BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

2.1.1. Implementasi *Opinion Mining*

Pernah dilakukan penelitian tentang opinion mining membahas tentang ekstraksi data opini publik pada perguruan tinggi. Pada penelitian tersebut dikembangkan sistem opinion mining untuk menganalisa opini publik pada perguruan tinggi. Pada subproses *document subjectivity* dan *target detection* digunakan *Part-of-Speech (POS) Tagging* menggunakan *Hidden Makov Model (HMM)*. Kemudian dari hasil *POS Tagging* tersebut diterapkan *rule* guna mengetahui dokumen tersebut termasuk opini atau bukan, serta digunakan untuk mengetahui bagian kalimat mana yang menjadi objek dari target opini. Selanjutnya dokumen yang sudah dikenali sebagai opini kemudian dilakukan klasifikasi menggunakan *Naive Bayes Clasifier(NBC)*ke dalam opini negatif atau opini positif. Ketika di uji didapatkan nilai precision dan recall untuk subproses *document subjectivity* adalah 0.99 dan 0.88, kemudian untuk subproses target *detection* adalah 0.92 dan 0.93 serta subproses opinion orientation adalah 0.95 dan 0.94 [6].

2.1.2. Klasifikasi *DeepSentiment Analysis E-Complaint*

Penelitian ini menggunakan algoritma *K-NN* untuk mengkategorikan keluhan-keluhan yang diterima oleh salah satu universitas di Indonesia

melalui sebuah fasilitas *E-Complaint* yang dimilikinya. Karena begitu banyak data masuk dan SDM yang menanganinya terbatas, maka diperlukan pengelompokan data-data tersebut sehingga mengetahui mana yang diprioritaskan terlebih dahulu. Data-data keluhan tersebut disimpan pada database yang kemudian dilakukan *prepocessing* data yang terdiri dari proses *case folding*, *tokenizing*, *filtering*, dan *stemming*. Kemudian dilakukan perhitungan pembobotan kata sehingga dapat dimasukkan ke dalam pengklasifikasian dengan metode *K-Nearest Neighbor* dan hasilnya dapat ditampilkan oleh sistem. Hasil pengujian dari penelitian ini didapatkan nilai akurasi dengan rata-rata 81,17647% menggunakan proses *stemming* dan ketika pengujian tanpa menggunakan proses *stemming* didapatkan nilai akurasi rata-rata hingga 78,82352% [5].

2.1.3. Sistem Klasifikasi dan Pencarian Jurnal

Kebutuhan konsumen akan informasi dalam bentuk jurnal semakin meningkat, sehingga pengelompokkan jurnal dibutuhkan untuk mempermudah pencarian informasi. Pengelompokkan jurnal dengan hanya mengacu pada topik tertentu sulit dilakukan jika menggunakan *query* biasa. Kemudian peneliti mencoba menggunakan sistem klasifikasi dan pencarian jurnal dengan metode *Naive Bayes* dan *Vector Space Model*dengan pendekatan *Cosine* diharapkan dapat membantu penentuan topik dan menghasilkan daftar jurnal berdasarkan urutan tingkat kemiripan.Untuk mempersiapkan kebutuhan dasar sistem dibutuhkan proses *text mining* yang

terdiri dari *text processing* dengan *parsing*, *text transformation* dengan *stemming* dan *stopword removal*, *feature selection* dan *pattern discovery*. Terdapat 5 kategori, jumlah data *training* adalah 250 jurnal dan didapatkan hasil dari *naive bayes* dengan FS-4 menghasilkan *precision* sebesar 64% dan VSM menghasilkan *recall* sebesar 54.8% dan *precission* sebesar 60.7% dengan jumlah data *training* 249 jurnal dan token unik sebanyak 3763 token [7].

2.2. Studi Literatur

Studi literatur ini berisi tentang ilmu yang akan diteliti menyangkut teori-teori yang telah ada sebelumnya dan telah digunakan oleh peneliti sebelumnya ataupun teori yang akan digunakan dalam pengembangan penelitian ini.

2.2.1. *Text Mining*

Text mining merupakan bagian dari penerapan data mining yang digunakan untuk mencari pola dalam teks. Berbeda dengan data mining yang menggunakan data biasa, dalam *text mining* ini yang digunakan adalah data dalam bentuk dokumen atau teks. *Text mining* ini bertujuan untuk menemukan pola dalam sebuah teks atau dokumen yang sebelumnya tidak terlihat menjadi pola yang diinginkan untuk tujuan tertentu [1].

