BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Persaingan opini yang terjadi di media sosial memiliki peran penting dalam meningkatkan calon pelanggan kepada perusahaan atau instansi. Ulasan adalah sumber yang kaya dan berguna untuk pemasaran, sosial, dan lainnya untuk penggalian dan penambangan opini seperti pandangan, suasana hati, dan perilaku. *Review* menggambarkan persepsi terhadap sesuatu, seperti *review* suatu produk, *review* layanan maskapai penerbangan, *review* restoran dan lain-lain. Analisis sentimen adalah bidang penelitian berbasis teks yang sedang berlangsung. Analisis sentimen atau penambangan opini adalah studi tentang cara untuk memecahkan masalah opini publik, sikap, dan emosi suatu entitas, di mana entitas tersebut dapat mewakili individu, peristiwa, atau topik. Analisis sentimen adalah alat penting untuk menganalisis opini di media sosial. Pengukuran ini diawali dengan prapengolahan yang terdiri dari *tokenizing, stopwords removal* dan *stemming*. Penelitian ini menggunakan algoritma *Naïve Bayes* dan algoritma genetika sebagai penerapan seleksi ciri. Ciri seleksi bertujuan untuk mengklasifikasikan teks untuk *review* perusahaan *fashion* daring. Pengukuran ini menghasilkan klasifikasi teks berupa teks positif dan teks negatif. Pengukuran didasarkan pada akurasi *Naïve Bayes* sebelum penambahan algoritma genetika dan setelah penambahan algoritma genetika sebagai seleksi ciri. Validasi menggunakan 10-fold *cross-validation*. Untuk akurasi pengukuran menggunakan matriks konfusi dan kurva ROC. Tujuan dari penelitian ini adalah untuk menghitung peningkatan akurasi algoritma *Naive Bayes* jika menggunakan algoritma genetika untuk seleksi ciri. Hasil penelitian menunjukkan bahwa algoritma genetika mampu meningkatkan akurasi. (Ernawati dkk, 2018).

Analisis sentimen adalah suatu kegiatan yang dilakukan untuk melihat tingkat sentimen atau opini publik yang berkaitan dengan barang atau jasa bahkan seorang tokoh, baik tokoh politik maupun selebriti. Pada penelitian ini dilakukan aplikasi analisis sentimen untuk analisis twitter pada calon presiden Republik Indonesia tahun 2019,

dengan menggunakan bahasa pemrograman python. Ada beberapa langkah yang dilakukan untuk melakukan analisis sentimen ini, yaitu mengumpulkan data menggunakan pustaka python, pengolahan teks, pengujian data latih, dan klasifikasi teks menggunakan metode *Naïve Bayes*. Metode *Naïve Bayes* digunakan untuk membantu mengklasifikasikan kelas atau tingkat sentimen masyarakat. Hasil penelitian ini menemukan bahwa nilai polaritas sentimen positif pasangan Jokowi-Ma'ruf Amin sebesar 45,45% dan nilai negatif sebesar 54,55%, sedangkan pasangan Prabowo-Sandiaga mendapatkan skor sentimen positif sebesar 44,32% dan negatif. 55,68%. Kemudian data gabungan tersebut diuji dari data latih yang digunakan untuk masing-masing calon presiden dan mendapatkan akurasi sebesar 80,90% 80,1%. Pada penelitian ini dilakukan perbandingan menggunakan metode *Naïve Bayes*, SVM dan *K-Nearest Neighbor* (K-NN) yang diuji menggunakan RapidMiner dengan menghasilkan nilai akurasi *Naïve Bayes* sebesar 75,58%, nilai akurasi SVM sebesar 63,99% dan nilai akurasi K-NN sebesar 73,34% (Wongkar dan Meylan, 2019).

