BAB 2

LANDASAN TEORI

Bab ini merupakan penjabaran teori yang digunakan guna mendukung jalannya penelitian ini. Teori-teori tersebut mencakup analisis sentimen, *text preprocessing*, metode *Naïve Bayes*, *Confusion Matrix, Grid-Search Cross Validation.*

2.1 Analisis Sentimen

Analisis sentimen atau *opinion mining* merupakan bidang studi untuk menganalisis pendapat, emosi, penailaian, sikap seseorang terhadap suatu barang, peristiwa, masalah, dll. Analisis sentimen juga sering dikenal sebagai *opinion extraction*, dan *sentiment mining*. Tugas dasar dari analisis sentimen ialah melakukan pengelompokkan berdasarkan polaritas yang ada di dalam suatu dokumen, kalimat, dan dapat menentukan kalimat atau fitur tersebut bersifat positif, negatif atau netral (Liu, 2012).

Terdapat beberapa konsep yang berhubungan antara sentimen dan opini, salah satunya adalah emosi atau perasaan. Emosi dasar manusia dapat dibedakan menjadi lima, diantaranya: cinta, senang, sedih, marah, dan takut. Sehingga emosi dasar manusia dapat dibagi menjadi dua kategori besar, yaitu emosi positif dan emosi negatif. (Shaver, dkk., 2001).

2.2 Depresi

Depresi merupakan sebuah gangguan psikologis yang umum ditemui. Depresi merupakan gangguan yang ditandai dengan kondisi emosi sedih dan muram yang berkaitan dengan gejala kognitif, fisik, dan interpersonal (Pane, 2020).

Depresi merupakan salah satu masalah kesehatan mental yang utama saat ini, Menurut data World Health Organization (WHO, 2020), lebih dari 264 juta orang dari segala aspek umur menderita gangguan psikologis berupa depresi, data juga menunjukkan bahwa perempuan lebih banyak menderita gangguan psikologis berupa depresi dibandingkan dengan laki-laki.

Berikut ini adalah pengertian Depresi menurut para ahli:

- Menurut Yosep dan Sutini (2007), depresi merupakan salah satu bentuk gangguan kejiwaan yang ditandai kemurungan, kesedihan, tidak ada semangat hidup, merasa tidak berdaya, putus asa, perasaan bersalah.
- Menurut Kartono (2010), depresi merupakan kondisi di mana hati merasakan kepedihan, kesenduan yang ditimbulkan oleh rasa sakit hati yang mendalam, penyalahan diri sendiri dan trauma secara psikis.
- Menurut Chaplin (2002), depresi didefinisikan atau dibagi menjadi 2 kondisi, kondisi normal dan kondisi yang masuk kedalam tahap serius, di mana penderita dengan kondisi normal merupakan sebuah kondisi hati yang merasakan kesedihan, patah semangat yang ditandai dengan menurunnya performa kerja, pesimisme, sedangkan penderita yang dalam tahap serius, depresi dapat didefinisikan dengan sebuah kondisi hati yang merasakan ketidakmauan ekstrim dalam menanggapi hal-hal yang bersangkutan dengan dirinya, ketidak mampuan dan putus asa dalam melakukan segala hal.

2.2.1 Penyebab Depresi

Menurut Adrian (2019), depresi dapat disebabkan oleh kombinasi dari beberapa faktor, di mana faktor-faktor tersebut dapat dibagi menjadi: faktor biologi, faktor psikologis dan faktor sosial. Faktor-faktor ini memiliki keterkaitan antara faktor yang satu dengan faktor yang lainnya.

Faktor psikologis kerap ditemukan pada seseorang yang berfokus pada tekanan yang dialami dibandingkan dengan perasaan dan juga sering kali merenung daripada mengalihkan perasaan tersebut. Sementara faktor biologis kerap kali dihubungkan dengan adanya perubahan hormon *norepinefrin* dan *serotonin* yang memiliki peran dalam perubahan suasana hati seseorang. Berbeda halnya dengan faktor psikologis, faktor sosial lebih berfokus pada tekanan yang dialami secara sosial, contohnya: masalah keuangan, trauma masa kecil, faktor usia dan jenis kelamin.

