BAB 2

TINJAUAN PUSTAKA

2.1 Tinjauan Statistik

2.1.1 Analisis Deskriptif

Analisis deskriptif yaitu bagian dari statistika mengenai pengumpulan data, penyajian, penentuan nilai-nilai statistika, pembuatan diagram atau gambar mengenai sesuatu hal (Nasution, 2017).

2.1.2 Text Mining

Text mining adalah salah satu penambangan informasi yang berguna dari data-data yang berupa tulisan, dokumen atau *text* dalam bentuk klasifikasi maupun clustering (Harjanta, 2015). *Text mining* akan memproses sekumpulan *text* pada dokumen yang berjumlah sangat besar. Untuk memproses data dengan jumlah yang besar tentu akan memakan waktu dan sumber daya yang tidak sedikit berkaitan dengan pengolahan data tersebut.

2.1.3 *Text Preprocessing*

Pada bidang *Text mining*, data *preprocessing* digunakan untuk mengekstraksi pengetahuan yang menarik dan penting serta dari data teks yang tidak terstruktur (Hermawan & Ismiati, 2020). Pada umumnya, praproses data dilakukan dengan cara mengeliminasi data yang tidak sesuai atau mengubah data menjadi bentuk yang lebih mudah diproses oleh sistem (Mujilahwati, 2016). *Text preprocessing* terdiri dari *case folding, tokenizing, filtering, dan stemming*.

2.1.3.1 *Normalization*

Normalization adalah proses untuk mengubah kata yang tidak baku menjadi baku dan mengubah singkatan menjadi kata asalnya (Khairunnisa et al., 2021).

Tabel 0.1 Contoh Ulasan Setelah Melalui Proses *Normalization*

	Normalization
Sebelum	Sy sering pakai Traveloka krn mempermudah unt transaksi perjalanan dan hotel.
Sesudah	Saya sering pakai Traveloka karena mempermudah untuk transaksi perjalanan dan hotel.

2.1.3.2 *Case Folding*

Case folding merupakan proses mengubah semua huruf dalam suatu dokumen atau kalimat menjadi huruf kecil. *Case folding* digunakan untuk mempermudah pencarian kata di dalam dokumen. Tidak semua data konsisten dalam penggunaan huruf kapital (Gunawan et al, 2018).

Tabel 0.2 Contoh Ulasan Setelah Melalui Proses *Normalization* dan *Case Folding*

	Case Folding
Sebelum	Saya sering pakai Traveloka karena mempermudah untuk transaksi perjalanan dan hotel.
Sesudah	saya sering pakai Traveloka karena mempermudah untuk transaksi perjalanan dan hotel.

2.1.3.3 *Tokenizing*

Tokenizing/tokenisasi merupakan proses pemisahan suatu rangkaian karakter berdasarkan karakter spasi, dan mungkin pada waktu yang bersamaan dilakukan juga proses penghapusan karakter tersebut, seperti tanda baca (Irmawati, 2017).

Tabel 0.3 Contoh Ulasan Setelah Melalui Proses *Normalization, Case Folding,* dan *Tokenizing*

	Tokenizing
Sebelum	saya sering pakai Traveloka karena mempermudah untuk transaksi perjalanan dan hotel.
Sesudah	['saya', 'sering', 'pakai', 'Traveloka', 'karena', 'mempermudah', 'untuk', 'transaksi', 'perjalanan', 'dan', 'hotel']

2.1.3.4 *Filtering*

Filtering dapat diartikan sebagai proses mengambil kata-kata penting dari hasil proses token atau penghapusan stopwords. *Stopwords* merupakan kosakata yang bukan merupakan ciri (kata unik) dari suatu dokumen (Ratniasih et al., 2017).

Tabel 0.4 Contoh Ulasan Setelah Melalui Proses *Normalization, Case Folding, Tokenizing,* dan *Filtering*

	Filtering
Sebelum	saya sering pakai Traveloka karena mempermudah untuk transaksi perjalanan dan hotel
Sesudah	pakai Traveloka mempermudah transaksi perjalanan hotel

2.1.3.5 *Stemming*

Stemming adalah proses untuk menggabungkan atau memecahkan setiap varian-varian suatu kata menjadi kata dasar. *Stemming* merupakan proses pemotongan imbuhan atau pengembalian kata berimbuh menjadi kata dasar (Rahman, 2017).

