BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Beberapa penelitian tentang metode *text mining* telah dilakukan oleh peneliti sebelumnya. Seperti penelitian yang dilakukan oleh Bilal dkk (2015) dengan menggunakan metode *text mining* untuk klasifikasi teks dalam dalam bahasa romawi-urdu dan bahasa inggris. Penelitian tersebut menggunakan data sebanyak 300 yang diambil dari *blog* kemudian dianalisis melalui 3 algoritma klasifikasi yang berbeda, yaitu *Naive Bayes*, *Decision tree* dan KNN. Penelitian menyimpulkan bahwa klasifikasi teks menggunakan algoritma klasifikasi KNN telah menghasilkan akurasi yang tinggi mencapai 95 % (Bilal dkk., 2015).

Penelitian dengan menerapkan metode *text mining* tentang ulasan pelanggan hotel telah dilakukan sebelumnya. Penelitian Khorsand dkk (2020), dengan sumber data ulasan pengguna 64 hotel di Teheran melalui laman TripAdvisor untuk memprediksi skor ulasan pengguna baru berdasarkan informasi profil dan fasilitas hotel. Penelitian tersebut menggunakan dan membandingkan 8 algoritma klasifikasi yaitu *Naive Bayes*, *decision tree*, KNN, *logistic regression*, *neural network*, SVM, *random forest* dan *Gradient boosting*. Kesimpulan yang dihasilkan dari penelitian yaitu KNN sebagai algoritma terbaik di antara delapan metode pembelajaran mesin tersebut untuk data hotel di Teheran (Khorsand dkk., 2020).

Dalam penelitian lain Kaur dkk (2018), menerapkan metode *text mining* untuk klasifikasi sentimen komentar di twitter dalam Bahasa inggris. Penelitian tersebut menggunakan kombinasi 2 metode yaitu *N-Gram* untuk ekstrasi fitur dan algoritma KNN untuk klasifikasi data input menjadi kelas positif, negatif atau netral. Hasil dari penelitian kemudian dibandingakan dengan pengolahan data menggunakan algoritma klasifikasi SVM, dan menghasilkan kesimpulan bahwa algoritma KNN mampu menghasilkan akurasi yang lebih tinggi mencapai 86 % (Kaur dkk., 2018).

Teknik *text mining* juga digunakan untuk analisis sentimen terhadap ulasan produk di *online shop*, seperti dalam penelitian Suganya dan Vijayarani (2020).

Penelitian menggunakan data yang diambil dari situs jual beli *amazon, flipcart* dan *snapdeal* sebanyak 43 ribu ulasan dengan menerapkan teknik *web scraping*. Ulasan produk tersebut dikumpulkan kemudian dianalisis menggunakan algoritma klasifikasi KNN, SVM, *Random Forest*, CNN (*Convolutional Neural Network*) dan kombinasi SVM-CNN. Hasil penelitian menunjukkan bahwa algoritma klasifikasi KNN dapat diterapkan dengan baik untuk analisis sentimen ulasan produk di *amazon, flipcart* dan *snapdeal* dan mampu menghasilkan akurasi yang tinggi mencapai 84.4 % (Suganya dan Vijayaran., 2020).

Penelitian Shah dkk (2020), *text mining* diterapkan untuk klasifikasi teks berita pada laman bbc *news*. Penelitian tersebut menggunakan pembobotan TF-IDF, sedangkan pada tahap klasifikasi tersebut menggunakan dan membandingkan 3 algoritma yang berbeda yaitu *Logistic Regression, Random Forest* dan KNN. Data yang digunakan berjumlah 2200 teks bahasa inggris yang dikelompokkan ke dalam 5 kategori yaitu bisnis, hiburan, olahraga, teknologi dan politik. Hasil dari penelitian menunjukkan bahwa KNN merupakan pendekatan yang efektif karena mampu menghasilkan tingkat akurasi yang tinggi mencapai 92% (Shah dkk., 2020).

Dalam penelitian Padurariu dan Breaban (2019), metode *random-over sampling* diterapkan untuk mengatasi *imbalanced* dataset pada klasifikasi teks ulasan pengalaman kerja. Penelitian tersebut menggunakan 2 algoritma yang berbeda yaitu SVM dan *logistic regression*. Data yang digunakan berjumlah 4235 teks yang dikelompokkan ke dalam 17 kategori, hasil dari penelitian menunjukkan bahwa metode *random-over sampling* merupakan pendekatan yang efektif karena mampu meningkatkan hasil F1 *score* dari 45% menjadi 58% untuk pengujian menggunakan algoritma *logistic regression* dan 50% menjadi 58% untuk pengujian menggunakan algoritma SVM (Padurariu dan Breaban, 2019).

