BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Penelitian

Penelitian sebelumnya mengenai analisis sentimen dengan menggunakan metode *Naive Bayes* telah banyak dilakukan, diantaranya adalah penelitian menggunakan data Twitter berjudul "Analisis Sentimen Tentang Opini Film pada Dokumen Twitter Berbahasa Indonesia Menggunakan *Naive Bayes* dengan Perbaikan Kata Tidak Baku" oleh (Antinasari et al., 2017). Penelitian ini menganalisis opini Twitter dan membaginya menjadi dua kelas yaitu positif dan negatif yang menghasilkan akurasi hingga 91.67%.

Selanjutnya, terdapat penelitian analisis sentimen pada *review* aplikasi *mobile* yang dilakukan oleh Firmansyah et al. (2016) berjudul "Sentiment Analysis Pada Review Aplikasi Mobile Menggunakan Metode *Naïve Bayes* Dan *Query Expansion*" hasil penelitian ini menghasilkan akurasi 95% dan 98% jika tanpa *Query Expansion*.

Penelitian mengenai analisis sentimen dengan *Naive Bayes* juga diterapkan pada *Big Data* berjudul "*Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier*" oleh (Liu et al., 2013). Hasil penelitian ini, *Naive Bayes* dapat berjalan dengan baik pada *Big Data* dan menghasilkan akurasi sebesar 82%.

Pada penelitian ini, metode seleksi fitur yang dipilih adalah *Query Expansion Ranking* yang pernah diterapkan dalam penelitian Parlar dan Özel (2016) berjudul "*A New Feature Selection Method for Sentiment Analysis of Turkish Reviews*". Penelitian ini menerapakan seleksi fitur *Query Expansion Ranking* untuk analisis sentimen empat kategori. Setelah itu, metode *Query Expansion Ranking* dibandingkan dengan metode seleksi fitur lain yaitu *Chi-Square* dan *Document Frequency Difference,* hasilnya *Query Expansion Ranking* menghasilkan akurasi terbaik dengan akurasi tertinggi sebesar 91%.

2.2 Pariwisata

Berdasarkan UU No. 10 Tahun 2009 tentang Pariwisata oleh Kementrian Pariwisata, "Wisata adalah kegiatan perjalanan yang dilakukan oleh seseorang atau sekelompok orang dengan mengunjungi tempat tertentu untuk tujuan rekreasi, pengembangan pribadi, atau mempelajari keunikan daya tarik wisata yang dikunjungi dalam jangka waktu sementara" (Kemenpar, 2009).

Sedangkan wisatawan adalah "orang yang melakukan pariwisata. Dan Pariwisata adalah berbagai macam kegiatan wisata dan didukung berbagai fasilitas serta layanan yang disediakan oleh masyarakat, pengusaha, Pemerintah, dan Pemerintah Daerah" (Kemenpar, 2009).

2.3 TripAdvisor®

TripAdvisor® adalah sebuah situs *website* bertema pariwisata yang terbesar di dunia, bertujuan untuk membantu penggunanya untuk menikmati perjalanan dengan lebih baik, mulai dari informasi hingga pemesanan hotel dengan harga

paling murah. TripAdvisor® mempunyai pengunjung sebanyak 390 juta setiap bulannya serta 435 juta ulasan dan opini mengenai restoran, akomodasi, dan objek wisata. TripAdvisor® juga beroperasi di 49 pasar di seluruh dunia (TripAdvisor Inc, 2017).

2.4 Pemrosesan Teks

Pemrosesan Teks atau *Text Mining* adalah proses pengambilan informasi yang biasanya bersifat baru dan berguna untuk pemecahan masalah dari banyak dokumen yang bersifat semi terstruktur dan tidak terstruktur. Pemrosesan teks mengubah kata dan frasa dalam dokumen menjadi data numerik yang berhubungan dengan data pada *database* menggunakan metode-metode *data mining* (Vijayarani dan Janani, 2016) .

Walaupun metode dan proses yang digunakan oleh *data mining* dan pemrosesan teks sama, namun metode data mining hanya bisa digunakan oleh data terstruktur yaitu data yang sudah disusun secara tetap dalam baris dan kolom dalam *database.* Sedangkan pemrosesan teks mampu mengolah data yang bersifat tidak terstruktur yaitu kebalikan dari data terstruktur, data yang tidak ditetapkan dalam baris dan kolom dalam *database* serta terdiri dari banyak tipe seperti *integer, float, decimal, char,* dll (Oracle, 2017)*,* dan data semi terstruktur yaitu data diantara terstruktur dana tidak terstruktur.