2.2.2 Risiko Yang Dtiimbulkan Akibat Depresi

Menurut Adrian (2019), terdapat beberapa risiko yang dapat ditimbulkan akibat mengalami gangguan kejiwaan, diantaranya:

a. Bunuh Diri
b. Gangguan Tidur
c. Gangguan Interpersonal
d. Gangguan dalam pekerjaan
e. Gangguan Pola Makan

2.3 Twitter

Twitter merupakan salah satu layanan jejaring sosial tekstual atau mikrobloging sehingga digunakan oleh penggunanya untuk mengirimkan serta membaca pesan yang biasa dikenal dengan *tweet* (Twitter, 2013). Mikroblog sendiri memiliki pengertian yaitu suatu bentuk blog yang memungkinkan penggunanya untuk menuliskan teks singkat kurang dari 200 karakter, akan tetapi pengguna Twitter hanya dapat melakukan mikroblog sebanyak 140 karakter untuk setiap *tweet* atau postingan yang dibuat. Mikroblog digunakan untuk membagikan informasi, pendapat, keluhan atas suatu fenomena. Selain dapat melakukan mikroblog, pengguna Twitter juga dapat menulis pesan berdasarkan topik yang sedang umum dibicarakan oleh masyarakat sekitar, yaitu dengan menggunakan tanda # (*hashtag)* (Nations, 2019).

Twitter juga memiliki berbagai fitur yang dapat digunakan atau diakses oleh penggunananya, antara lain:

- Halaman Utama

  Pada halaman utama, pengguna dapat melihat *tweet* atau yang dikirimkan oleh diri sendiri, maupun orang yang menjadi teman dari pengguna.

- Pengikut (*Followers*)

  Pengikut pada aplikasi Twitter merupakan pengguna lain yang ingin mengikuti tulisan-tulisan atau *tweet* yang dibuat oleh pengguna yang diikutinya.

- Profil (*Profile*)

  Pada halaman ini, pengguna aplikasi Twitter dapat mengubah atau menghapus seluruh data diri serta *tweet* yang pernah dikirim.

- *Mentions*

  Fitur ini dapat digunakan oleh pengguna untuk meyebutkan pengguna lainnya agar dapat melakukan percakapan.

- Pesan Langsung (*Direct Message*)

  Fungsi ini dapat digunakan untuk melakukan pengguna untuk mengirimkan pesan langsung kepada pengguna lainnya layaknya menggunakan SMS.

- *Hashtag*

  Hashtag atau "#" dapat digunakan oleh pengguna dalam *tweet* untuk mencari suatu topik atau tergabung dalam topik tersebut.

- Topik Terkini

  Fitur ini disediakan oleh Twitter guna agar penggunanya dapat melihat topik-topik apa saja yang sedang ramai dibicarakan pada aplikasi Twitter.

Twitter juga menyediakan sebuah cara agar peneliti dapat mengumpulkan data *tweet* pengguna Twitter untuk diolah menjadi sebuah informasi baru, yang disebut dengan Twitter *API*. Untuk mendapatkan akses terhadap Twitter *API*, peneliti harus mendaftarkan aplikasinya kepada pihak Twitter, agar akses tersebut tidak disalah gunakan (Twitter, 2013; Harijiatno, 2019).

2.4 Text Preprocessing

Tahap praproses data atau *pre-processing* data merupakan tahap lanjutan setelah mendapatkan data mentah sebelum melakukan proses lain. Pada umumnya, praproses data dilakukan guna mengeliminasi data yang tidak diperlukan sehingga data lebih mudah untuk diproses oleh sistem. Tahap ini merupakan tahapan yang penting, terutama jika ingin melakukan analisis sentimen (Mujilahwati, 2016).