2.2.2. Analisis Sentimen

Analisis sentimen merupakan bagian dari opinion mining, adalah proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi[3]. Dilakukan untuk mengetahui sikap seorang pembicara atau penulis berkaitan dengan topik yang dibahas dari komentar

yang di berikan. Komentar dari seseorang yang dipengaruhi oleh emosional penulis (*sentiment analysis*) nantinya akan dikelompokkan ke dalam kelompok sentiment baik itu positif maupun negatif.

Analisis sentimen dapat digunakan untuk melacak sebuah produk, merek maupun orang yang menentukan apakah hal tersebut dianggap sebagai hal yang positif ataupun negatif [1]. Dengan analisis sentimen ini seseorang dapat mengetahui pandangan orang lain terhadap sesuatu hal tertentu yang dapat dijadikan sebagai acuan untuk menentukan keputusannya terhadap sesuatu hal tersebut.

2.2.3. *Text Preprocessing*

Dikarenakan dokumens teks memiliki data yang tidak terstruktur maka digunakanlah *text processing* ini untuk merubah data yang belum terstruktur itu menjadi sebuah data yang terstruktur sehingga dapat siap untuk digunakan dalam proses selanjutnya. *Text Preprocessing* ini memiliki beberapa tahapan yaitu:

a. Mengekstrak teks yang akan kita olah.
b. Melakukan *stopword removal,* yaitu menghilangkan kata-kata yang tidak bermakna misalkan kata hubung.
c. Melakukan *stemming,* yaitu menghilangkan imbuhan-imbuhan dalam sebuah kata, singkatnya ialah merubah kata berimbuhan menjadi kata dasar.

2.2.4. *Term Frequency Inverse Document Frequency* (TF-IDF)

TF *(Term Frequency)* ialah frekuensi kemunculuan kata pada setiap dokumen, dari TF tersebut didapatkan DF *(document frequency)* yaitu banyaknya dokumen yang mengandung suatu kata tersebut.TF-IDF merupakan sebuah nilai yang digunakan untuk menghitung bobot sebuah kata yang muncul dalam dokumen. TF-IDF didapatkan dari hasil perkalian antar TF dan IDF, dimana IDF merupakan hasil invers dari DF. Perhitungannya dapat dituliskan sebagai berikut [7]:

$$\mathrm{IDF(w)} = \log\left(\frac{\mathrm{N}}{\mathrm{DF(w)}}\right) \quad (1)$$

$$\mathrm{TF-IDF(w,d)} = \mathrm{TF(w,d)} \times \mathrm{IDF(w)} \quad (2)$$

Keterangan:

IDF(w) : bobot kata dalam seluruh dokumen
w : sebuah kata
TF(w,d) : frekuensi kemunculan kata *w* dalam dokumen *d*
IDF(w) : inverse *DF* dari kata *w*
N : jumlah seluruh dokumen
DF(w) : jumlah dokumen yang mengandung kata *w*

Jika diperhatikan dari perhitungan *IDF* rumus nomer 1, apabila *N* = *DF(w)* maka akan didapatkan hasil 0 (nol). Untuk mensiasati hal tersebut maka dapat ditambahkan nilai 1 pada sisi *IDF*, dan perhitungan *TF (w,d)* menjadi sebagai berikut [7]:

$$\mathrm{TF-IDF(w,d)} = \mathrm{TF(w,d)} \times \left(\log\left(\frac{\mathrm{N}}{\mathrm{DF(w)}}\right) + 1\right) \quad (3)$$

Kemudian untuk menstandarisasi nilai TF-IDF ke dalam interval 0 sampai 1 diperlukan normalisasi, rumus (3) dinormalisasi dengan rumus (4) sebagai berikut [7]:

$$\mathrm{TF-IDF(w,d)} = \frac{\mathrm{TF-IDF(w,d)}}{\sqrt{\sum_{w=1}^{n} \mathrm{TF-IDF(w,d)^2}}} \quad (4)$$

Dimana $\sum_{w=1}^{n} \mathrm{TF-IDF(w,d)}$ adalah jumlah dari nilai TF-IDF dari kata pertama hingga kata ke n yang terdapat dalam dokumen d.

