BAB 2
LANDASAN TEORI

2.1 Text Mining

Text mining merupakan suatu teknik untuk mengolah dan mengesktraksi pola dari suatu data menjadi sebuah pengetahuan dan informasi yang bermanfaat. *Text mining* mengubah data yang tidak terstruktur menjadi terstruktur dalam bentuk teks. Proses yang dilakukan pada *text mining* berawal dari pengumpulan sebuah dokumen dari berbagai sumber, selanjutnya data akan melewati tahap *preprocessing* dengan memeriksa format dan karakter set, kemudian dilanjutkan dengan analisis teks untuk memperoleh sebuah informasi [7]. Hasil dari proses tersebut dapat dipergunakan untuk menggali informasi yang lebih bermanfaat, contohnya dapat dipergunakan sebagai mendapatkan ringkasan kata-kata dari sebuah dokumen.

2.2 Analisis Sentimen

Analisis sentimen atau yang dibiasa disebut dengan *Opinion Mining* merupakan suatu teknik untuk mendeteksi sebuah opini terhadap suatu subjek (produk, individu atau organisasi) dalam sebuah kumpulan data. Maksud dari opini disini adalah bentuk pengekspresian suatu sikap mengenai persoalan suatu hal. Opini dapat dengan mudah ditemui di media sosial ataupun di sebuah situs [8]. Dalam analisis sentimen terdapat tiga buah subproses, yaitu *document subjectivity*, *opinion orientation* dan *target detection* [9]. Biasanya analisis sentimen sering dimanfaatkan dalam dunia bisnis, salah satu contohnya pada bisnis tempat wisata. Analisis sentimen dimanfaatkan untuk menganalisa secara otomatis opini pengunjung tentang tempat wisata yang baru saja dikunjungi.

2.2.1 Analisis Sentimen Berdasarkan Aspek

Analisis sentimen berdasarkan aspek atau biasa disebut dengan *Aspect Based Sentiment Analysis* (ABSA) merupakan perkembangan dari analisis sentimen yang hanya mengacu pada sebuah kalimat saja. Analisis sentimen berdasarkan aspek mengacu pada entitas yang spesifik dan aspek yang dibahasnya. Analisis sentimen berdasarkan aspek bertujuan untuk mendeteksi polaritas teks tertulis berdasarkan dengan aspek tertentu.

Penelitian analisis sentimen berdasarkan aspek biasanya terdiri dari berberapa *task,* diantaranya *Aspect Extraction* dan *Aspect Sentiment Orientation Classification.* Pada *aspect extraction* berfokus pada mengekstraksi sebuah aspek dalam sebuah kalimat, sehingga sistem dapat mengetahui aspek apa saja yang terkandung dalam kalimat tersebut. Sedangkan pada *aspect sentiment orientation classification* berfokus terhadap penentuan polaritas apakah kalimat tersebut termasuk ke dalam polaritas positif atau negatif berdasarkan berbagai aspek yang terkandung.

Pada tahap mengidentifikasikan aspek terdapat beberapa pendekatan yang dapat dilakukan, diantaranya *frequency based*, *relation based*, *supervised learning*, dan *topic modelling*. Untuk penentuan sentimen terhadap aspek mempunyai dua pendekatan, yaitu *supervised learning* dan *lexicon based* [10]. Pada penelitian ini akan berfokus pada penentuan sentimen terhadap aspek saja dan pendekatan yang digunakan dalam penentuan sentimen terhadap aspek adalah *supervised learning*. Penggunaan *Supervised learning* bertujuan untuk mengklasifikasikan data uji berdasarkan pola pada data latih. Untuk mengembangkan sistem analisis sentimen berdasarkan aspek, data latih akan ditentukan aspek dan sentimennya untuk melakukan proses testing. Dataset yang sudah ditentukan aspek dan sentimennya secara manual berasal dari kalimat ulasan dari tempat wisata *Orchid Forest* pada *Google Maps.*

2.3 Google Maps

Google Maps merupakan layanan pemetaan dalam bentuk aplikasi yang dikembangkan oleh *Google*. *Google Maps* dapat dengan mudah diakses di berbagai perangkat mulai dari komputer, tablet hingga *smartphone* dengan berbagai macam sistem operasi yang berbeda. *Google Maps* memiliki tiga fitur utama yang membuat hampir seluruh dunia menggunakan aplikasi tersebut, berikut dibawah ini tiga fitur utama yang dihadirkan pada *Google Maps*.