Praproses data dilakukan guna untuk menghilangkan *noise* yang ada pada data mentah, maka untuk menghilangkan *noise* tersebut dapat dilakukan beberapa cara, antara lain (Mujilahwati, 2016; Latha, dkk., 2012):

a) *Case Folding*

Case folding merupakan salah satu bentuk *text preprocessing* yang paling sederhdana dan efektif. Tujuan dari dilakukannya *case folding* adalah mengubah semua huruf dalam sebuah dokumen menjadi huruf kecil. Selain itu *case folding* juga dilakukan agar dokumen hanya berisikan huruf saja, karakter selain huruf akan dihilangkan dan dianggap sebagai sebuah *delimiter*.

Ada beberapa cara yang dapat digunakan untuk melakukan tahap *case folding*, diantaranya:

- Menghapus angka

  Penghapusan angka ini bertujuan agar kalimat-kalimat yang ada pada *dataset* terhindar dari angka-angka yang tidak *relevan* terhadap analisa yang akan dilakukan, contoh:

  Input : Berikut ini adalah 5 negara dengan pendapatan tertinggi.

  Output : Berikut ini adalah negara dengan pendapatan tertinggi.

- Menghapus tanda baca

  Sama halnya dengan menghapus angka, menghapus tanda baca juga dilakukan agar kalimat pada *dataset* menjadi lebih *relevan* pada saat dilakukan analisis. Contoh:

  Input : Ini adalah (contoh) kalimat. [dengan] tanda baca.

  Output : Ini adalah contoh kalimat dengan tanda baca

- Menghapus karakter kosong

  Tahap ini dilakukan guna untuk menghapus spasi di awal dan di akhir kalimat. Contoh :

  Input : "\t ini kalimat contoh\t"

  Output : ini kalimat contoh

b) Normalisasi

Normalisasi merupakan proses penyamaan sebuah ejaan atau kata yang memiliki arti/makna yang sama namun memiliki perbedaan penulisan yang berbeda (Manning, 2008). Contohnya:

Input : "ak", "w", "gw", "gua", "gue"

Output : "aku"

c) *Filtering* (*Remove Stopword*)

Filtering merupakan tahap mengambil kata-kata penting dari hasil token menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (meyimpan kata penting). Contoh kata yang dianggap tidak memiliki makna dalam Bahasa Indonesia adalah "yang", "dan", "di", dari", dll. *Stopword* dilakukan guna untuk sistem dapat berfokus pada kata-kata penting saja.

d) *Stemming*

Stemming merupakan proses menghilangkan infleksi kata ke bentuk dasarnya, namun bentuk dasar tersebut tidak berarti sama dengan akar kata (*root word*). Misalnya kata “mendengarkan”, “dengarkan”, “didengarkan” akan diubah menajdi kata “dengar”. Proses *stemming* antar bahasa yang satu dengan yang lainnya tentu memiliki cara yang berbeda-beda, pada Bahasa Inndonesia, semua kata imbuhan baik itu sufik dan prefiks juga dihilangkan.

e) Tokenisasi (*tokenizing*)

Tokenisasi adalah proses pemisahan kata pada kalimat menajdi potongan-potongan yang disebut sebagai *token* yang akan digunakan untuk analisa. Kata, angka, simbol, tanda baca, dan entitas penting lainnya dapat dianggap sebagai token. Contoh:

Input : Andi mencuci bajunya setiap hari

Output : |Andi| |mencuci | |bajunya| |setiap| |hari|

2.5 Klasifikasi Teks

Klasifikasi teks merupakan sebuah proses untuk membentuk golongan-golongan (kelas-kelas) pada suatu dokumen yang didasari oleh kelas kelompok yang telah diketahui atau dibuat sebelumnya (Ramya dan Pinakas, 2014). Terdapat beberapa tahapan pada klasifikasi teks antara lain:

- Rekayasa Fitur

  Rekayasa fitur merupakan tahapan latih atau *training* yang terdiri dari beberapa tahapan, diantara lain: seleksi fitur, pembobotan fitur (*feature weighting*), dan *dictionary construction*. Tujuan dari dilakukannya rekayasa

fitur adalah untuk menghilangkan fitur-fitur yang tidak relevan dan selalu muncul pada semua dokumen (Nikhath, dkk., 2016).