Dalam penelitian ini, peneliti menggunakan SVM untuk klasifikasi teks. Terdapat normalisasi kata *stemming* atau *Iemmatisasi* dengan penambahan seleksi ciri *chi-square* pada klasifikasi yang dibuat. Ada juga prapengolahan data yang dilakukan yaitu *stopwords removal* dan *tokenize*. Dalam penelitian ini menggunakan dataset BBC yang berisi 2.225 dokumen dan 5 kategori. Ada 21.813. Ciri hasil penggunaan *stemming* dan 31.007 ciri hasil penggunaan *lemmatization*. Setiap ciri mewakili jumlah kata yang keluar dalam dokumen. Penelitian ini menggunakan matriks kebingungan untuk mengevaluasi hasil klasifikasi teks. Kinerja klasifikasi teks SVM menggunakan *stemming* yang ditingkatkan dengan *chi-square* (metode 1) mendapatkan hasil yang lebih baik daripada menggunakan lemmatisasi yang ditingkatkan dengan *chi-square* (metode 2). Kinerja terbaik diperoleh dengan pengurangan ciri 80%, metode 1 mendapatkan nilai presisi 95%, nilai *recall* 95%, dan nilai akurasi 95,05%. Metode 2 hanya mendapatkan nilai presisi 93%, nilai *recall* 93%, dan nilai akurasi 93,24% menggunakan jumlah pengurangan ciri yang sama (Haryanto dkk, 2018).

Kemunculan virus Covid-19 di awal tahun 2020 menjadi pandemi yang menakutkan bagi dunia, termasuk Indonesia. Infeksi virus Covid-19 berlangsung cepat

karena penularannya bisa melalui kontak manusia. Kondisi ini menimbulkan kekhawatiran di masyarakat. Selain itu, kekhawatiran tersebut juga terjadi pada penumpang angkutan umum, khususnya jalur komuter. Penumpang dalam jumlah besar dan saling mendorong akan menimbulkan kekhawatiran jika penumpang jalur komuter akan menularkan virus Covid-19 ke jalur komuter. Banyak penumpang menuliskan opininya tentang penularan pandemi Covid-19 di media sosial Twitter. Hal ini menimbulkan berbagai pendapat yang bisa bersifat positif, negatif, atau bahkan netral. Oleh karena itu, untuk melihat opini penumpang jalur komuter, maka dilakukan penelitian untuk menganalisis sentimen penularan Covid-19 kepada penumpang komuter. Penelitian ini dilaksanakan dengan menggunakan perbandingan 2 metode, *Naïve Bayes* mengungguli *Decision T*ree dengan akurasi 73,59%. Selanjutnya hasil analisis sentimen merupakan klasifikasi positif dibandingkan dengan 2 kelas lainnya (Sihwi dkk, 2018).

2.2 Dasar Teori

2.2.1 Analisis Sentimen

Analisis Sentimen adalah suatu teknik mengekstraksi data teks untuk mendapatkan informasi tentang sentimen bernilai netral maupun positif dan negatif. Analisis sentimen diberikan oleh pengguna internet pada media sosial untuk memberikan suatu penilaian atau opini pribadi. Secara teknik, analisis sentimen dapat dibagi menjadi empat jenis pendekatan (19), yaitu *Machine learning approach, Lexicon-based approach, Rule-based approach, dan Statistical model approach.* Penentuan polaritas sentimen pada penelitian ini, menggunakan matching kata berdasarkan kamus leksikon (*Lexicon-based approach*). Dilakukan untuk melihat ekspresi pendapat pada sebuah masalah ataupun objek terhadap pandangan negatif atau positif. Analisis sentimen digunakan untuk mengekstraksi informasi yang dapat bernilai dari kumpulan data dengan menentukan polaritas teks tertentu berdasarkan sentimen yang diekstraksi dari teks (Khader, 2018)

2.2.2 Twitter

Twitter adalah layanan jejaring sosial dan microblog daring yang memungkinkan penggunanya untuk mengirim dan membaca teks hingga 140 karakter, yang dikenal dengan sebutan kicauan (Tangdilintin, 2019). Twitter diciptakan oleh Jack Dorsey di tahun 2006 dan pertama meluncur di dunia maya saat Juli 2006 dengan

alamat http://www.twitter.com yang masih digunakan hingga saat ini. Pengguna dapat menulis pesan berdasarkan topik dengan menggunakan tanda # (*hashtag*). Sedangkan untuk menyebutkan atau membalas pesan dari pengguna lain bisa menggunakan tanda @. Terhitung 21 Maret 2016, Twitter genap memasuki usianya yang ke-10. Media sosial ini secara global memiliki sekitar 332 juta pengguna bulanan, dengan 500 juta kicauan dikirim setiap hari dan 200 miliar kicauan dalam setahun. Keunggulan dari Twitter salah satunya adalah dengan akses informasi yang sangat cepat, dibandingkan dengan jejaring sosial sejenis.