2.2. Dasar Teori

2.2.1. *Data Mining*

Data Mining adalah penambangan atau penemuan informasi baru dengan mencari pola atau aturan tertentu dari sejumlah data yang sangat besar. *Data mining* juga disebut sebagai serangkaian proses untuk menggali nilai tambah berupa

pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data. *Data mining* sendiri merupakan bagian dari proses penemuan pengetahuan yang lebih besar, yang mencakup pra-pemprosesan dan pasca-pemprosesan. Cakupan mengenai pra-pemprosesan adalah ekstraksi data, pembersihan data, penggabungan data, reduksi data, dan konstruksi fitur. Untuk pasca-pemprosesan misalnya interpretasi pola dan model, pembangkitan dan konfirmasi asumsi. *Knowledge discovery* dan *data mining* merupakan dua proses yang iteratif dan interaktif (Larose, 2005).

Dapat disimpulkan *data mining* adalah kegiatan menemukan pola yang menarik dari data dalam jumlah besar, data dapat disimpan dalam basis data, data *warehouse*, atau penyimpanan informasi lainnya. *Data mining* berkaitan dengan bidang ilmu–ilmu lain, seperti *database system*, data *warehousing*, statistik, *machine learning*, *information retrieval*, dan komputasi tingkat tinggi. Selain itu, data mining didukung oleh ilmu lain seperti *neural network*, pengenalan pola, *spatial data analysis*, *image database* dan *signal processing*. *Data mining* didefinisikan sebagai proses untuk menemukan pola-pola dalam data untuk mendapatkan informasi yang berguna dan dapat memberikan keuntungan secara ekonomi (Kotu dan Deshpande, 2015).

2.2.2. Fungsi Penerapan *Data Mining*

Data mining memiliki beberapa kegunaan fungsi untuk penerapan adalah *association, classification, clusterization, descriptive, forecasting* dan *estimation* (Larose, 2005).

1. *Assosiation* merupakan proses mengidentifikasi relasi (hubungan) dari setiap kejadian atau peristiwa yang sudah terjadi pada suatu waktu tertentu.
2. *Classification* adalah proses pengelompokkan data kedalam beberapa kategori/kelas dengan tujuan untuk mempermudah dalam mengolah dan menganalisis data.
3. *Clusterization* merupakan proses mengidentifikasi kelompok dari produk ataupun barang yang memiliki karakteristik khusus.

4. *Description* merupakan fungsi untuk tujuan memahami lebih dalam mengenai data, sehingga anda dapat mengamati setiap perubahan perilaku pada informasi tersebut.
5. *Forecasting* merupakan teknik peramalan data yang dilakukan untuk memperoleh gambaran mengenai nilai suatu data di masa yang akan datang sesuai pengumpulan informasi dengan jumlah informasi yang besar.
6. *Estimation* adalah merupakan teknik peramalan data yang dilakukan untuk memperoleh gambaran mengenai nilai suatu data di masa yang akan datang dengan variabel numerik.

2.2.3. Metode Pengembangan *Data Mining*

Data mining adalah proses yang tidak dapat dipisahkan dengan *knowledge discovery in database* (KDD), semua prosesnya adalah kegiatan yang akan di lakukan dengan proses pengambilan data pengembangan data mining (Fayyad, 1996). Berikut ini beberapa penjelasan mengenai proses pengambilan data adalah sebagai berikut.

1. *Data cleaning*, yaitu fase data masih tidak lengkap, mengandung pesan error, dan tidak konsisten. Sehingga, perlu untuk melakukan pembersihan data lebih lanjut.
2. *Data integration*, yaitu proses terjadinya integrasi data, dengan sumber data yang berulang – ulang serta dapat dikombinasikan dengan file lainnya ke dalam suatu sumber.
3. *Selection*, pada tahapan ini data yang relevan dan sesuai dengan analisis dapat dipilih pada informasi koleksi tersebut.
4. *Data transformation*, tahap data yang telah terpilih akan ditransformasikan ke dalam bentuk yang cocok untuk prosedur penggalian lebih lanjut dengan cara melakukan proses normalisasi dan agregasi.
5. *Data mining*, pada tahapan ini termasuk pada langkah – langkah utama untuk mengekstrak pola yang berpotensi sebagai sumber informasi yang berguna.