- Generasi Model Klasifikasi

  Generasi model klasifikasi merupakan tahap untuk membangun algoritma klasifikasi, pada penelitian ini menggunakan metode *Multinomial Naïve Bayes* dan *Complement Naïve Bayes*, metode ini digunakan untuk mengklasifikasikan dokumen yang tidak diketahui kategorinya (Nikhath, dkk., 2016).

- Pengkategorian Dokumen

  Tahapan ini merupakan tahapan untuk melakukan klasifikasi pada dokumen yang tidak diketahui asal kategorinya, pengkategorian dokumen ini hanya dapat dilakukan apabila dokumen yang akan dikategorikan telah melewat tahapan *preprocessing* dan *feature weighting*.

2.6 Eksraksi Fitur dengan TF-IDF

Setiap dokumen diwakili oleh vektor dengan pengenalan elemen-elemen yang dikenali dari tahap ekstraksi dari dokumen. Vektor merupakan bobot dari setiap pemberhentian yang menggunakan dasar perhitungan dengan metode TF-IDF. TF-IDF merupakan metode yang mengaitkan antara *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) (Wijaya dan Santoso, 2016; Suharno, dkk., 2017).

Pada tahap atau skema TF-IDF, TF merupakan jumlah kemunculan setiap kata dalam tiap dokumen, sementara IDF merupakan jumlah kemunculan kata dalam keseluruhan dokumen. Skema TF-IDF juga berguna untuk mengubah atau

mengurangi panjang dokumen yang bervariasi menjadi panjang dokumen yang tetap (Prihatini, 2016).

Untuk mencari bobot atau nilai dari TF-IDF, dapat digunakan rumus seperti di bawah ini (Manning, dkk., 2009):

a. *Term Frequency* (TF)

TF merupakan cara pembobotan kata yang paling sederhana. Bobot pada kata t pada dokumen dapat diberikan dengan:

$$w_{ij} = tf_{ij}\ x\ idf \qquad \ldots(2.1)$$

Keterangan:

W_{ij} : bobot kata i pada dokumen j

tf_{ij} : jumlah kemunculan kata i pada dokumen j

df_j : jumlah dokumen j yang berisi kata i

b. *Inverse Document Frequency* (IDF)

Jika TF melihat kemunculan kata pada dokumen, IDF melihat kemunculan kata dalam kumpulan dokumen. Nilai IDF pada suatu kata diberikan dengan cara:

$$idf = \log \frac{N}{df_j} \qquad \ldots(2.2)$$

Keterangan:

N : jumlah dokumen

2.7 Naïve Bayes (NB)

Naive Bayes merupakan salah satu metode pada *Machine Learning* dengan melakukan perhitungan probabilitas. Pada dasarnya, *Naïve Bayes Classifier* menggunakan konsep teorema Bayes. Ciri utama dari *Naïve Bayes Classifier* adalah mengasumsikan yang sangat kuat (naif) untuk masing-masing kondisi atau kejadian (Natalius, 2010).

Pada teorema Bayes, bila terdapat dua kejadian yang terpisah (kejadian A dan kejadian B), maka teorema Bayes akan dirumuskan sebagai berikut (Jurafsky dan Martin, 2019):

$$P(X_k|Y) = \frac{P\ (Y|X_k)}{\sum_i P(Y|X_i)} \quad \ldots(2.3)$$

Di mana keadaan Posterior (Probabilitas X_k dalam Y) dapat dihitung dari keadaan Prior (Probabilitas Y dalam X_k dibagi dengan jumlah dari semua probabilitas Y di dalam semua X_i.