2.2.5. *Cosine Similarity*

Cosine similarity berfungsi untuk membandingkan kemiripan antar dokumen, dalam hal ini yang dibandingkan adalah *query* dengan dokumen latih [3]. Dalam menghitung *cosine similarity* pertama yaitu melakukan perkalian skalar antara *query* dengan dokumen kemudian dijumlahkan, setelah itu melakukan perkalian antara panjang dokumen dengan panjang *query* yang telah dikuadratkan, setelah itu di hitung akar pangkat dua. Selanjutnya hasil perkalian skalar tersebut di bagi dengan hasil perkalian panjang dokumen dan *query*. Rumus dapat dilihat sebagai berikut [7]:

$$\mathrm{cosSim}(d_j, q_k) = \frac{\sum_{i=1}^{n} (td_{ij} \times tq_{ik})}{\sqrt{\sum_{i=1}^{n} {td_{ij}}^2 \times \sum_{i=1}^{n} {tq_{ik}}^2}} \quad (5)$$

Keterangan:

cosSim(dj,qk)	: tingkat kesamaan dokumen dengan query tertentu
tdij	: term ke-i dalam vektor untuk dokumen ke-j
tqik	: term ke-i dalam vektor untuk query ke-k
n	: jumlah *term* yang unik dalam data set

2.2.6. *K-Nearest Neighbor (K-NN)*

Algoritma *k-nearest neighbor* (K-NN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut. *K-NN* termasuk algoritma *supervised learning* dimana hasil dari *query instance* yang baru diklasifikan berdasarkan mayoritas dari kategori pada *K-NN*. Kelas yang paling banyak muncullah yang akan menjadi hasil klasifikasi [4].

Tujuan dari algoritma ini adalah mengklasifikasikan obyek baru bedasarkan atribut dan *training sample*. *Classifier* tidak menggunakan model apapun untuk dicocokkan dan hanya berdasarkan pada memori. Diberikan titik *query*, akan ditemukan sejumlah *k* obyek atau (titik *training*) yang paling dekat dengan titik *query*. Klasifikasi menggunakan *voting* terbanyak diantara klasifikasi dari k obyek.. algoritma *k-nearest neighbor (K-NN)* menggunakan klasifikasi ketetanggaan sebagai nilai prediksi dari *query instance* yang baru [4].

Terdapat dua rumus dalam perhitungan *K-NN* ini, yang pertama yaitu jika nilai k=1 maka nilai maksimal dari cosSim akan menjadi hasil dari klasifikasinya. Dapat dituliskan dalam rumus sebagai berikut :

$$SIM_{max}(X) = \max_{d \in T} SIM(X, d_j) \quad (6)$$

Dimana SIMmax(X) adalah nilai kemiripan dokumen X yang paling tinggi. SIM(X,dj) adalah nilai kemiripan antara dokumen X dengan dokumen latih d. Sedangkan maxd∈T SIM(X,dj) adalah nilai maksimum

kemiripan dokumen X dengan dokumen d yang merupakan bagian dari dokumen latih T [1].

Kemudian jika digunakan k>1 maka perhitungannya adalah dengan menjumlahkan semua nilai kemiripan yang tergolong dalam satu kategori kemudian membandingkan manakah yang lebih besar

$$\mathrm{p}(x, \mathrm{c_m}) = \sum_{l=1}^{m} SIM\left(X, d_j\right) \in \mathrm{c_m} \quad (7)$$

Keterangan:

P(x,cm) : probabilitas dokumen X menjadi anggota kategori cm

sim(x,dj)∈cm yang : kemiripan antara dokumen X dengan dokumen latih dj merupakan anggota dari kategori cm

m : jumlah sim(x,dj) yang termasuk dalam kategori cm

2.2.7. *Confusion Matrix*

Confusion matrix adalah sebuah metode yang biasa digunakan untuk perhitungan akurasi pada bidang data mining. *Confusion matrix* ini nantinya akan melakukan perhitungan yang melakukan 4 keluaran, yaitu *recall* (proporsi kasus positif yang diidentifikasi dengan benar), *precision* (proporsi kasus dengan hasil positif yang benar, *accuracy* (perbandingan kasus yang diidentifikasi benar dengan jumlah seluruh kasus) dan *eror rate* (kasus yang diidentifikasi salah dengan jumlah seluruh kasus [9]. Rumus metode ini adalah sebagai berikut:

$$\text{recall} = \frac{d}{c+d}$$

$$\text{precision} = \frac{d}{(b+d)}$$

$$\text{accuracy} = \frac{(a+d)}{(a+b+c+d)}$$

$$\text{error rate} = \frac{(b+c)}{a+b+c+d}$$

Keterangan:

a= jika hasil klasifikasi (-) dan data asli (-)

b= jika hasil klasifikasi (+) dan data asli (-)

c= jika hasil klasifikasi (-) dan data asli (+)

d= jika hasil klasifikasi (+) dan data asli (+)