1. Peta

   Memvisualisasikan dunia dalam bentuk digital merupakan fokus utama dari *Google Maps* sejak pertama kali dibuat. Hasil pemetaannya pun tidak main-main, cakupan petanya telah ada pada 200 negara di dunia dan setiap harinya rata-rata ada 25 juta pembaharuan yang membuat informasi lokasi yang diberikan menjadi akurat. Fitur peta saat ini tergolong sangat maju, tampilan peta dalam *Google Maps* banyak menyajikan informasi yang sangat penting, mulai dari bentuk jalan, tata letak bangunan, gangguan pada lalu lintas, kontur tanah, hingga citra langsung dari satelit. Saat ini hadirlah fitur baru yaitu *Street View* yang dapat melihat keadaan suatu tempat atau jalan dalam bentuk foto 360 derajat. Dengan hadirnya fitur tersebut, memudahkan penggunanya dalam melakukan pemetaan suatu daerah hingga melihat kondisi lingkungan tanpa harus datang ke tempat tersebut.

2. Rute

   Rute merupakan fitur yang sangat bermanfaat dan banyak digunakan di dunia. Fitur tersebut dapat membantu penggunanya dalam menentukan rute terbaik untuk pergi ke suatu tempat dengan cepat. Rute yang disajikan pun tidak hanya untuk pengguna kendaan saja, terdapat fitur untuk pejalan kaki, pengguna sepeda motor, mobil, hingga transportasi umum seperti kereta api. Dalam menentukan rute terdapat tiga fitur utama diantaranya petunjuk jalan, matriks jarak hingga jalan. Petunjuk arah akan memberikan petunjuk dalam melakukan perjalanan penggunanya, menghitung waktu perjalanan yang sedang dilakukan atau memprediksi waktu kedepannya berdasarkan kepadatan lalu lintas secara langsung. Matriks jalan memberikan beberapa

rute pilihan kepada pengguna dengan estimasi waktu sampai yang berbeda. Dan terakhir terdapat fitur jalan, yang memberikan rencana perjalanan yang tepat dengan memberikan rute yang telah dilalui.

3. Tempat

   Fitur tempat dapat memberikan informasi kepada penggunanya dalam mengetahui suatu tempat. Pengguna dapat mecari tempat yang akan dicari atau dapat melihat lingkungan sekitar tempat dimana pengguna berada. Fitur utama pada tempat diantaranya detail tempat, tempat saat ini, temukan tempat, pelengkapan otomatis, *geocoding*, geolokasi dan zona waktu. Detail tempat akan menyajikan informasi mengenai tempat yang telah dipilih, seperti nama tempat, peringkat, ulasan hingga informasi kontak tempat tersebut. Tempat saat ini memperlihatkan lokasi pengguna secara langsung. Temukan tempat dapat mempermudah pengguna dalam mencari suatu lokasi hanya dengan mencarinya lewat nama, alamat atau nomor telepon. Pelengkapan otomatis berfungsi ketika pengguna akan mencari suatu tempat pada fitur pencarian, maka secara otomatis akan memberikan saran tempat saat pengguna mengetik. *Geocoding* dapat mengubah alamat menjadi sebuah koordinat geografis atau sebaliknya. Geolokasi dapat memperlihatkan lokasi perangkat berdasarkan *Wi-Fi* atau menara seluler, dan zona waktu untuk memperlihatkan zona waktu untuk lokasi yang pengguna kunjungi [11].

Dari ketiga fitur utama pada *Google Maps*, hanya fitur tempat yang akan digunakan pada penelitian ini. Pada fitur detail tempat memberikan kemudahan dalam menyajikan informasi seputar tempat tersebut, terutama dalam memberikan penilaian dan ulasan, sehingga pengguna dapat mengetahui apakah tempat tersebut layak untuk dikunjungi atau tidak. Tidak hanya bagi pengunjung tempat saja yang mendapatkan keuntungan, pemilik tempat tersebut dapat mengetahui minat masyarakat terhadap tempat tersebut, sehingga pemilih tempat dapat memberikan pelayanan dan fasilitas terbaik pada tempat tersebut.