2.2.3 Python

Python merupakan bahasa pemrograman tingkat tinggi yang dewasa ini telah menjadi standar dalam dunia komputasi ilmiah. Python merupakan bahasa pemrograman sumber terbuka *multi-platform* yang dapat digunakan pada berbagai macam sistem operasi (Windows, Linux, dan MacOS). Selain itu, Python juga merupakan bahasa pemrograman yang fleksibel dan mudah untuk dipelajari. Program yang ditulis dalam Python umumnya lebih mudah dibaca dan jauh lebih ringkas dibandingkan penulisan program dalam bahasa C atau Fortran. Python juga memiliki modul standar yang menyediakan sejumlah besar fungsi dan algoritma, untuk menyelesaikan pekerjaan seperti mengurai data teks, memanipulasi dan menemukan berkas dalam tempat penyimpanan (*disk*), membaca/menuliskan file terkompresi, dan mengunduh data dari server web. Dengan menggunakan Python, para pemrogram juga dapat dengan mudah menerapkan teknik komputasi tingkat lanjut, seperti pemrograman berorientasi objek (Herho, 2018).

2.2.4 *Text Mining*

Text Mining adalah proses ekstraksi pola dari sejumlah sumber data yang tidak terstruktur. *Text mining* memiliki tujuan dan menggunakan proses yang sama dengan *data mining*. Masukan untuk *text mining* adalah kata yang tidak terstruktur, seperti dokumen, komentar, ulasan, dan lain-lain. Sementara itu, masukan untuk *data mining* adalah data terstruktur. Data terstruktur direpresentasikan dalam skema yang jelas sehingga mudah untuk dianalisa maupun diintegrasikan dengan data terstruktur lainnya. Sedangkan data tidak terstruktur direpresentasikan dalam berbagai bentuk sehingga

sangat sulit untuk dianalisa maupun diintegrasikan dengan sumber data lain (Afifanto, 2015).

Selain itu, *text mining* memiliki peran yang semakin penting dalam aplikasi, seperti mengetahui isi pada teks secara langsung dari proses *text mining* tanpa perlu membaca satu persatu teks atau tulisan yang ada. Proses *text mining* adalah sama dengan *data mining*, kecuali beberapa metode dan data yang dikelolanya seperti data teks yang tidak terstruktur, terstruktur sebagian maupun terstruktur seperti teks email, teks HTML, maupun teks komentar serta dari berbagai sumber (Muhammad dkk, 2019).

2.2.5 Prapengolahan

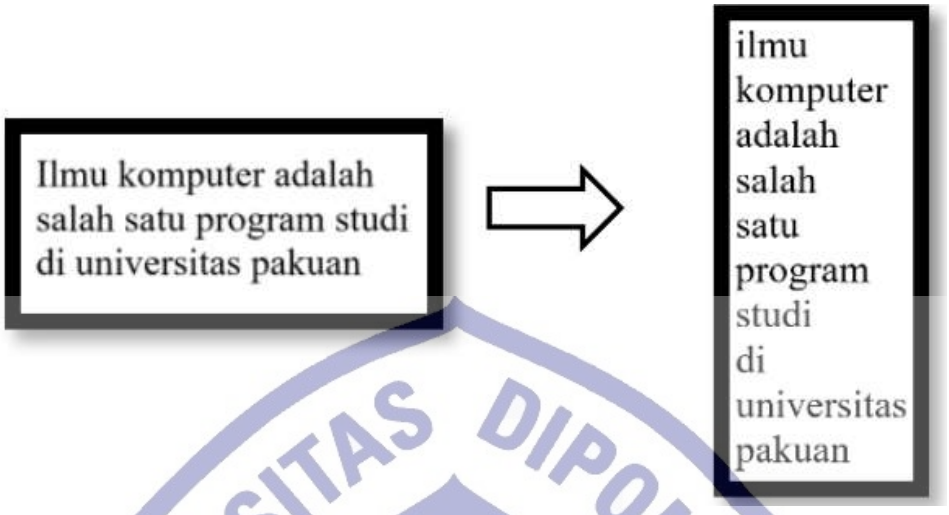
Prapengolahan merupakan proses awal klasifikasi dokumen yang bertujuan menyiapkan data agar menjadi terstuktur. Prapengolahan merupakan salah satu langkah yang penting dalam analisis sentiment (Nugroho dkk, 2016). Tahapan dalam prapengolahan di antaranya adalah:

1. *Case Folding*

   Tidak semua teks dalam dokoumen konsisten dalam penggunaan huruf kapital. Maka dari itu *case folding* digunakan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (menjadi huruf kecil). Sebagai contoh, pengguna ingin mendapatkan informasi "KOMPUTER" dengan mengetik "KomPuter" atau "kompUTER", tetapi diberikan hasil pencarian yang sama yakni "komputer". Dalam hal ini *case folding* digunakan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. (Zuraiyah, Utami, & Herlambang, 2016)

2. *Tokenizing*

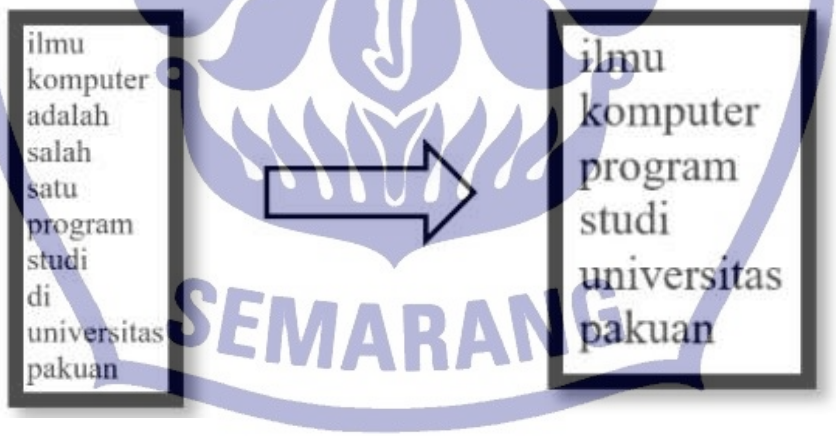
   Tahapan tokenizing merupakan tahapan pemotongan string input berdasarkan tiap kata yang menyusunnya (Zuraiyah, Utami, & Herlambang, 2016). Dapat dilihat pada Gambar 2.1



Gambar 2. 1 Tahap *tokenizing*

3. *Filtering*

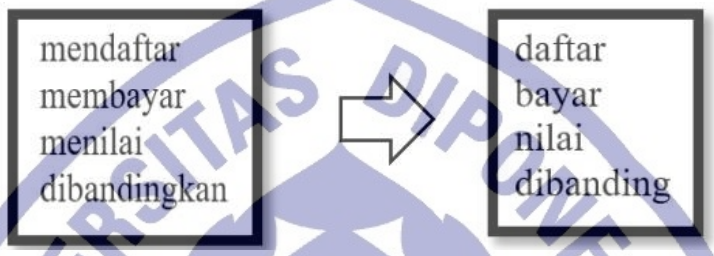
Tahap *filtering* adalah tahap mengambil kata-kata penting dari hasil token menggunakan *stoplist* yang berfungsi untuk membuang kata kurang penting atau *wordlist* yang dapat menyimpan kata penting (Zuraiyah, Utami, & Herlambang, 2016). Contohnya dapat dilihat pada gambar pada Gambar 2.2.



Gambar 2. 2 Tahap *filtering*

4. *Stemming*

Teknik *stemming* diperlukan untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen dan juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk yang berbeda (Zuraiyah, Utami, & Herlambang, 2016). Gambar 2.3 menunjukkan proses *stemming chatbot*.



