Metode ini dirancang untuk menentukan frekuensi dari istilah atau *term*, dimana berapa kali istilah atau *term* muncul dalam sebuah dokumen [25]. Berikut persamaan dari Algoritma *Naive Bayes Multinomial* pada persamaan 2 dibawah ini [26].

$$P(\mathrm{D}|c) = \quad P(c) \times \frac{\mathrm{C}(\mathrm{x}_1|c)}{\mathrm{C}(\mathrm{y})} \times \ldots \times \frac{\mathrm{C}(\mathrm{x}_n|c)}{\mathrm{C}(\mathrm{y})}$$

Dimana:

$\mathrm{x}_{1\ldots}\mathrm{x}_n$: kata ke-1…kata ke-n dari data

$P(\mathrm{D}|c)$: Probabilitas dari dokumen D per kelas c

$P(c)$: Probabilitas dari masing-masing kelas c

$\mathrm{C}(\mathrm{x}_n|c)$: Jumlah kemunculan *term* (x) per kelas c

$\mathrm{C}(\mathrm{y})$: Jumlah seluruh kemunculan *term* per kelas c

2.4.2 Laplace Smoothing

Misalnya diberikan data untuk diklasifikasi, *Naïve Bayes* mengasumsikan atribut-atributnya bersifat independen bersyarat dan temukan $P(\mathrm{D}|c)$. jika kita berakhir dengan hasil probabilitas sama dengan nol pada salah satu atau beberapa $\mathrm{C}(\mathrm{x}_n|c)$ dimana x_n merupakan jumlah banyaknya istilah atau kata yang muncul pada dokumen. maka hal itu menyebabkan hasil probabilitas menjadi nol untuk $P(\mathrm{D}|c)$. Untuk mengatasi permasalahan ini, maka diperlukannya teknik yang bernama *Laplace Smoothing* [27]. Dimana kita menambahkan nilai 1 ke setiap hitungan sehingga nilainya tidak akan nol. Untuk menyeimbangkan hal ini, kita akan menambahkan jumlah kemungkinan kata per kelas ke dalam pembagi, sehingga pembagian tidak akan lebih besar dari 1. Sehingga persamaannya akan terlihat seperti berikut.

$$P(\mathrm{D}|c) = \quad P(c) \times \frac{\mathrm{C}(\mathrm{x}_1|c)+1}{\mathrm{C}(\mathrm{y}|c)+\mathrm{C}(\mathrm{z})} \times \ldots \times \frac{\mathrm{C}(\mathrm{x}_n|c)+1}{\mathrm{C}(\mathrm{y}|c)+\mathrm{C}(\mathrm{z})}$$

Dimana:

$\mathrm{x}_{1\ldots}\mathrm{x}_n$: kata ke-1…kata ke-n dari data

$P(\mathrm{D}|c)$: Probabilitas dari dokumen D per kelas c

$P(c)$: Probabilitas dari masing-masing kelas c

$\mathrm{C}(\mathrm{x}_n|c)$: Jumlah kemunculan *term* (x) per kelas c

$\mathrm{C}(\mathrm{y}|c)$: Jumlah seluruh kemunculan *term* per kelas c

$\mathrm{C}(\mathrm{z})$: Jumlah total *Unique term* atau istilah unik

2.5 Balancing data

Proses *balancing data* merupakan proses untuk menyeimbangkan suatu data latih yang memiliki jumlah distribusi tiap kelas yang tidak sama atau dengan kata lain salah satu kelasnya memiliki dominasi yang sangat besar jika dibandingkan dengan kelas yang lainnya, hal ini sering terjadi pada saat melakukan klasifikasi dengan metode *supervised learning*, kejadian ini biasa disebut sebagai *imbalance*

data [6]. Ada beberapa metode yang digunakan dalam mengatasi hal ini, salah satunya adalah dengan pendekatan pada tingkat data. Terdapat dua macam teknik pengambilan sampel dasar yaitu *random undersampling* (RUS) dan *random oversampling* (ROS).

2.5.1 Random Under Sampling

Random Undersampling merupakan salah satu teknik balancing data dimana data dari kelas mayoritas akan dibuang secara acak untuk bisa memenuhi keseimbangan dari data sampel. Teknik ini dapat menyebabkan adanya kekurangan data dan juga menyebabkan hilangnya informasi yang mungkin berharga [6], [28].

2.5.2 Random Over Sampling

Random Oversampling merupakan salah satu teknik balancing data dimana data dari kelas minoritas akan digandakan atau ditingkatkan secara acak sehingga data dari masing masing kelas bisa setara atau seimbang. Teknik ini dapat menyebabkan adanya kelebihan data dan duplikasi informasi yang sama persis dari data kelas minoritas [6], [29],

2.6 Alat Pendukung

Adapun beberapa tools pendukung dalam pembangunan sistem analisis sentiment berbasis aspek ini adalah sebagai berikut:

2.6.1 PHP

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman yang digunakan untuk membuat suatu website yang dinamis dan juga aplikasi web. Tidak seperti HTML yang hanya bisa menampilkan konten-konten statis, PHP bisa berinteraksi dengan database, folder dan file, sehingga membuat PHP dapat menampilkan konten yang dinamis dari sebuah website. Misalnya seperti Blog, Toko Online, CMS, Forum, dan Website Social Networking. PHP merupakan bahasa scripting, dan bukan bahasa tag-based seperti HTML. PHP termasuk dalam bahasa pemrograman yang cross-platform, yang berarti PHP bisa berjalan pada sistem operasi yang berbeda-beda seperti Windows, Linux, ataupun Mac.