6. *Pattern evaluation,* pada tahapan ini, masuk pada pola atau skema yang menarik dengan mempresentasikan pengetahuan yang telah diidentifikasi berdasarkan hasil pengukuran (*measure*) yang telah dilakukan.
7. *Knowledge representation,* merupakan tahap yang terakhir, dengan hasil informasi berupa pengetahuan yang berhasil diperoleh akan disajikan atau divisualisasikan kepada pengguna (*user*).

2.2.4. *Text Mining*

Penambangan teks (*text mining*) adalah proses penemuan akan informasi atau *trend* baru yang sebelumnya tidak terungkap dengan memproses dan menganalisa data teks dalam jumlah besar. Jenis masukan untuk penambangan teks ini disebut data tak terstruktur dan merupakan pembeda utama dengan penambangan data yang menggunakan data terstruktur atau basis data sebagai masukan. Dalam menganalisa sebagian atau keseluruhan data tidak terstruktur tersebut, *text mining* mencoba untuk mengasosiasikan satu bagian teks dengan yang lainnya berdasarkan aturan-aturan tertentu. Hasil yang di harapkan adalah informasi baru yang tidak terungkap jelas sebelumnya. Penambangan teks dapat dianggap sebagai proses dua tahap yang diawali dengan penerapan struktur terhadap sumber data teks dan dilanjutkan dengan ekstraksi informasi dan pengetahuan yang relevan dari data teks terstruktur ini dengan menggunakan teknik dan alat yang sama dengan penambangan data (Kotu dan Deshpande, 2015).

Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen, sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur. Proses yang umum dilakukan menggunakan *text mining* antara lain yaitu pengkategorisasian teks *(text categorization),* pengelompokan *(text clustering)* dan analisis sentimen (Kotu dan Deshpande, 2015).

2.2.5. Analisis Sentimen

Analisis sentimen adalah teknik untuk menganalisis pendapat, sentimen, evaluasi, penilaian, sikap, pendapat, dan emosi terhadap suatu entitas seperti

produk, layanan, organisasi, individu, isu, peristiwa dan topik yang bersifat positif, negatif dan netral. Analisis sentimen termasuk ke dalam bidang ilmu komputer yang berkaitan dengan interaksi antara komputer dan bahasa manusia atau yang biasa disebut dengan istilah *Natural Languange Processing* (NLP) yang berisi makna dari suatu kalimat (Kontopoulos dkk., 2013). Analisis sentimen dapat dilakukan menggunakan dua pendekatan yaitu *supervised* dan *unsupervised*. Pendekatan *supervised* memiliki proses yang lebih panjang dikarenakan diharuskan melakukan labelisasi atau memberikan nilai sentimen pada setiap data latih. Pendekatan tersebut banyak digunakan oleh para peneliti yang secara tidak langsung menggambarkan bahwa pendekatan tersebut memberikan tingkat akurasi yang lebih baik. Sedangkan pendekatan *unsupervised* adalah metode klasifikasi teks yang tidak memakai label kelas pada data latih untuk menganalisa hubungan antar kedua kata. Dalam pendekatan *supervised* banyak algoritma yang dapat digunakan untuk melakukan analisis sentimen mulai dari SVM, Naive Bayes, dan KNN (Khamis dkk., 2014).

2.2.6. Prapengolahan Teks

Prapengolahan teks (*preprocessing text*) merupakan tahapan awal dalam mengolah data input sebelum memasuki proses tahapan utama dari *text mining*. Prapengolahan teks adalah suatu proses untuk menyeleksi data teks agar menjali lebih terstruktur. Beberapa tahapan dalam prapengolahan teks adalah *case folding, tokenizing/parsing, filtering, stemming* (Dharini dan Parvathi, 2019). Penjelasan empat tahapan dalam proses prapengolahan teks adalah sebagai berikut:

1. *Case folding* Merupakan tahapan yang mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf „a‟ sampai dengan „z‟ yang diterima. Karakter selain huruf akan dihilangkan dan dianggap sebagai *delimiter* (pembatas).
2. *Tokenizing/parsing* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Selain itu, spasi digunakan untuk memisahkan antar kata tersebut.

3. *Filtering* adalah tahap mengambil kata-kata penting dari hasil *tokenizing*. Proses *filtering* dapat menggunakan bantuan pustaka *stoplist* (membuang kata yang kurang penting) atau *wordlist* (menyimpan kata penting). *Stoplist/ stopword* adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam prapengolahan teks.
4. Teknik *Stemming* diperlukan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen dan untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk atau form yang berbeda karena mendapatkan imbuhan yang berbeda.