Gambar 2. 3 Tahap *Stemming*

5. *Cleaning*

Tahap *Cleaning* merupakan proses membersihkan tinjauan dari kata-kata yang tidak diperlukan untuk mengurangi kesalahan acak di dalam nilai atribut pada proses klasifikasi. Kata-kata yang dihilangkan adalah karakter.

2.2.6 *Chi Square*

Uji *Chi*-kuadrat merupakan metode statistika pengujian hipotesis data diskrit yang mengevaluasi korelasi antar dua variabel dan menentukan apakah variabel tersebut tidak berkaitan atau saling terkait. Pada tes keterkaitan, ketika diterapkan pada populasi suatu subjek, menentukan apakah subjek tersebut terkait atau tidak (Suharno, Fauzi, dan Perdana, 2017). Fungsi dari uji *Chi*-kuadrat dapat dilihat pada Persamaan 2.1

$$X^2(t,c) = \frac{N(AD-CB)^2}{(A+C)(B+D)(A+B)(C+D)} \quad (2.1)$$

Keterangan:

t : Kata

c : Kelas/Kategori

N : Jumlah dokumen latih

A : Jumlah dokumen pada kategori c yang memuat t

B : Jumlah dokumen bukan kategori c yang memuat t

C : Jumlah dokumen pada kategori c yang tidak memuat t

D : Jumlah dokumen bukan kategori c yang tidak memuat t

Untuk dapat melakukan seleksi ciri yang tidak dipakai berdasarkan nilai *Chi*-kuadrat dari sebuah kata terhadap kategori, diperlukan nilai *Chi*-kuadrat tunggal dari kata. Untuk dapat mengetahui nilai *Chi*-kuadrat tunggal dari suatu kata diperoleh dengan menjumlahkan nilai *Chi*-kuadrat tiap kata antar kategori. Fungsi untuk mendapatkan nilai *Chi*-kuadrat tunggal tiap kata dapat dilihat pada Persamaan 2.

$$X^2(t) = \sum_{c=1}^{k} x^2(t,c) \quad (2.2)$$

Keterangan:

t : Kata

c : Kelas/Kategori

Setelah nilai *Chi*-kuadrat pada tiap kata diketahui, dilakukan pengurutan kata berdasarkan nilai *Chi*-kuadrat tertinggi hingga terendah. Hal ini menandakan bahwa semakin besar nilai *Chi*-kuadrat, semakin dependen suatu ciri, dan semakin penting ciri tersebut untuk digunakan dalam proses klasifikasi.

2.2.7 *Naïve Bayes*

Pengklasifikasian *Naïve Bayes* adalah suatu model independen yang membahas mengenai klasifikasi sederhana berdasarkan teorema Bayes. Naïve Bayes merupakan suatu algoritma yang dapat mengklasifikasikan suatu variable tertentu dengan menggunakan metode probabilitas dan statistik. Secara garis besar algoritma *Naïve Bayes* dapat dijelaskan seperti persamaan 2.3 (Kurniawan, 2018):

$$P(R|S) = \frac{P(R)P(S|R)}{P(S)} \quad (2.3)$$

Keterangan:

R : Data yang belum diketahui kelasnya

S : Hipotesis pada data R yang merupakan class khusus

P(R|S) : Nilai probabilitas pada hipotesis R yang berdasarkan kondisi S

P(R) : Nilai probabilitas pada hipotesis R

P(S|R) : Nilai probabilitas S yang berdasarkan dengan kondisi hipotesis

R P(S) : Nilai probabilitas S

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

Metode TF-IDF merupakan metode untuk menghitung bobot suatu kata (*term*) terhadap dokumen. Metode ini juga terkenal efisien, mudah dan memiliki hasil yang akurat[8]. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan inverse frekuensi dokumen yang mengandung kata tersebut. Frekuensi kemunculan kata di dalam dokumen yang diberikan menunjukan seberapa penting kata tersebut di dalam dokumen (Riyani, Naf'an, dan Burhanuddin, 2019).

