BAB II

LANDASAN TEORI

2.1 Cyberbullying

Cyberbullying merupakan penyalahgunaan Internet untuk melecehkan, mengancam, mempermalukan, dan mengejek orang lain. *Cyberbullying* tidak membutuhkan pertemuan tatap muka, serta tanpa melibatkan kekuatan fisik (Yuda, 2013). Sebuah studi oleh Aliansi Anti-*Bullying* di Inggris menunjukkan bahwa 45% orang tua khawatir anak mereka diganggu secara *online*. Para orang tua khawatir karena dalam penelitian yang dilakukan oleh Aliansi Anti-*Bullying* di Inggris, lebih dari separuh remaja mengalami *cyberbullying*. Seperti halnya intimidasi tatap muka. Hal ini memalukan bagi korban dan kebanyakan dari korban tidak memberi tahu siapapun kapan hal tersebut terjadi. *Cyberbullying* meningkatkan anonimitas pelaku, berkat adanya layar digital antara pelaku dan korbannya (Lazuardi, 2018).

2.2 Sentiment Analysis

Sentiment analysis merupakan salah satu bidang dari *Natural Languange Processing* (NLP) yang membangun sistem untuk mengenali dan mengekstraksi opini dalam bentuk teks (Annisa, 2020). Informasi berbentuk teks saat ini banyak terdapat di Internet dalam format forum, blog, media sosial, serta situs berisi review. Dengan bantuan *sentiment analysis*, informasi yang tadinya tidak terstruktur dapat diubah menjadi data yang lebih terstruktur (Burhanudin, 2018). Proses dalam

sentiment analysis terbagi menjadi tiga yaitu memahami, mengekstrak, dan mengolah data teks secara otomatis sehingga menjadi suatu informasi yang bermanfaat.

Data tersebut dapat menjelaskan opini masyarakat mengenai produk, merek, layanan, politik, atau topik lainnya. Perusahaan, pemerintah, maupun bidang lainnya kemudian memanfaatkan data-data tersebut untuk membuat analisis marketing, review produk, umpan-balik produk, dan layanan masyarakat.

Guna menghasilkan opini yang dibutuhkan, *sentiment analysis* tidak hanya harus bisa mengenali opini dari teks. Proses yang juga disebut sebagai opini mining ini juga perlu bekerja dengan mengenali tiga aspek berikut:

1. Subjek: topik apa yang sedang dibicarakan.
2. Polaritas: apakah opini yang diberikan bersifat positif atau negatif.
3. Pemegang opini: seseorang yang mengeluarkan opini tersebut (Isa, 2017).

2.3 N-Gram

N-Gram adalah potongan N-karakter yang diambilkan dari suatu string (Permadi, 2008). Blank ditambahkan pada awal dan akhir suatu string untuk mengetahui batas awal dan akhir suatu string (Cavnar dan Trenkle, 1994). Sebagai contoh suatu string “TEXT” setelah ditambah awal dan akhir dengan “_” sebagai pengganti blank akan didapat N-*Gram* sebagai berikut :

Unigram : T,E,X,T

Bigram : _T, TE, EX, XT, T_

Trigram : _TE, TEX, EXT, XT_, T_ _

Model estimasi N-*gram* memberikan probabilitas kemungkinan pada kata berikutnya yang mungkin dapat digunakan untuk melakukan kemungkinan penggabungan pada keseluruhan kalimat (Hamzah, 2010).

2.4 TF-IDF

TF-IDF terbagi atas 2 kata yaitu *Term Frequency (TF)* dan *Inverse Document Frequency (IDF). TF* merupakan frekuensi dari kemunculan sebuah *term* dalam dokumen yang bersangkutan. Semakin besar jumlah kemunculan suatu *term,* semakin besar pula nilai kesesuaian yang dihasilkan. *IDF* sendiri merupakan sebuah perhitungan dari bagaimana sebuah *term* didistribusikan secara luas pada dokumen yang bersangkutan (Sierra, 2019). *IDF* menunjukkan hubungan ketersediaan sebuah *term* dalam sebuah dokumen. Semakin sedikit jumlah dokumen yang mengandung *term* yang dimaksud, nilai *IDF* akan semakin besar (Stecanella, 2019).

Cara menghitung *TF-IDF* adalah dengan mengkalikan nilai *TF* dan *IDF* dengan rumus sebagai berikut.

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D) \quad \ldots(1)$$

Fungsi tfidf(t,d,D) mengembalikan bobot *term* t pada sebuah dokumen d berdasarkan korpus (sekumpulan dokumen) D. Nilai tf(t,d) diperoleh dengan menggunakan persamaan sebagai berikut.

$$tf(t, d) = \log(1 + freq(t, d)) \quad \ldots(2)$$

Di mana *freq(t,d)* merupakan frekuensi *term*(t) pada *document(*d). Sedangkan nilai idf(t,D) diperoleh dengan menggunakan persamaan sebagai berikut.

$$idf(t, D) = \log(\frac{D}{df(t, D)}) \quad \ldots(3)$$

Variabel D merupakan jumlah semua dokumen dalam koleksi, sedangkan df merupakan jumlah dokumen yang mengandung *term* (t).

2.5 Naive Bayes Classifier

Teorema Bayes menemukan probabilitas suatu peristiwa terjadi mengingat probabilitas peristiwa lain yang telah terjadi. Metode ini dapat diasumsikan bahwa ada atau tidaknya suatu ciri tertentu dari suatu *class* tidak ada hubungannya dengan ciri dari *class* lainnya. Berikut persamaan umum teorema *Bayes* (Widianto, 2019).