2.4 PHP

PHP atau biasa disebut dengan *PHP Hypertext Preprocessor* merupakan sebuah bahasa pemrograman web berbasis *server-site* yang mampu memparsing kode PHP dari kode web dengan ekstensi .php, hasil dari bahasa PHP menampilkan situs yang dinamis dari sisi pengguna. Bahasa pemrograman PHP pertama kali dikembangkan oleh seorang programmer bernama Rasmus Lerdorf pada tahun 1995, selanjutnya dikembangkan oleh kelompok independen yang disebut *Group PHP*. Kelompok tersebut yang mendefinisikan standar *de facto* untuk PHP. Dan untuk saat ini pengembangan PHP dipimpin oleh Andi Gutmans dan Zeev Suraski [12].

Bahasa pemrograman PHP termasuk ke dalam perangkat lunak bebas *(open source)* yang dirilis dibawah lisensi PHP. Sehingga banyak sekali pengguna bahasa pemrograman tersebut karena gratis dan bebas untuk dikembangkan. Dalam penggunaanya, sintaks program PHP dapat dengan mudah dibedakan dengan bahasa pemrograman web lainnya, karena sintaks programnya ditulis menggunakan apitan tanda khusus PHP. Terdapat empat macam pasangan *tag* PHP yang dapat digunakan untuk menandai *block script* PHP [13]:

1. <?php.....?>
2. <script language=”PHP”>.....</script>
3. <?.....?>
4. <%.....%>

2.5 Preprocessing

Preprocessing merupakan tahapan untuk mempersiapkan suatu data mentah menjadi data yang siap diolah pada tahap berikutnya. Data mentah yang akan dilakukan pada proses biasanya memiliki beberapa karakteristik berdimensi tinggi, terdapat noise dan sturkturnya yang tidak baik [14]. Adapun tahapan *preprocessing* yang akan dilakukan pada penelitian ini diantaranya *filtering, convert emoticon, case folding, tokenizing, spelling normalization, stopword removal* dan *stemming*.

2.5.1 *Filtering*

Filtering merupakan penghapusan karakter tertentu yang tidak dibutuhkan [15]. Biasanya karakter yang dihapus terdapat pada *special character* ASCII 33-47, 58-65, 91-96 dan 123-126. Pada Tabel 2.1 merupakan daftar *special character* ASCII yang dihapus pada proses *filtering*.

Tabel 2.1 Daftar *Special Character* ASCII Yang Dihapus Pada Proses *Filtering*

Desimal	Simbol
33	!
34	“
35	#
36	$
37	%
38	&
39	‘
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
58	:
59	;
60	<
61	=
62	>

63	?
64	@
91	[
92	\
93	]
94	^
95	_
96	`
123	{
124	|
125	}
126	~

2.5.2 *Convert Emoticon*

Emoticon merupakan sebuah bentuk pengungkapan ekspresi perasaan dalam bentuk tekstual. Hal tersebut dapat membantu dalam menentukan sentimen dalam suatu kalimat ulasan. Maka setiap *emoticon* akan dikonversikan ke dalam *string* yang bersesuaian [16]. Untuk membantu dalam mengkonversi *emoticon*, maka pada penelitian ini menggunakan *library* yang dikembangkan oleh Briquz Studio. *Library* tersebut dapat mengubah *emoticon* menjadi *unicode*, bahkan *emoticon* yang dapat dirubah telah *support* di iOS 6,7,8,9 hingga OS X [17]. Contoh perubahan *emoticon* dapat dilihat pada kutipan berikut "😄, hi", setelah masuk ke tahap convert emoticon maka kutipan tersebut akan berubah seperti ini "\ud83d\ude04, hi".

2.5.3 *Case Folding*

Case folding merupakan proses untuk menyeragamkan semua bentuk teks menjadi satu bentuk. Biasanya teks yang masuk pada tahap ini akan dirubah benjadi huruf kecil [18].