Pendekatan TF-IDF menyajikan teks dengan ruang tabel yang disetiap ciri dalam teks sesuai dengan satu kata. TF (*Term Frequency*) akan menghitung frekuensi kemunculan sebuah kata dan dibandingkan jumlah seluruh kata yang ada di dalam dokumen, berikut persamaan 2.4 yang digunakan untuk menghitung TF.

$$tf\ (i) = \frac{freq\ (t_1)}{\sum freq\ (t)} \quad (2.4)$$

Keterangan:

tf(i) : nilai *Term Frequency* sebuah kata dalam sebuah dokumen.

Freq (t_1) : frekuensi kemunculan sebuah kata dalam sebuah dokumen.

$\sum freq\ (t)$: jumlah keseluruhan kata dalam dokumen.

Sementara IDF (*Inverse Document Frequency*) menghitung algoritma dari jumlah seluruh dokumen dan dibandingkan dengan jumlah dokumen, dalam dokumen tersebut kata (*t*) yang dimaksud muncul. Berikut persamaan 2.5 yang digunakan untuk menghitung IDF.

$$tdf\ (i) = \log \frac{|D|}{|\{d:t_i \in d\}|} \quad (2.5)$$

Keterangan:

idf(i) : nilai *Inverse Document Frequency* sebuah kata di seluruh isi dokumen.

|D| : jumlah seluruh dokumen.

|(d: ti∈d}| : jumlah dokumen yang mengandung kata (t).

Dengan kedua persamaan tersebut maka dapat ditentukan nilai bobot (*w*) sebuah kata dalam sekumpulan dokumen, dengan menghitung perkalian dari kedua persamaan sebelumnya. Berikut persamaan 2.6 untuk menentukan nilai bobot (*w*) sebuah kata.

$$Weight\ (tf - idf)_i = tf(i) \times idf(i) \quad (2.6)$$

2.2.9 *Confusion Matrix*

Confusion Matrix adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada confusion matrix yaitu *True Positives, True Negatives, False Positive, dan False Negatives*. *True positives* adalah jumlah *record* positif yang diklasifikasikan sebagai positif, *false positives* adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, *false negatives* adalah jumlah *record* positif yang diklasifikasikan sebagai negatif, kemudian masukkan data uji. Setelah data uji dimasukkan ke dalam *confusion matrix*, hitung nilai-nilai yang telah dimasukkan tersebut untuk dihitung jumlah sensitivitas (*recall*), *specificity*, *precision*, dan *accuracy*. Tabel 2.1 merupakan tabel *Confusion Matrix*

Tabel 2. 1 *Confusion Matrix*

		True Value	
		True	*False*
Prediction	*True*	TP *Correct result*	FP *Unexpected Result*
	False	FN *Missing result*	TN *Correct Absence of result*

Accuracy mengukur jumlah total prediksi yang benar dibandingkan dengan total data (Ramdhani, Andreswari, dan Hasibuan, 2018). Berikut ini adalah rumus untuk menghitung *accuracy* ditunjukkan pada Persamaan 2.7.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2.7)$$

Precision adalah rasio item yang relevan dipilih untuk semua item dipilih (Ramdhani, Andreswari, dan Hasibuan, 2018). Berikut ini adalah rumus untuk menghitung *precision* ditunjukkan pada Persamaan 2.8.

$$Precision = \frac{TP}{TP+FP} \quad (2.8)$$

Recall didefinisikan sebagai rasio item yang relevan dipilih dengan jumlah total item yang relevan (Ramdhani, Andreswari, dan Hasibuan, 2018). Berikut ini adalah rumus untuk menghitung *recall* ditunjukkan pada Persamaan 2.9.

$$Recall = \frac{TP}{TP+FN} \quad (2.9)$$

F1-Measure adalah kombinasi rata-rata *harmonic precision* dan *recall* yang berbanding lurus dengan nilai keduanya (Ramdhani, Andreswari, dan Hasibuan, 2018). Berikut ini adalah rumus untuk menghitung *F1-Measure* ditunjukkan pada Persamaan 2.10.

$$F1 - Measure = \frac{2*precision*recall}{precision+recall} \quad (2.10)$$