Tabel 0.5 Contoh Ulasan Setelah Melalui Proses *Normalization, Case Folding, Tokenizing, Filtering,* dan *Stemming*

	Stemming
Sebelum	pakai Traveloka mempermudah transaksi perjalanan hotel
Sesudah	pakai Traveloka mudah transaksi jalan hotel

2.1.4 **Seleksi Fitur (*Feature Selection*)**

Seleksi fitur (kata) merupakan tahapan proses yang berguna terutama dalam mengurangi dimensi data, menghilangkan data yang tidak relevan, serta meningkatkan hasil akurasi (Sanjaya dan Absar, 2015). Seleksi fitur mempunyai dua tujuan utama. Pertama, membuat data latih yang digunakan untuk klasifikasi lebih efisien dengan cara mengurangi ukuran kosakata yang tidak efektif. Kedua, seleksi fitur biasanya dapat meningkatkan akurasi klasifikasi dengan menghilangkan fitur noise. Seleksi fitur sendiri secara umum dibagi menjadi dua metode, yaitu:

a. *Unsupervised feature selection* adalah sebuah metode seleksi fitur yang tidak menggunakan informasi kelas dalam data latih ketika memilih fitur untuk klasifikasi. Contoh dari *Unsupervised feature selection* adalah *Term Frequency & Inverse Document Frequency*.
b. *Supervised feature selection* adalah metode seleksi fitur yang menggunakan informasi kelas dalam data latih, sehingga untuk menggunakan seleksi fitur ini harus tersedia sebuah *set pre-classied.* Contoh dari *supervised feature selection* adalah *Mutual Information, N-Gram.*

Seleksi fitur digunakan untuk memberikan karakterisik dari data. Seleksi fitur merupakan salah satu penelitian yang banyak dilakukan di berbagai bidang seperti *pattern recognition, process identification,* dan *time series modelling* (Putra, 2017).

2.1.4.1 Pembobotan Kata (*Term Weighting)*

Tahap selanjutnya yaitu pembobotan. Hal yang perlu diperhatikan dalam pencarian informasi dari dokumen yang heterogen (bervariasi) adalah pembobotan *term*. Tahap ini bertujuan untuk memberi nilai frekuensi suatu kata sebagai bobot. *Term* dapat berupa kata, frase atau unit hasil *indexing* lainnya dalam suatu dokumen yang dapat digunakan untuk mengetahui konteks dari dokumen tersebut. Karena setiap kata dalam dokumen memiliki tingkat kepentingan masing-masing, maka untuk setiap kata tersebut diberikan sebuah indikator, yaitu *term weight* (Zafikri, 2008). Menurut Zafikri (2008) *term weighting* atau pembobotan kata dipengaruhi oleh hal-hal di antaranya:

1. ***Term Frequency (TF)***

   Merupakan frekuensi kemunculan sebuah kata (*term)* dalam sebuah dokumen. Semakin besar jumlah *term* yang muncul (TF tinggi) maka semakin besar bobot dokumen atau memberikan nilai kesesuaian yang semakin besar (Informatikalogi, 2016). Menurut (Zafikri, 2008) terdapat beberapa jenis formula yang dapat digunakan pada *Term Frequency* (TF), yaitu:

   a. TF biner (*binary TF*)

      Hanya memperhatikan apakah suatu kata ada atau tidak dalam dokumen. Jika ada akan diberi nilai satu, tetapi jika tidak diberi nilai nol.

   b. TF murni (*raw TF*)

      Nilai TF yang diberikan berdasarkan jumlah kemunculan suatu kata dalam dokumen. Contohnya, jika muncul lima kali maka kata tersebut akan bernilai lima.

c. TF logaritmik

TF ini untuk menghindari dominansi dokumen yang mengandung sedikit kata dalam *query*, namun mempunyai frekuensi yang tinggi.

$$tf = 1 + \log(tf) \quad (0.1)$$

d. TF normalisasi

Menggunakan perbandingan antara frekuensi sebuah kata dengan jumlah keseluruhan kata pada dokumen.

$$tf = 0{,}5 + 0{,}5x\left(\frac{tf}{\max tf}\right) \quad (0.2)$$

2. *Inverse Document Frequency* (IDF)

Inverse Document Frequency (IDF) merupakan metode statistik numerik yang menghitung seberapa pentingnya kata dalam sebuah dokumen di mana dalam konteks ini yaitu pengurangan dominansi *term* yang sering muncul di berbagai dokumen. Metode ini digunakan sebagai bobot dalam pencarian informasi dalam *text mining* (Fanani, 2017). Hal ini diperlukan karena *term* yang banyak muncul di berbagai dokumen, dapat dianggap sebagai *term* umum, sehingga menjadi tidak penting nilainya. Sebaliknya kata yang jarang muncul dalam dokumen harus diperhatikan dalam pemberian bobot. Menurut Mandala (dalam Witten, 1999) kata yang muncul pada sedikit dokumen harus dipandang sebagai kata yang lebih penting (*uncommon terms*) daripada kata yang muncul pada banyak dokumen. Pembobotan akan memperhitungkan faktor kebalikan frekuensi dokumen yang mengandung suatu kata (*Inverse Document Frequency*). Formula IDF dapat dituliskan sebagai berikut:

$$idf_j = log\left(\frac{D}{df_j}\right) \quad (0.3)$$

Keterangan:

D : jumlah semua dokumen dalam koleksi

df_j : jumlah dokumen yang mengandung *term* t_j

Apabila nilai D sama dengan maka akan didapatkan hasil nol untuk perhitungan IDF sehingga perlu ditambahkan nilai 1 untuk rumus IDF menjadi:

$$idf_j = log\left(\frac{D}{df_j}\right) + 1 \quad (0.4)$$

Maka, rumus umum untuk TF-IDF adalah penggabungan dari formula perhitungan *raw TF* dan formula IDF dengan cara mengalikan nilai keduanya, seperti berikut:

$$w_{ij} = tf_{ij} \times idf_j \quad (0.5)$$

$$w_{ij} = tf_{jj} \times \left(log\left(\frac{D}{df_j}\right) + 1\right) \quad (0.6)$$

Keterangan:

w_{ij} : Bobot term t_j terhadap dokumen d_i

tf_{ij} : Jumlah kemunculan term t_j dalam dokumen d_i

D : Jumlah semua dokumen dalam koleksi

df_i : Jumlah dokumen yang mengandung term t_j (minimal ada satu kata yaitu term t_j).

2.1.4.2 Simulasi TF-IDF

Misalkan terdapat tiga buah dokumen yang diambil dari kelas positif sebagai berikut (Latifah, 2018):

Dokumen 1 (D1) : Traveloka tambahkan layanan pemesanan tiket bus

Dokumen 2 (D2) : Traveloka sangat disarankan buat pesan tiket

Dokumen 3 (D3) : membeli tiket pesawat murah di situs Traveloka

Ketiga dokumen tersebut dilakukan perhitungan pembobotan kata/*query* menggunakan metode TF-IDF. Misal kata/*query* yang digunakan adalah “tiket”, ”bus”, dan ”pesawat”. Ketiga dokumen tersebut dilakukan proses *preprocessing* maka akan mengalami perubahan kata seperti berikut:

Dokumen 1 : layan pesan **tiket bus**

Dokumen 2 : saran pesan **tiket**

Dokumen 3 : beli **tiket pesawat** murah situs

Berdasarkan ketiga dokumen tersebut diperoleh beberapa document term sebagai berikut:

-layan	-bus	-pesawat
-saran	-murah	-tiket
-pesan	-beli	-situs

Nilai bobot term kata “tiket” atau W(tiket) dalam dokumen 1 dapat dihitung dengan mengetahui:

1. Jumlah kata tiket dalam dokumen 1 yaitu 1, maka TF(tiket)=1.
2. Jumlah seluruh dokumen yaitu 3, maka D=3.
3. Jumlah dokumen yang memuat kata tiket yaitu 3 dokumen, maka df(tiket)=3.

Oleh karena itu, dengan menggunakan rumus pada Persamaan 2.5 diperoleh nilai bobot term untuk kata “tiket” pada dokumen 1 sebagai berikut:

$$w_{tiket} = 1 \times \left(log\left(\frac{3}{3}\right) + 1 \right)$$

$$w_{tiket} = 1$$

Bobot term kata dalam masing-masing dokumen ditunjukan oleh Tabel 2.6 berikut:

Tabel 0.6 Simulasi Perhitungan TF-IDF

Query	Dokumen			df_j	$\frac{D}{df_j}$	$IDF = log\left(\frac{D}{df_j}\right) + 1$	W		
	1	2	3				1	2	3
tiket	1	1	1	3	1	1	1	1	1
bus	1	0	0	1	3	1,48	1,48	0	0
pesawat	0	0	1	1	3	1,48	0	0	1,48
Nilai Bobot Setiap Dokumen							**2,48**	**1**	**2,48**

Nilai bobot pada dokumen menunjukkan tinggi rendahnya kesesuaian antara dokumen dengan query. Berdasarkan Tabel 2.6 diketahui bahwa dokumen yang memiliki tingkat simliaritas paling tinggi terhadap query “tiket”, “bus”, dan “pesawat”, adalah dokumen 1 dan dokumen 3, sedangkan dokumen 2 memiliki similiaritas terendah terhadap ketiga query.