2.5.4 *Tokenizing*

Tokenizing merupakan proses untuk memisahkan sebuah teks menjadi bentuk kata, frasa, simbol atau elemen yang bermakna lain yang disebut token [19]. Pada penelitian ini lebih berfokus pada pemisahan sebuah kalimat ulasan menjadi sebuah token kata berdasarkan spasi yang ditemukan.

2.5.5 *Spelling Normalization*

Spelling normalization merupakan proses untuk identifikasi kata silang dan penulisan kata berlebihan kemudian diganti dengan kata yang sesuai pada Kamus Besar Bahasa Indonesia (KBBI). Pada proses tersebut, setiap ditemukan kata yang penggunaan huruf berlebihan dan kata yang tidak baku akan diubah menjadi kata yang benar [20]. Dalam mengidentifikasi kata silang, sistem akan menggunakan kamus yang berisikan kata baku dan tidak baku yang dikembangkan oleh Vita Anggraini [21]. Untuk memperbaiki penulisan kata yang berlebihan, sistem akan menggunakan kamus kata dasar yang dikembangkan oleh Andri Setiawan [22].

2.5.6 *Stopword Removal*

Stopword removal merupakan proses menghilangkan kata-kata umum yang tidak memiliki makna yang dibutuhkan. Pengurangan ukuran dimensi kata dalam teks dengan menghilangkan beberapa kata kerja, kata sifat dan kata keterangan lainnya dapat dimasukkan ke dalam *stopword list* [23]. Sistem akan menggunakan *stopword list* bahasa indonesia yang dikembangkan oleh Devid Haryalesmana untuk mengenali kata-kata apa saja yang termasuk dalam *stopword* [24].

2.5.7 *Stemming*

Stemming merupakan proses pengubahan sebuah kata menjadi bentuk kata dasar dengan menghilangkan imbuhan yang terdiri dari awalan, akhiran, awalan dan akhiran, dan sisipan. Dalam klasifikasi teks, *stemming* berfungsi untuk menyederhanakan kata-kata tanpa menghilangkan makna sehingga ukuran kumpulan data akan berkurang [25].

2.6 *Raw Term-Frequency*

Term Frequency (TF) merupakan metode pembobotan *heuristic* yang menentukan bobot suatu dokumen berdasarkan kemunculan *term* (istilah). Bobot sebuah dokumen ditentukan oleh banyak sedikitnya istilah yang muncul. Salah satu metode pembobotan yang akan digunakan adalah *Raw Term-Frequency*. *Raw Term-Frequency* berfungsi untuk menentukan bobot sebuah dokumen terhadap istilah yang muncul dengan menghitung frekuensi kemunculan suatu kata [26].

2.7 *Modified K-Nearest Neighbor* (MKNN)

Klasifikasi merupakan proses pengelompokkan objek yang memiliki karakteristik yang sama ke dalam beberapa kelas. Pada umumnya, klasifikasi dokumen dilakukan dengan menentukan fitur-fitur yang diwakili oleh kalimat-kalimat penting [27]. Dalam penelitian ini dalam melakukan klasifikasi teks akan menggunakan metode *Modified K-Nearest Neighbor* (MKNN).

Modified K-Nearest Neighbor (MKNN) termasuk ke dalam metode klasifikasi yang didasarkan pada kedekatan pada data latih. MKNN merupakan peningkatan peforma dan pengembangan dari metode *K-Nearest Neighbor* (KNN), dimana setiap data *sample* akan dilakukan validasi untuk mengatasi adanya data *outlier*, sehingga pembobotan di setiap datanya dapat memberikan hasil yang maksimal [28]. Metode *Modified K-Nearest Neighbor* (MKNN) terdiri dari enam tahapan utama, yaitu [29]:

1. Menentukan Nilai K

   Nilai K berfungsi untuk menentukan jumlah tetangga terdekat. Nilai K akan digunakan pada tahap menghitung nilai validitas dan penentuan kelas kata.