Tabel 1.1 Contoh penerapan prapengolahan teks

No	*Review*	Hasil prapengolahan
1	We were surprised to see cement encroachment at Queen of hills Mussoorie.still it is beautiful place to visit. this particular hotel is at excellent location close to mall road.you can take a quick walk on it. Food quality is excellent.	surprised cement encroachment queen hills beautiful place particular hotel excellent location close mall quick walk food quality excellent
2	The stay was overall good. But these people are not providing us the wifi access and if you want to get the wifi they will be charging us for that. Secondly no network connection. The food was good. Must visit place!	stay overall good people provide wifi access want wifi charge secondly network connection food good visit place

2.2.7. Metode *Resampling*

Teknik *resampling* adalah salah satu teknik distribusi data untuk mengurangi efek distribusi kelas tidak seimbang dalam proses pembelajaran (Jian dkk., 2016). Teknik tersebut dilakukan dengan mencoba menyeimbangkan data asli berdasarkan serangkaian algoritma sampling dengan menyesuaikan jumlah sampel dalam kelas yang berbeda. Salah satu teknik *resampling* yang umum digunakan yaitu *random over-sampling*, merupakan metode yang bertujuan untuk menyeimbangkan distribusi kelas melalui replikasi acak contoh kelas minoritas. *Random over-sampling* bertujuan untuk meningkatkan sampel kelas minoritas

sampai sama dengan kelas mayoritas lain dengan menduplikasi secara acak sampel kelas minoritas (He dkk., 2018).

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

TF-IDF (*Term Frequency Inverse Document Frequency*) adalah metode untuk mengubah suatu data teks menjadi nilai numerik sehingga data dapat diolah menggunakan algoritma klasifikasi. TF-IDF merupakan metode yang digunakan untuk menentukan nilai frekuensi sebuah kata di dalam sebuah dokumen dan juga frekuensi di dalam banyak dokumen. Perhitungan tersebut menentukan seberapa relevan sebuah kata di dalam sebuah dokumen. TF-IDF adalah sebuah algoritma yang umumnya digunakan untuk pengolahan data teks dalam jumlah besar. Algoritma TF-IDF melakukan pemberian bobot pada setiap kata kunci disetiap kategori untuk mencari kemiripan kata kunci dengan kategori yang tersedia (Shah dkk., 2020). Tahapan algoritma TF-IDF adalah sebagai berikut (Shah dkk., 2020).

1. TF (*Term Frequency*)

TF merupakan jumlah berapa banyak keberadaan suatu *term* dalam satu dokumen dan kemudian dilogaritmikan agar mengurangi besarnya bilangan, dengan logaritmik suatu bilangan akan mengurangi digit jumlah. Untuk menghitung nilai TF dapat dilakukan dengan persamaan:

$$TF_{t,d} = 1 + \log(TF_{t,d}) \quad (2.1)$$

dengan $TF_{t,d}$ merupakan banyaknya kata *t* pada dalam dokumen *d*, sebagai contoh apabila *term* terdapat dalam suatu dokumen sebanyak 5 kali maka diperoleh bobot = 1 + log (5) =1.699. Tetapi jika *term* tidak terdapat dalam dokumen tersebut, bobotnya adalah nol. Untuk menghitung nilai TF juga dapat dilakukan dengan menggunakan rumus TF biner, dengan memperhatikan apakah suatu kata atau term ada atau tidak dalam dokumen. Jika kata atau term terdapat dalam sebuah dokumen maka akan bernilai satu, jika tidak terdapat dalam dokumen maka akan bernilai nol.

2. IDF (*Inverse Document Frequency*)

IDF berfungsi mengurangi bobot suatu *term* jika kemunculannya banyak tersebar di seluruh data teks yang dipakai. Untuk menghitung nilai IDF dapat dilakukan dengan persamaan:

$$IDF = \log(\frac{d}{df}) \quad (2.2)$$

dengan *d* merupakan jumlah semua dokumen, dengan *df* merupakan jumlah dokumen yang mengandung suatu *term.* Berdasarkan persamaan TF dan IDF diatas, kemudian dapat ditentukan bobot akhir suatu *term* dengan menggunakan persamaan:

$$W_{t,d} = tf_{t,d} * idf_t \quad (2.3)$$

dengan $W_{t,d}$ merupakan nilai bobot akhir term, dengan $tf_{t,d}$ banyaknya *term* dalam dokumen, dengan idf_t merupakan pembobotan keseluruhan.