2.1.5 Analisis Sentimen

Analisis sentimen merupakan proses memahami, mengekstrak dan mengolah data tekstual secara otomatis untuk mendapatkan informasi sentimen yang terkandung dalam suatu kalimat opini (Rozi, Hadi, & Achmad, 2012). Analisis sentimen pada umumnya melakukan suatu proses pengelompokan dari teks yang ada di dalam dokumen, kalimat, atau suatu pendapat sehingga menghasilkan suatu

nilai yang mengartikan apakah dokumen, kalimat, dan pendapat tersebut bernilai positif atau negatif (Fitri, Yuliani, Rosyida, & Gata, 2020).

2.1.6 Klasifikasi

Klasifikasi digunakan untuk menempatkan bagian yang tidak diketahui pada data ke dalam kelompok yang sudah ada (B, Saptono, & Anggrainingsih, 2018). Klasifikasi menggunakan variabel target dengan nilai nominal. Kalsifikasi juga menentukan hubungan antara fitur dengan variabel target.

2.1.7 *Multinomial Naïve Bayes*

Klasifikasi *Multinomial Naïve Bayes* menentukan kategori dokumen tidak hanya berdasarkan kata yang muncul pada dokumen, namun juga berdasarkan jumlah kemunculannya (Witten, et al, 2011). Manning, et al. (2009) menyatakan bahwa probabilitas dokumen d yang terletak pada kategori c memiliki persamaan:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c) \qquad (0.7)$$

Keterangan:

1. P(c), adalah *prior probability* dari sebuah dokumen yang terdapat dalam kelas c. Bila *term* dari sebuah dokumen tidak memberikan petunjuk yang jelas untuk satu kelas dibandingkan dengan kelas lainnya, maka dipilih satu kelas yang memiliki *prior probability* yang tertinggi.
2. $<t_1,t_2, ..., tn_d>$, adalah kumpulan token dalam dokumen d yang merupakan bagian dari *vocabulary* yang digunakan untuk mengklasifikasi dan n_d adalah jumlah token tersebut di dalam dokumen d. Contoh, $<t_1,t_2, ..., tn_d>$ untuk dokumen dengan satu kalimat *Beijing and Taipei join the WTO*, menjadi <*Beijing*, *Taipei*, *join*, *WTO*>, dengan $n_d = 4$, jika *term and* dan *the* dianggap

sebagai *stop words* (Manning et al, 2009). Untuk memperkirakan *prior probability* P(c) digunakan persamaan sebagai berikut:

$$P(c) = \frac{N_c}{N} \quad (0.8)$$

Keterangan:

N_c = Jumlah dari dokumen *training* dalam kelas c.

N = Jumlah keseluruhan dokumen *training* dari seluruh kelas.

Ketegori terbaik dalam klasifikasi Naïve Bayes metode Multinomial adalah kategori yang memiliki nilai maksimal atau *maximum a posteriori* (MAP) *class* C_{map}:

$$C_{map} = arg\ max\ P(c|d) \quad (0.9)$$

Persamaan dari *prior probability* dari t_k model Multinomial sama dengan model Bernoulli. *Conditional probability* akan menghitung kata atau *term* t_k pada seluruh dokumen latih pada kategori c dengan persamaan yaitu:

$$P(t_k|c) = \frac{N_k + 1}{|V| + N'} \quad (0.10)$$

|V| adalah jumlah seluruh term atau kata unik (jika, berulang, tetap dihitung 1) pada data pelatihan. N_k adalah jumlah kemunculan t_k dalam dokumen latih pada suatu kategori c dan N' adalah jumlah total *term* yang terdapat pada c dokumen latih. Penambahan angka 1 berfungsi sebagai *Laplace Smoothing* (Manning et al, 2009).

2.1.8 Evaluasi

Evaluasi sistem klasifikasi dilakukan untuk melihat hasil yang didapatkan dari klasifikasi (Asiyah, 2016). Terdapat beberapa cara untuk mengukur performa,

beberapa cara yang sering digunakan adalah dengan tabel kontingensi (*contingency table*) atau *confusion matrix* (Flach, 2012).

Confusion matrix merupakan matriks yang menampilkan prediksi klasifikasi dan klasifikasi yang actual (Alfiani Mahardhika et al, 2016). Evaluasi dilakukan dengan menggunakan *confusion matrix* yaitu *true positive rate*, *true negative rate*, *false positive rate*, dan *false negative rate* sebagai indikator. *True Positive rate* atau TP *rate* adalah persentase dari kelas positif yang berhasil diklasifikasi sebagai kelas positif, sedangkan *true negative rate* atau TN *rate* adalah presentase dari kelas negative yang berhasil diklasifikasi sebagai kelas negatif. *False positive rate* atau FP *rate* adalah kelas negatif yang diklasifikasi sebagai kelas positif. *False negative rate* atau FN *rate* adalah kelas positif yang diklasifikasi sebagai kelas negatif (Herlinawati et al., 2020).