Setelah itu, untuk dapat dilakukan proses klasifikasi oleh sistem dengan menggunakan HMAP (*Hypothesis Maximum Appropri Probability*) untuk menyatakan hipotesa berdasarkan nilai probabilitas pada kondisi prior yang diketahui. HMAP merupakan model penyederhanaan dari metode bayes atau yang dikenal juga dengan *Naïve Bayes*. HMAP juga digunakan oleh sistem *machine learning* untuk mendapatkan hipotesis terhadap suatu keputusan. Rumus HMAP yang digunakan oleh sistem, seperti di bawah ini (Jurafsky dan Martin, 2019):

$$P(S|X) = argmax_{x \in X}\ P(Y \mid X)\ P(X) \quad \ldots(2.4)$$

2.8 Klasifikasi Multinomial Naïve Bayes

Klasifikasi Multinomial *Naïve Bayes* adalah klasifikasi yang diawali dengan tahap pengambilan jumlah kata yang tampil pada tiap dokumen dengan asumsi setiap dokumen memiliki beberapa kejadian dalam kata dengan panjang yang tidak tergantung pada kelasnya.

Model klasifikasi ini merupakan hasil perkalian antara *prior probability* dan *conditional probability* dan menghasilkan sebuah *posterior probability* dengan nilai paling besar untuk suatu kelas tertentu dengan rumus di bawah ini (Harijiatno, 2019; Manning, dkk., 2009):

$$c_{map} = \arg\max_{c \in C} P(c|d) \prod 1 \leq k \leq n_d \; P(t_k|c) \qquad \ldots(2.5)$$

Keterangan:

a. arg max : fungsi untuk mencari nilai *posterior probability* terrbesar suatu kelas

b. $P(t_k|c)$: *Conditional probability*, peluang kemunculan kata *k* dalam suatu kelas tertentu

c. P(c) : *Prior probability*, peluang kemunculan sebuah kelas dari seluruh pengamatan yang dilakukan. Nilai probabilitas sebuah dokumen d yang berada dalam kelas c (P(c)) dihitung dengan rumus:

$$P(c) = \frac{N_c}{N\prime} \qquad \ldots(2.6)$$

Keterangan:

a. N_c : Jumlah dokumen di dalam kelas c

b. N’ : Jumlah total dokumen *training*

Untuk menghitung *conditional probability* dapat menggunakan rumus di bawah ini:

$$P(t|c) = \frac{T_{ct}}{\sum t' \in V \, T_{ct'}'} \quad \text{...(2.7)}$$

Keterangan:

a. T_{ct} : Frekuensi suatu kata dalam kelas c pada dokumen *training* termasuk yang berulang

b. $T_{ct'}$: Jumlah total kata dalam suatu kelas c

Pada proses klasifikasi, seringkali terdapat kata yang tidak pernah muncul pada suatu kelas tertentu pada data training, sehingga dapat menyebabkan peluang kata dalam kelas tersebut memiliki nilai 0 karena *conditional probability*nya bernilai 0, hal ini dapat menyebabkan kesalahan sistem dalam melakukan klasifikasi terhadap kata-kata dalam suatu dokumen. Untuk menghilangkan atau menghindari adanya peluang dengan nilai 0, maka perlu digunakan cara *add-one smoothing* (*Laplace smoothing).* Cara kerja dari *Laplace smoothing* adalah menambahkan angka 1 pada setiap perhitungan angka (Harijiatno, 2019; Manning, dkk., 2009):

$$P(t|c) = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B'} \quad \text{…(2.8)}$$

Keterangan:

B’ : Total kosakata (kata unik) pada keseluruhan kelas dalam dokumen training.

2.9 Klasifikasi Complement Naïve Bayes

Complement Naïve Bayes merupakan pengembangan dari metode *Naïve Bayes* dengan menerapkan *parameter* kelas dengan menggunakan data dari semua kelas kecuali kelas terfokus. Sedangkan metode *Naïve Bayes* akan mengestimasi *parameter* kelas menggunakan data dari kelas terfokus.