Pemrosesan teks adalah area baru dalam bidang ilmu komputer yang mencoba untuk menyelesaikan masalah seputar *data mining, machine learning, information extraction,* pemrosesan bahasa alami*,* sistem temu kembali informasi*, knowledge management,* dan klasifikasi.

2.5 Preprocessing

Preprocessing adalah proses penting dalam pemrosesan teks, pemrosesan bahasa alami*,* dan sistem temu kembali informasi yang berguna untuk mengekstrak pengetahuan yang bersifat menarik dan tidak biasa dari data yang tidak terstuktur (Gurusamy dan Kannan, 2014). *Preprocessing* digunakan sebelum data siap diolah untuk pemrosesan teks dengan tujuan mengurangi ukuran data sehingga keefektifan data dapat meningkat. Metode yang digunakan dalam *preprocessing* ini biasanya terbagi menjadi tiga tahap yaitu tokenisasi, *filtering,* dan *stemming.*

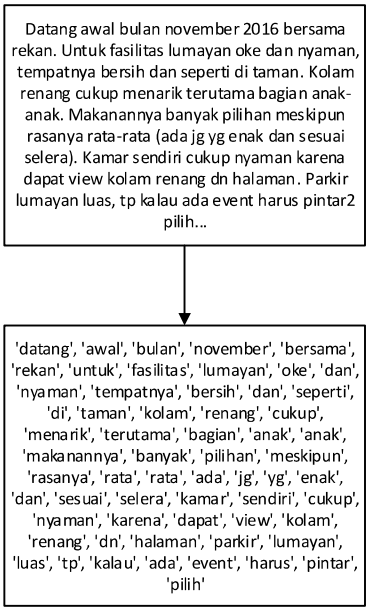
2.5.1 Tokenisasi

Tokenisasi atau *Lexical Analysis* adalah proses memecah dokumen menurut kata, frasa, atau simbol, atau hal-hal penting lain yang disebut dengan token. Tokenisasi bertujuan untuk melakukan eksplorasi kata-kata dalam sebuah kalimat (Gurusamy dan Kannan, 2014). Daftar token dari tokenisasi akan dijadikan input pada pengolahan selanjutnya seperti *filtering, stemming,* dan pemrosesan teks. Selain memecah token, tokenisasi juga berguna untuk menghilangkan tanda baca, *link*, angka, dll yang sering dianggap tidak berguna sehingga dokumen bisa lebih konsisten.

Untuk mengambil token pada tahap ini, tokenisasi biasanya dilakukan melalui tahap berikut:

- Melakukan *case folding* atau mengubah semua kata menjadi huruf kecil.
- Melakukan *cleaning* dengan menghilangkan *link, tag html, script*, dsb.
- Tanda baca dan spasi akan dibuang, atau dianggap bukan token.
- Semua string yang berdekatan dari karakter abjad adalah bagian dari satu token, begitu juga dengan angka.
- Token dianggap terpisah jika ada spasi, *line break*, atau karakter tanda baca.

Proses tokenisasi diilustrasikan dengan Gambar 2.1.



Gambar 2.1 Proses tokenisasi

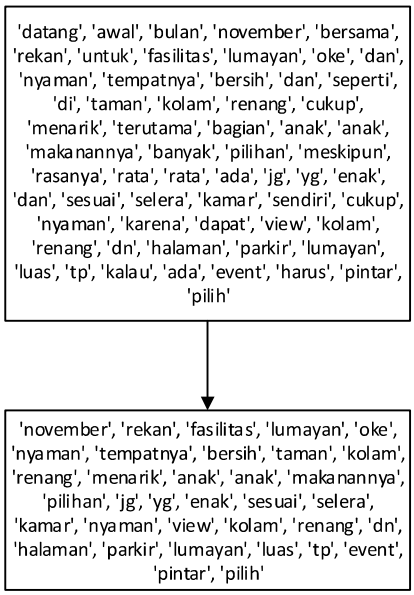
2.5.2 Filtering

Filtering atau *stopword removal* adalah tahap penghilangan *stopword* dari dokumen karena dianggap membuat dokumen menjadi berat dan kurang penting untuk dianalisis dan penghilangan *stopword* dapat mengurangi dimensi ruang dokumen (Ferilli et al., 2014).