2. Menghitung Jarak Antar Data Latih

   Setiap antar data latih akan dihitung nilai jaraknya. Untuk mengetahui jarak antar data latih dapat menggunakan rumus *Euclidean Distance* pada Persamaan (2.1):

$$D(p,q) = \sqrt{\sum_{i=1}^{n} (\, p_i - q_i \,)^2} \qquad (2.1)$$

Keterangan:

D(p,q)	=	Jarak skalar dari dua buah vektor data *p* dan *q*
n	=	Ukuran dimensi data
p	=	Masukkan data ke-*i* dari vektor data *p*
q	=	Masukkan data ke-*i* dari vektor data *q*

Selanjutnya hasil perhitungan di atas akan diurutkan secara *ascending* dengan memilih tetangga terdekat sesuai dengan nilai K. Tetangga terdekat yang akan dipilih akan digunakan pada perhitungan nilai validitas.

3. Menghitung Nilai Validitas Data Latih

Validitas berisikan proses perhitungan jumlah titik dengan label yang sama pada semua data latih. Setiap data memiliki validitas yang bergantung pada tetangga terdekatnya. Rumus yang digunakan dalam menghitung nilai validitas pada data latih dapat menggunakan Persamaan (2.2) [30]:

$$Validitas\,(x) = \frac{1}{H} \sum_{i=1}^{H} S(lbl(x), lbl(N_i(x))) \qquad (2.2)$$

Keterangan:

Validitas(x)	=	Nilai validitas dari data latih *x*
k	=	Jumlah titik terdekat
lbl(x)	=	Kelas *x*
lbl(Ni(x))	=	Label kelas titik terdekat *x*

Variabel S digunakan untuk menghitung kesamaan antara titik a dan data ke-b pada tetangga terdekat dengan menggunakan Persamaan (2.3):

$$S(a,b) = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases} \qquad (2.3)$$

dimana a merupakan kelas a pada data latih dan b merupakan kelas selain a pada data latih.

4. Menghitung Jarak Antara Data Latih dan Data Uji

   Jarak antara data latih dan data uji diperoleh menggunakan rumus *Euclidean Distance* pada persamaan (2.1). Proses perhitungan dilakukan untuk seluruh data latih.

5. Menghitung *Weight Voting*

   Perhitungan *weight voting* merupakan proses yang dapat mengatasi kelemahan dari setiap data yang mempunyai jarak dengan *weight* yang memiliki banyak masalah dalam *outlier*. Rumus yang digunakan pada *weight voting* terdapat pada persamaan (2.4):

$$W(i) = Validitas(i) x \frac{1}{d_e + 0.5} \tag{2.4}$$

   Keterangan:

   W(i) = Nilai dari *weight voting*

   Validitas(i) = Nilai validitas data ke-*i*

   d_e = Jarak antara data latih dan data uji

6. Penentuan Kelas Kata

   Hasil perhitungan *weight voting* yang telah didapatkan akan diurutkan secara *descending*. Selanjutnya memilih bobot terbesar sesuai dengan nilai K untuk mendapatkan klasifikasi kelas.

2.8 Teknik Validasi Model

Validasi model merupakan bagian dari proses pengembangan model. Hal tersebut membantu untuk menemukan model terbaik dan mengukur seberapa baik model yang dipilih [31]. Pada penelitian ini, teknik validasi yang akan digunakan adalah *Hold-out* dengan 2 sub himpunan. Dengan teknik tersebut, sebuah dataset akan diolah secara acak dan dibagi menjadi dua bagian, satu untuk data latih dan

satu untuk data uji. Sehingga dalam pembangunan model menggunakan data latih dan menguji model menggunakan data uji.

2.9 Tahapan Pengujian

Tahapan pengujian merupakan kegiatan untuk menemukan kesalahan atau kekurangan pada sebuah sistem yang telah dibangun sehingga dapat diketahui apakah sistem tersebut telah memenuhi kriteria sesuai dengan tujuan penelitian atau tidak. Adapun metode pengujian yang digunakan pada sistem ini adalah sebagai berikut.