Tabel 0.7 *Confusion Matrix*

		Actual Class	
		Sentimen Positif	**Sentimen Negatif**
Predicted Class	**Sentimen Positif**	*True Positive (TP)*	*False Positive (FP)*
	Sentimen Negatif	*False Negative (FN)*	*True Negative (TN)*

Kinerja klasifikasi bisa dievaluasi dengan memperhatikan ukuran-ukuran sebagai berikut:

1. Akurasi

   Akurasi adalah tingkat kedekatan antara nilai prediksi dengan nilai aktual yang dihitung dari persentase prediksi yang benar dari total keseluruhan jumlah prediksi positif. Berikut persamaannya:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (0.11)$$

2. Presisi

   Presisi atau confidence adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data yang sebenarnya. Berikut persamaannya:

$$Precision = \frac{TP}{TP + FP} \quad (0.12)$$

3. Recall

   Recall atau sensitivitas adalah proporsi kasus positif yang sebenarnya diprediksi positif secara benar. Berikut persamaannya:

$$Recall = \frac{TP}{TP + FN} \quad (0.13)$$

2.2 Tinjauan Non Statistik

Traveloka merupakan bisnis digital yang menyediakan layanan perjalanan atau travel. Bisnis digital ini didirikan pada tahun 2012 oleh Ferry Unardi, Derianto Kusuma, dan Albert. Mereka memulai bisnis digital ini karena melihat dan merasakan sulitnya mendapatkan tiket perjalanan pesawat dan kereta. Aplikasi Traveloka terus-menerus mengalami perkembangan baik dari teknologinya maupun dari *user interface*nya (Latuheru & Irwansyah, 2018).

Pada tahun 2013 Traveloka berubah menjadi situs reservasi yang konsentrasi dalam pemesanan tiket pesawat. Pada tahun 2014 Traveloka masuk ke bisnis reservasi dan menjadi situs pemesanan hotel. Dilansir dari situs Traveloka, pada tahun 2017 sudah menyediakan pemesanan tiket kereta api, tiket perjalanan wisata, dan paket wisata dengan memberikan pelayanan terbaik untuk konsumen. Traveloka telah melakukan banyak inovasi dengan menghadirkan beragam fitur

pada pelayanannya agar kebutuhan konsumennya selalu terpenuhi. Traveloka juga meraih berbagai prestasi dengan meraih penghargaan, yaitu "*Most Innovative Brand*" atas perannya dalam melahirkan beragam inovasi dan memberikan kemudahan kepada konsumen melalui teknologi. Traveloka juga mendapatkan gelar "Unicorn" yang artinya nilai valuasi perusahaan tersebut lebih dari USD 1 miliar dan Traveloka merupakan perusahaan *Online Travel Agent* lokal pertama yang meraih gelar tersebut.

2.2.1 *Google Play*

Google Play adalah layanan distribusi digital yang dioperasikan dan dikembangkan oleh Google. Google Play berfungsi sebagai toko aplikasi resmi untuk sistem operasi pada Android yang memungkinkan penggunanya untuk menelusuri dan mengunduh aplikasi yang diterbitkan melalui Google. *Google Play* juga berfungsi sebagai toko media digital yang menawarkan produk-produk seperti musik/lagu, buku, aplikasi, permainan, ataupun pemutar media berbasis cloud (Yosmita Praptiwi, 2018).

2.2.2 *Online Review*

Kehadiran fitur *online review* menjadi informasi tambahan yang dapat memengaruhi asumsi dan keputusan konsumen terkait penjualan atau produk yang bersangkutan (Agustina & Fayardi, 2019).

2.2.3 *Word cloud*

Word cloud merupakan salah satu metode visualisasi dokumen teks yang sering digunakan. *Word cloud* merupakan representasi grafis dari sebuah dokumen

yang dilakukan dengan plotting kata-kata yang sering mencul pada sebuah dokumen pada ruang dua dimensi. Frekuensi dari kata yang sering muncul ditunjukkan melalui ukuran huruf kata tersebut. Semakin besar ukuran kata menunjukkan semakin besar frekuensi kata tersebut muncul dalam dokumen. Berikut merupakan contoh dari visualisasi dokumen teks dengan *Word cloud* (Castella & Sutton, 2014).

Gambar 0.1 Contoh Visualisasi Data dengan *Word cloud*