Program PHP ditulis dalam file plain text (teks biasa) dan mempunyai akhiran ".php" [30], [31].

2.7 K-Fold Cross Validation

Model Validasi K-Fold Cross Validation atau K-Fold CV pertama kali di usulkan oleh Geisser [32] pada tahun 1975 untuk mengurangi waktu komputasi dari model validasi leave-one-out cross-validation (LOOCV) Oleh Stone [33] pada tahun 1974. Dimana K-Fold CV akan membagi kumpulan data menjadi K dengan ukuran yang hampir sama. Lalu, K-1 Fold akan digunakan sebagai data uji dengan Fold sisanya dijadikan sebagai data latih. Selama iterasi prosedur ini untuk K kali, masing masing K dari Fold secara berurutan dijadikan sebagai data validasi. Pada praktiknya, Nilai dari K biasanya antara 5 hingga 10 [34]. Berikut ini adalah prosedur secara umum dari K-fold:

- Data dipilih secara acak.
- Dataset di partisi menjadi K kelompok
- Pada setiap kelompoknya:
    - Kelompok ini akan menjadi dataset testing
    - Jadikan Kelompok sisanya menjadi dataset training
    - Lakukan Pelatihan pada data training dan evaluasikan di data testing
    - Simpan hasil evaluasinya
- Jumlahkan seluruh hasil data pada setiap kelompok untuk mendapatkan Rata-rata dari hasil.

Dengan kata lain, Setiap data sample dimasukan pada suatu kelompok dan akan berada didalam kelompok itu selama prosedur berlangsung. Yang berarti setiap data sampel akan menjadi data testing sebanyak 1 kali dan manjadi data training sebanyak K-1 kali.

2.8 UML

The Unified Modeling Language (UML) merupakan sebuah pemodelan visual yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan bagian bagian dari sebuah Sistem Perangkat Lunak [35]. UML adalah standar dari penulisan seperti *Blueprint* yang dimana terdapat sebuah bisnis

proses juga penulisan kelas-kelas dalam Bahasa yang spesifik [36]. Berikut ini diagram UML yang sering digunakan dalam pengembangan suatu sistem [35]:

- ***Use Case*** adalah gambaran dari fungsionalitas yang terdapat pada sistem dan dipresentasikan menjadi suatu interaksi antara aktor dan sistem dimana aktor digambarkan suatu entitas manusia atau sistem yang pelakukan pekerjaan di sistem.
- ***Activity Diagram*** adalah gambaran dari aktifitas-aktifitas yang berjalan didalam sistem
- ***Class Diagram*** adalah gambaran deskripsi dan struktur dari Package, Class, dan Object yang saling berhubungan seperti asosiasi, dsb.
- ***Sequence Diagram*** adalah interaksi yang terjadi antar objek di sekitar dan didalam sistem berupa suatu pesan berurut berdasarkan waktu.

2.9 Hasil Pengujian dan evaluasi

Hasil Pengujian merupakan hasil dari proses klasifikasi analisis sentimen menggunakan algoritma *naïve bayes multinomial* dan evaluasi dilakukan untuk menguji apakah penelitian yang dilakukan sudah berjalan sesuai tujuan penelitian atau tidak. Dimana data yang sudah terlabeli oleh *classifier* kemudian akan dihitung untuk mendapatkan tabel confussion matrix sebagai bahan untuk dilakukannya proses perhitungan akurasi, presisi, recall, f1-score.

2.7.1 Confusion Matrix

Confussion matrix merupakan suatu metode yang umumnya digunakan untuk menghitung tingkat akurasi pada data mining[23]. Pada *Confussion matrix* terdapat variable seperti True Positive (TN), False Positive (FP), True Negative (TN), dan False Negative (FN). Variabel-variabel inilah yang diperlukan untuk tahapan evaluasi berikutnya yaitu perhitungan Akurasi, Presisi, Recall, dan F1-Score.

2.7.2 Pengujian Akurasi, Precision, Recall, dan F1-Score

Pengujian akurasi dilakukan untuk melihat seberapa akuratnya classifier dalam melakukan proses klasifikasi. Berikut ini adalah perhitungan dari Akurasi, Presisi, dan Recall.

Perhitungan Akurasi:

$$\text{Accuracy} = (TP+TN) / (TP+FP+TN+FN)$$

Perhitungan Presisi:

$$\text{Precision} = (TP) / (TP+FP)$$

Perhitungan Recall:

$$\text{Recall} = (TP) / (TP+FN)$$

Perhitungan F1-Score

$$\text{F1Score} = 2 \text{ x } (\text{Precision x Recall}) / (\text{Precision} + \text{Recall})$$