2.9.1 Pengujian Black Box

Pengujian *black box* merupakan teknik pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak yang akan diuji. Pengujian *black box* bekerja dengan mengabaikan struktur kontrol sehingga difokuskan pada informasi domain. Kelebihan dari pengujian *black box* adalah penguji tidak harus memiliki pengetahuan tentang bahasa pemrograman pada program yang akan diuji, pengujian dilakukan dari sudut pandang pengguna, membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan, programmer dan tester keduanya saling membutuhkan. Kekurangan dari pengujian *black box* adalah uji kasus sulit didisain tanpa spesifikasi yang jelas, kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh *programmer* dan beberapa bagian *back end* tidak diuji sama sekali [32].

2.9.2 Pengujian Akurasi

Pengujian akurasi pada penelitian ini menggunakan *confusion matrix*. *Confusion matrix* merupakan teknik yang digunakan untuk mengevaluasi model klasifikasi untuk memperkirakan hasil klasifikasi yang benar atau salah. Sebuah matriks dari prediksi akan dibandingkan dengan kelas asli yang berisi informasi aktual dan prediksi nilai klasifikasi [33]. Berikut pada Tabel 2.2 merupakan bentuk *confusion matrix*.

Tabel 2.2 *Confusion Matrix*

		Kelas Aktual	
		Iya	Tidak
Kelas Prediksi	Iya	TP	FP
	Tidak	FN	TN

Keterangan:

TP (*True Positive*) = Data positif yang terdeteksi dengan benar.

TN (*True Negative*) = Data negatif yang terdeteksi dengan benar.

FP (*False Positive*) = Data negatif namun terdeteksi sebagai data positif.

FN (*False Negative*) = Data positif namun terdeteksi sebagai data negatif.

Dalam *confusion matrix* memiliki beberapa rumus untuk mengukur evaluasi model klasifikasi, diantaranya pengukuran *accuracy* untuk mengukur tingkat pengenalan, *precision* untuk mengukur kepastian persentase data yang dilabeli positif adalah benar, recall untuk mengukur kelengkapan berapa persen data positif yang dilabeli positif. Berikut dibawah ini beberapa rumus yang digunakan dalam *confusion matrix* dari Persamaan (2.5) hingga Persamaan (2.7).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad (2.5)$$

$$Precision = \frac{TP}{TP + FP} * 100\% \quad (2.6)$$

$$Recall = \frac{TP}{TP + FN} * 100\% \quad (2.7)$$

2.10 Diagram Konteks

Diagram konteks merupakan bagan yang terdiri dari suatu proses yang menggambarkan ruang lingkup suatu sistem yang akan dibangun. Bagan tersebut berisikan siapa saja yang memberikan data masukkan ke sistem serta kepada siapa

data informasi yang harus dihasilkan sistem [34]. Diagram konteks menggunakan tiga buah simbol, yaitu simbol untuk menggambarkan *external entity*, simbol untuk menggambarkan *data flow* dan simbol untuk melambangkan *process*. Dalam menggambarkan diagram konteks hanya diperbolehkan terdiri dari satu proses saja, dan tidak menggambarkan *data store*, serta prosesnya tidak ada penomoran [35].

2.11 Data Flow Diagram (DFD)

Data Flow Diagram merupakan sebuah model yang menggambarkan dari mana asal data dan tujuan dari data tersebut, dimana data disimpan, proses apa yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan serta proses yang dilakukan pada data tersebut. *Data Flow Diagram* menggunakan empat simbol dasar yang menggambarkan entitas, alur data, proses, dan *data store* yang digunakan untuk menggambarkan pergerakan data. Diagram tersebut digunakan untuk merancang suatu sistem, baik yang telah ada ataupun sistem baru yang dikembangkan [36].

2.12 Pemrograman Terstruktur

Pemrograman terstruktur merupakan konsep pemrograman yang membagi program berdasarkan fungsi-fungsi yang dibutuhkan program komputer. Pemodelan tersebut fokus bagaimana memodelkan data dan fungsi-fungsi atau yang harus dibuat [37]. Sehingga hal tersebut akan meningkatkan produktifitas *programmer* dalam mengurangi waktu yang dibutuhkan dalam penulisan, pengujian, penelusuran kesalahan dan pemeliharaan suatu program [38].