Filtering dilakukan karena banyak dokumen cenderung bersifat kurang berarti karena dokumen terdiri dari kata-kata yang disatukan menjadi paragraf sehingga beberapa kata dari dokumen tersebut juga tidak menggambarkan isi dokumen, seperti kata “dan”, “atau”, “di”, dan “ke” yang mempunyai frekuensi tinggi pada dokumen. Maka dari kata-kata tersebut atau yang biasa dikenal dengan nama

stopword harus dibuang. Selain tidak penting, *stopword* tidak membantu dalam proses klasifikasi, dan pembuangannya dapat meningkatkan performa sistem.

Pada umumnya metode *filtering* terbagi menjadi dua yaitu *wordlist* dan *stoplist*. Metode *wordlist* bekerja dengan membuang kata-kata bila kata tersebut ada pada daftar *wordlist*, sedangkan *stoplist* bekerja dengan membuang kata-kata apabila kata tersebut tidak ada dalam daftarnya. Proses *filtering* diilustrasikan dengan Gambar 2.2.



Gambar 2.2 Proses *filtering*

2.5.3 Stemming

Stemming adalah pengubahan sebuah kata menjadi bentuk umumnya atau kata dasarnya yang biasanya dilakukan dengan cara menghilangkan akhiran. Cara ini sering diterapkan pada pencarian teks, terjemahan mesin, rangkuman dokumen, dan klasifikasi teks (Adriani et al., 2007). Sebagai contoh, dalam bahasa Inggris kata *computer, computing,* dan *computation* mempunyai kata dasar yang sama yaitu *comput*- sehingga stemming mengubah ketiga kata tersebut menjadi *compute*.

Dalam bahasa Indonesia, *stemming* tidak hanya cukup dengan menghilangkan akhiran, namun banyak hal yang haris dipertimbangkan seperti awalan, sisipan, dan akhiran sehingga bisa menciptakan kata yang sesuai.

Metode untuk *stemming* Bahasa Indonesia bermacam-macam, namun pada penelitian kali ini digunakan *stemming* Sastrawi yaitu library *open source* yang didapatkan dari GitHub dan bisa diterapkan langsung ke program. Pada Sastrawi, algoritme yang digunakan adalah Nazief Adriani yang mendukung penyusunan kembali kata-kata yang mengalami *stemming* berlebih (Agusta, 2009). Adapun aturan dalam algoritme *stemming* Nazief Adriani adalah:

1. Cari kata yang akan di-*stem* dalam kamus. Jika kata ditemukan maka diasumsikan kata tesebut adalah *root word*. Maka algoritme berhenti.

2. *Inflection Suffixes* akhiran (“-lah”, “-kah”, “-ku”, “-mu”, atau “-nya”) dibuang. Jika akhiran berupa partikel (“-lah”, “-kah”, “-tah” atau “-pun”) maka langkah ini diulangi lagi untuk menghapus *Possesive Pronouns* (“-ku”, “-mu”, atau “-nya”), jika ada.

3. Hapus *Derivation Suffixes* atau akhiran (“-i”, “-an” atau “-kan”). Jika kata tersebut ditemukan di kamus, maka algoritme berhenti. Jika tidak maka ke langkah 3a

    a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k”, maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritme berhenti. Jika tidak ditemukan maka lakukan langkah 3b.

    b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan, lanjut ke langkah 4.

4. Hapus *Derivation Prefix* (awalan). Jika pada langkah 3 terdapat akhiran yang dihapus maka pergi ke langkah 4a, jika tidak pergi ke langkah 4b.

    a. Periksa tabel kombinasi awalan-akhiran yang tidak diijinkan. Jika ditemukan maka algoritme berhenti, jika tidak pergi ke langkah 4b.