Complement Naïve Bayes (CNB) merupakan suatu metode untuk menghitung probabilitas suatu data pada kelas tertentu dengan cara mengidentifikasi bahwa data tersebut berada di kelas lain (Rennie, dkk., 2019). Probabilitas dapat dihitung dengan menggunakan rumus:

$$\theta_{ci} = \frac{N_{ci} + \alpha_i}{\mathrm{N_c} + \alpha} \qquad \ldots(2.9)$$

Keterangan:

a. θ_{ci} : Probabilitas kata i muncul pada kelas lain selain kelas c
b. α_i : *Laplace smoothing* untuk semua kata
c. α : Menunjukkan jumlah α_i untuk setiap kata yang muncul
d. i : kata
e. N_{ci} : jumlah kata i yang muncul sebagai tambahan untuk kelas c
f. $\mathrm{N_c}$: jumlah seluruh kata yang muncul sebagai tambahan untuk kelas c

Laplace Smoothing digunakan untuk menghindari terjadinya perhitungan nilai probabilitas dengan nilai 0. *Laplace Smoothing* memiliki nilai 1 sebagai nilai dasarnya.

Untuk melakukan klasifikasi, *Complement Naïve Bayes* akan menggunakan rumus:

$$argmin\ p(y) \prod \frac{1}{p(w|y)^{fi}} \quad \ldots(2.10)$$

Pada metode yang dilakukan oleh *Complement Naïve Bayes*, klasifikasi akan dilakukan dengan mencari nilai terendah dari semua kemungkinan yang ada.

2.10 *Confusion Matrix*

Confusion matrix merupakan suatu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. *Confusion matrix* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem untuk diukur keakuratannya. *Confusion matrix* juga merupakan salah satu cara dalam melakukan visualisasi terhadap hasil pembelajaran sistem, visualisasi yang ditampilkan memuat dua kategori atau lebih (Rahman, dkk., 2017). Tabel di bawah merupakan contoh hasil *confusion matrix* prediksi dua kelas.

Tabel 2.1 Confusion Matrix

		Kelas Sebenarnya	
		1	2
Kelas Prediksi	1	True Positive	False Negative
	2	False Positive	True Negative

Keterangan:

a. *True Positive* (TP): merupakan jumlah data dengan kelas positif yang diklasifikasikan positif.
b. *True Negative* (TN): merupakan jumlah data dengan kelas negative yang diklasifikasikan negatif

c. *False Positive* (FP): merupakan jumlah data dengan kelas positif yang diklasifikasikan negatif.

d. *False Negative* (FN): merupakan jumlah data dengan kelas negatif yang diklasifikasikan positif.

Perhitungan Akurasi yang dilakukan oleh *confusion matrix* berdasarkan Tabel di atas dapat menggunakan persamaan sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \; x \; 100\% \qquad \ldots(2.11)$$

2.11 *Grid Search Cross Validation*

Grid Search Cross Validation merupakan metode alternatif untuk mencari parameter terbaik untuk suatu model, sehingga algoritma pengklasifikasi dapat lebih akurat melakukan prediksi terhadap data yang belum dilakukan *labeling*, pada kasus ini, *Grid Search Cross Validation* dikategorikan sebagai metode yang lengkap, karena parameter terbaik harus dilakukan uji coba atau dieksplorasi menggunakan setiap parameter yang ada. Setelah dilakukan eksplorasi, *Grid Search Cross Validation* akan menampilkan skor untuk setiap parameter, agar dapat dilakukan pengambilan skor terbaik dari parameter yang telah ditentukan. (Ataei dan Osanloo, 2004).

Langkah-langkah umum metode ini adalah sebagai berikut:

1. Membagi *dataset* berdasarkan jumlah CV yang ditentukan
2. Menentukan *hyperparameter* yang akan dikerjakan oleh sistem, sering kali metode ini membutuhkan tambahan fungsi yang bernama *Pipeline* agar sistem dapat menjalankan beberapa fungsi secara bergantian dalam waktu proses yang sama, selain itu, pada *hyperparameter* juga dapat ditentukan

nilai apa yang ingin dilakukan kalkulasi (*scoring* yang digunakan pada penelitian ini adalah *F1_micro*) dan menentukan bagaimana sistem menjalankan metode ini (*n_jobs*).

3. Menghitung dan menampilkan skor terbaik berdasarkan *best parameter*.