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \quad \ldots(4)$$

Di mana:

B = Data dengan class yang belum diketahui.

A = Hipotesis data.

P(A|B) = Peluang hipotesis (A) berdasarkan kondisi (B).

P(A) = Peluang hipotesis (A).

P(B|A) = Peluang berdasarkan kondisi(B) pada hipotesis(A).

P(B) = Peluang B.

Rumus di atas menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam (A|B) (*Posterior*) adalah peluang munculnya (A) (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik karakteristik sampel pada kelas (B|A) (disebut juga *likelihood*), dibagi

dengan peluang kemunculan karakteristik sampel secara global (disebut juga *evidence*). Karena itulah rumus teorema Bayes dapat ditulis sebagai berikut

$$posterior = \frac{prior \times likelihood}{evidence} \qquad \ldots(5)$$

Nilai e*vidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai dari *posterior* tersebut nantinya akan dibandingkan dengan nilai nilai *posterior* kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Penjabaran lebih lanjut rumus Bayes tersebut dilakukan dengan menjabarkan (c|x1,…,xn). Penjabaran yang dimaksud tertulis sebagai berikut.

$$\begin{aligned} &P(C \mid X1,..., Xn) = P(C)(P(X1,..., Xn \mid C) \\ &= P(C)P(X1 \mid C)(X2,..., Xn \mid C, X1) \\ &= P(C)P(X1 \mid C)P(X2 \mid C, X1)(X3,..., Xn \mid C, X1, X2) \end{aligned} \qquad \ldots(6)$$

Penjabaran tersebut menyebabkan semakin banyak dan semakin kompleksnya syarat yang mempengaruhi nilai probabilitas, hampir mustahil untuk dianalisa satu persatu. Disinilah digunakan asumsi independensi yang sangat tinggi (naif), bahwa masing masing petunjuk independen satu sama lain. Hal ini umum disebut dengan *Naive Bayes Classifier.*

Naive Bayes Classifier merupakan sebuah metode klasifikasi yang berakar pada teorema Bayes. Metode pengklasifikasian dengan menggunakan metode probabilitas dan statistik yg dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya

sehingga dikenal sebagai Teorema Bayes. Ciri utama dari *Naive Bayes Classifier* ini adalah asumsi yg sangat kuat (naif) akan independensi dari masing-masing kondisi / kejadian (Hidayat, 2016).

Keuntungan pengguna adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*training data*) yg kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Karena yang diasumsikan sebagai variabel independen, maka hanya varians dari suatu variabel dalam sebuah kelas yang dibutuhkan untuk menentukan klasifikasi, bukan keseluruhan dari matriks kovarians (Gandhi, 2018). Secara matematis, rumus *Naive Bayes Classifier* adalah sebagai berikut.

$$P(c \mid X1,..., Xn) = P(C)\prod\nolimits_{i=1}^{n} P(Xi \mid C) \qquad \ldots(7)$$

Untuk menghindari nilai probabilitas nol, dilakukan proses *smoothing.* Cara kerjanya dengan setiap perhitungan data ditambah satu dan tidak akan membuat perbedaan yang berarti pada estimasi probabilitas sehingga bisa menghindari nilai probabilitas nol. *Smoothing* dapat dirumuskan sebagai berikut.

$$P(t_k \mid c) = \frac{1 + N_k}{|V| + N} \qquad \ldots(8)$$

Di mana N_k adalah jumlah kemunculan t_k dalam dokumen pelatihan di c dan N adalah jumlah total kemunculan kata dalam c.

2.6 Gaussian Naive Bayes

Gaussian Naive Bayes sendiri merupakan cara pengerjaan *Naive bayes classifier* dengan menggunakan distribusi *Gaussian*. Metode *Gaussian* sendiri hanya

membutuhkan estimasi rata-rata dan standar deviasi dari *training data. Gaussian Naive Bayes* adalah algoritma yang memiliki pendekatan secara probabilitas (Chakrabarty, 2019). Cara penghitungan yang dilakukan sebagai berikut.

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp^{-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}} \quad \ldots(9)$$

Keterangan:

σ = Standar deviasi

μ = Rata-rata / Mean

Untuk menghindari probabilitas nol kata muncul dalam dokumen, dibutuhkan proses *smoothing*. Proses *smoothing* dilakukan untuk mendapatkan probabilitas yang lebih akurat. Rumus untuk melakukan *smoothing* menggunakan *Laplace Smoothing* seperti pada Rumus 8.

2.7 Multinomial Naive Bayes

Multinomial Naive bayes adalah salah satu metode bayes yang dipakai dengan memperhitungkan frekuensi masing-masing kemunculan kata dalam sebuah dokumen dan probabilitas (Rennie, 2003). *Multinomial naive bayes* memperhitungkan distribusi tiap fitur dalam dokumen sehingga mendapat akurasi yang baik (Huang,2017). Perumusan *multinomial naive bayes* dilakukan sebagai berikut.

$$Cmap = \arg \max_{C \in \{Cl, Cs\}} P(c) \prod_{k=1}^{m} P(t_k \mid c) \quad \ldots(10)$$

Parameter P(tk|c) atau *probability likelihood* diestimasikan dengan menghitung kejadian tk pada semua dokumen latih di c dengan menggunakan *Laplacean Prior* seperti pada Rumus 8.