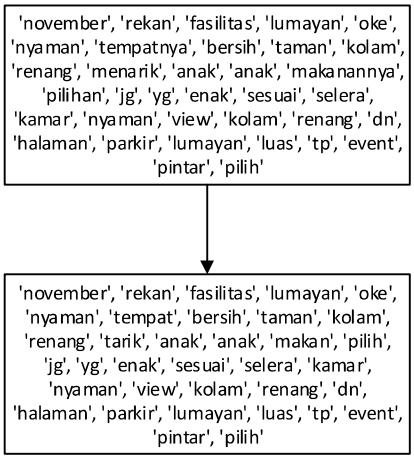
    b. For i = 1 to 3, tentukan tipe awalan kemudian hapus awalan. Jika *root word* belum juga ditemukan lakukan langkah 5, jika sudah maka algoritme berhenti. Catatan: jika awalan kedua sama dengan awalan pertama algoritme berhenti.

5. Melakukan Recoding.

6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai *root word.*

7. Selesai

Proses *stemming* diilustrasikan dengan Gambar 2.3.



Gambar 2.3 Proses *stemming*

2.6 Analisis Sentimen

Analisis sentimen adalah bidang interdisipliner, sebuah bidang dimana pendekatan pemecahan masalahnya dengan menggunakan tinjauan dari berbagi sudut pandang ilmu serumpun secara relevan dan terpadu. Analisis sentiment terdiri dari pemrosesan bahasa alami, analisis teks dan komputasi linguistik untuk mengidentifikasi sentimen dari suatu dokumen (Vinodhini dan Chandrasekaran, 2016).

Analisis Sentimen digunakan untuk menemukan emosi tersembunyi dalam teks secara otomatis. Analisis sentimen memiliki perbedaan dengan *text mining* tradisional yang lebih berfokus kepada melakukan *topic mining* atau klasifikasi karena prosesnya lebih kompleks. Analisis sentimen bisa dibilang termasuk dalam proses *binary-classification* yang membagi datanya menjadi dua kelas yaitu positif dan negatif (Luo, 2016). Proses klasifikasi pada analisis sentimen berbasis *machine learning* terbentuk dari dua tahap yaitu tahap *training* (pelatihan) dan *testing* (pengujian).

2.7 *Naive Bayes Classifier*

Naive Bayes Classifier adalah metode probabilitas sederhana dengan cara menghitung nilai frekuensi dan kombinasi data untuk klasifikasi. Metode ini didasarkan oleh teorema Bayes yang mengasumsikan semua fitur adalah atribut independen yang nilainya tergantung oleh kelasnya (Patil dan Sherekar, 2013).

Meskipun *Naive Bayes* terlihat sederhana, namun metode ini terbukti bekerja dengan baik di banyak bidang, termasuk bidang klasifikasi dokumen dan *spam filtering*. Metode ini membutuhkan data latih untuk mengestimasi parameter yang penting. *Naive Bayes* mewakili metode *supervised* dengan model probabilistik yang memungkinkan kita menangkap ketidakpastian tentang beberapa kejadian (Parveen dan Pandey, 2016). Selain itu, metode ini terbukti memiliki akurasi dan kecepataan yang tinggi saat diaplikasikan kedalam *database*. Secara umum perhitungan dari *Naive Bayes* ditunjukkan oleh persamaan 2.1.

$$P(c|d) = \frac{P(c)\,P(d|c)}{P(d)} \quad (2.1)$$

Keterangan:

$P(c|d)$: Peluang kelas c berdasarkan kondisi d *(posterior probability)*

$P(c)$: Probabilitas kelas *c (prior probability)*

$P(d|c)$: Probabilitas d berdasarkan pada kondisi hipotesis c (*likelihood probability*)

$P(d)$: Probabilitas dari dokumen d (*evidence)*

d : Dokumen dengan kelas yang belum diketahui

c : Kelas yang akan dihitung peluangnya

Nilai *evidence* pada setiap kelas akan selalu tetap. Sedangkan nilai dari posterior nantinya akan dibandingkan dengan nilai posterior kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Maka dari itu *evidence* sering dihilangkan dalam perhitungan *Naive Bayes.* Sehingga rumus *Naive Bayes* setara dengan persamaan 2.2.

$$P(c|d) \approx P(c)\,P(d|c) \quad (2.2)$$

2.7.1 *Multinomial Naive Bayes*

Multinomial Naive Bayes merupakan sebuah metode yang bekerja dengan cara menghitung frekuensi setiap kata pada dokumen (McCallum dan Nigam, 1998). Sehingga peran tokenisasi dalam *Multinomial Naive Bayes* ini sangat penting.