2.2.9. Algoritma K-*Nearest Neighbor*

Algoritma *k-nearest neighbor* (KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap sekumpulan data atau dokumen yang berdasarkan pembelajaran data yang sudah terklasifikasikan sebelumnya. KNN merupakan algoritma *supervised learning* dengan hasil dari *query instance* yang baru diklasifikan berdasarkan mayoritas kedekatan jarak dari kategori yang ada dalam KNN. Kelas yang paling banyak muncul nantinya kan menjadi kelas dari hasil klasifikasi. *K-Nearest Neighbor* berdasarkan konsep *"learning by analogy"*. Data *learning* dideskripsikan dengan atribut numerik n-dimensi. Tiap data *learning* merepresentasikan sebuah titik, yang ditandai dengan c, dalam ruang n-dimensi. Jika sebuah data *query* yang labelnya tidak diketahui di inputkan, maka *K-Nearest Neighbor* akan mencari k buah data learning yang jaraknya paling dekat dengan data *query* dalam ruang n-dimensi. Jarak antara data *query* dengan data *learning* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data *learning* dengan rumus *Euclidean Distance* (Bramer, 2007):

$$D_{xy} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (2.4)$$

dengan D_{xy} merupakan jarak antara objek x dan y, dengan *x* merupakan data uji yang akan diprediksi, dengan *y* merupakan data latih, dengan *n* merupakan jumlah data latih, dengan *i* merupakan data ke-*i.*

Pada proses klasifikasi data berbentuk teks tingkat kemiripan antara dua objek x dan y dihitung menggunakan *cosine similarity* dengan persamaan (Makki dkk, 2018):

$$Cos_{ij} = \frac{\sum_k (d_{ik} d_{jk})}{\sqrt{\sum_k d_{ik}^2} \sqrt{\sum_k d_{jk}^2}} \quad (2.5)$$

dengan Cos_{ij} merupakan nilai kemiripan data uji dengan data latih, dengan d_{ik} merupakan panjang vektor data uji, dengan d_{jk} merupakan panjang vektor data latih.

Tahapan algoritma klasifikasi KNN adalah sebagai berikut:

1. Menentukan parameter K (jumlah tetangga paling dekat).
2. Menghitung jarak masing-masing data latih terhadap data uji yang diberikan dengan menggunakan persamaan 2.4 atau 2.5.
3. Setelah jarak antar data latih dan uji didapatkan kemudian mengurutkan data-data tersebut ke dalam kelompok yang mempunyai jarak terkecil.
4. Dengan menggunakan kategori *Nearest Neighbor* jumlah mayoritas kelas akan dijadikan sebagai hasil klasifikasi.

Untuk menentukan nilai k terbaik dalam algortima klasifikasi KNN dengan menggunakan optimasi parameter k-*fold cross-validation* yang merupakan salah satu metode yang digunakan untuk mengetahui kinerja dari suatu model algoritma dengan cara membagi data menjadi sejumlah k, kemudian dilanjutkan dengan melakukan perulangan sesuai jumlah k, sehingga semua data akan akan berkesempatan menjadi data latih sekaligus menjadi data uji. Jumlah k yang sering dipakai dan menghasilkan akurasi optimal adalah menggunakan 10-*fold cross-validation*, sehingga data dibagi menjadi 10 bagian dengan perulangan pertama menggunakan data 1 sebagai data uji dan data 2 sampai 10 digunakan sebagai data latih. Perulangan kedua menggunakan data 2 sebagai data uji dan data lainnya digunakan sebagai data latih dan seterusnya sampai perulangan ke 10.

2.2.10. Pengukuran Kinerja Klasifikasi

Kinerja sistem klasifikasi merupakan proses untuk menggambarkan seberapa baik sistem dalam mengklasifikasikan data. Salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi adalah *confusion matrix*. Pada dasarnya *confusion matrix* mengandung informasi dengan membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi aktual. Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat istilah sebagai representasi hasil proses klasifikasi yaitu *True Positive* (TP), merupakan data yang diklasifikasikan dengan benar (Sokolova dan Lapalme, 2009). Berdasarkan nilai tersebut dapat diperoleh nilai *accuracy* yang menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi diperoleh dengan menggunakan persamaan :

$$Accuracy = \frac{TP}{\text{Jumlah data uji}} \quad (2.6)$$