Dalam *Multinomial Naive Bayes*, urutan kejadian munculnya kata dalam dokumen tidak dipedulikan, jadi dokumen dianggap seperti *"bag of words",* sehingga setiap kata diolah menggunakan distribusi *multinomial.* Persamaan *Multinomial Naive Bayes* ditunjukkan oleh persamaan 2.3.

$$P(c|d) = P(c)\prod_{i=1}^{n} P(w_i|c) \quad (2.3)$$

Persamaan 2.3 diatas adalah persamaan *Naive Bayes* dengan menghilangkan evidence.

Keterangan:

$P(c|d)$: Peluang kelas c berdasarkan dokumen d

n : Jumlah seluruh kata pada dokumen

$$P(c) = \frac{N_c}{N} \quad (2.4)$$

Persamaan 2.4 diatas adalah persamaan untuk menghitung peluang setiap kelas dengan keterangan:

$P(c)$: Peluang kelas c

c : Kelas

N_c : Jumlah dokumen kelas c

N : Jumlah seluruh dokumen

Menghitung peluang kata dalam kelas ditunjukkan dengan persamaan dengan persamaan 2.5.

$$P(w_i|c) = \frac{count(w_i, c) + 1}{count\ (c) + |V|} \quad (2.5)$$

Keterangan:

$P(w_i|c)$: Peluang kata ke-i pada kelas c

$count(w_i, c)$: Jumlah kata ke-i pada kelas c

$count\ (c)$: Jumlah seluruh kata pada kelas c

$|V|$: Jumlah kata unik pada semua kelas

2.8 Seleksi Fitur

Seleksi fitur adalah suatu proses pada *Text Mining* yang bertujuan untuk mengurangi dimensi atau jumlah fitur namun tetap mempertahankan fitur yang penting bagi dokumen tanpa mengurangi keakuratannya (Luo, 2016).

Seleksi fitur bekerja dengan menghapus fitur yang berlebihan, tidak relevan, dan tidak berguna dari dokumen karena fitur-fitur tersebut dianggap tidak berkontribusi banyak terhadap proses selanjutnya serta dapat mengurangi akurasi. Jumlah fitur yang sedikit dapat mengurangi kompleksitas dari sebuah dokumen, dan semakin sederhana sebuah dokumen maka akan lebih mudah dimengerti.

2.8.1 *Query Expansion Ranking*

Query Expansion Ranking adalah sebuah metode seleksi fitur yang terinsipirasi dari metode *Query Expansion* yang berguna untuk meningkatkan kualitas *query* yang dimasukkan oleh pengguna kemudian digabung dengan cara *probabilistic weighting model* untuk memberi skor pada pada setiap fitur (Parlar dan Özel, 2016). Perhitungan dari QER ditunjukkan oleh persamaan 2.5.

$$Score_f = \frac{|p_f + q_f|}{|p_f - q_f|} \quad (2.6)$$

Keterangan:

$Score_f$: Skor atau nilai QER

p_f : Peluang fitur *f* dalam dokumen kelas positif

q_f : Peluang fitur *f* dalam dokumen kelas negatif

Nilai-nilai diatas dihitung berdasarkan perhitungan 2.4 dan 2.5.

$$p_f = \frac{df_+^f + 0.5}{n^+ + 1.0} \tag{2.7}$$

$$q_f = \frac{df_-^f + 0.5}{n^- + 0.5} \tag{2.8}$$

Keterangan:

df_+^f : Jumlah dokumen positif yang mengandung fitur *f*

df_-^f : Jumlah dokumen negatif yang mengandung fitur *f*

n^+ : Jumlah dokumen positif

n^- : Jumlah dokumen negatif

2.9 Evaluasi

Pada penelitian ini, evaluasi dilakukan dengan cara menghitung akurasi daridata uji menggunakan rumus akurasi dua kelas (*binary*) dari *Confussion Matrix* sebagai berikut.

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{2.9}$$

Keterangan:

TP : *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.

TN: *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.

FP : *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem.

FN : *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.

Gambaran mengenai *Confusion Matrix* ditunjukkan melalui Tabel 2.1.

Tabel 2.1 *Confusion Matrix*

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	*TP (True Positive)*	*FN (False Negative)*
Negatif	*FP (False Positive)*	*TN (True Negative)*