BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada tinjauan pustaka ini akan membahas beberapa penelitian tentang analisis yang sudah pernah dibuat sebelumnya. Yang memiliki kesamaan dalam system yang akan dibuat.

Akhmad Deviyanto dan M. Didik R. Wahyudi, 2018, melakukan penelitian tentang implementasi *K-Nearest Neighbor* pada analisis sentiment masyarakat terhadap kandidat Pilkada DKI 2017. Penelitian tersebut menggunakan medote *K-Nearest Neighbor* dengan nilai akurasi terbesar adalah 67,2% ketika k=5, presisi tertinggi 56,94% ketika k=5, dan recall 78,24% dengan k=15.

Yessivha Imanuela Claudy, Rizal Setya Perdana, dan M. Ali Fauzi, 2018, melakukan penelitian tentang klasifikasi dokumen Twitter untuk mengetahui karakter calon karyawan. Penelitian tersebut menggunakan metode algoritma *K-Nearest Neighbor* dengan nilai akurasi sebesar 66% untuk k=4. Hasil tersebut adalah hasil dimana 53 data uji yang benar dan 27 data uji yang salah.

Siti Ernawati dan Risa Wati, 2018, melakukan penelitian tentang penerapan algoritma *K-Nearest Neighbor* pada analisis sentiment review agen travel. Penelitian tersebut menggunakan metode *K-Nearest Neighbor* dengan akurasi mencapai 87% dan titik AUC adalah 0,916.

Billy Gunawan, Helen Sasty Pratiwi , dan Enda Esyudha Pratama, 2018 melakukan penelitian tentang analisis sentimen pada ulasan produk, analisis

sentimen yang dibangun menggunakan algoritma klasifikasi Naive Bayes, dengan nilai akurasi terendah pada pengujian 5 kelas menggunakan dataset 80% latih dan 20% data uji sebesar 52.66%, sedangkan pada pengujian 3 kelas menggunakan dataset 90% data latih dan 10% data uji memiliki akurasi tertinggi sebesar 77.78%..

Didik Garbian Nugroho, Yulison Herry Chrisnanto, dan Agung Wahana, 2016, melakukan penelitian jasa ojek online menggunakan metode Naïve Bayes, Hasil dari pengujian metode Naïve Bayes akurasi yang didapat sebesar 80%.

Tabel 2.1 Hasil Tinjauan Pustaka Peneliti Sebelumnya

No.	Peneliti	Tahun	Objek	Metode	Hasil
1	Akhmad Deviyanto, Muhammad Didik Rohman Wahyudi	2018	Sentimen masyarakat terhadap kandidat Pilkada DKI 2017	K-Nearest Neighbor	Sentimen positif dan negatif.
2	Yessivha Imanuela Claudy, Rizal Sertya Perdana, dan M. Ali Fauzi	2018	Tweet calon karyawan dari suatu perusahaan.	K-Nearest Neighbor	Klasifikasi kepribadian atau karakter calon karyawan.
3	Siti Ernawati, Risa Wati	2018	Sentimen ulasan agen perjalanan	K-Nearest Neighbor	Sentimen positif dan negative.
4	Billy Gunawan, Helen Sasty Pratiwi , Enda Esyudha Pratama	2018	Sistem Analisis Sentimen pada Ulasan Produk	Naïve Bayes	Sentiment positif, Negatif, dan Netral
5	Didik Garbian Nugroho, Yulison	2016	Analisis Sentimen	Naïve Bayes	Sentimen positif dan negative

	Herry Chrisnanto, Agung Wahana		Pada Jasa Ojek Online		
6	Usulan	2023	Analisis opini masyarakat terhadap proyek IKN	K-Nearest Neighbor dan Naïve Bayes	Sentimen positif, negative, dan netral

2.2 Dasar Teori

2.2.1 Twitter

Twitter adalah sebuah media sosial dan layanan microblogging yang mengijinkan penggunanya untuk mengirimkan pesan realtime. Pesan ini populer dengan sebutan tweet. Tweet adalah sebuah pesan pendek dengan panjang karakter yang dibatasi hanya sampai 140 karakter. Dikarenakan keterbatasan karakter yang bisa dituliskan, sebuah tweet seringkali mengandung singkatan, bahasa slang maupun kesalahan pengejaan (Agarwal et al., 2014). Untuk pengambilan data Twitter menyediakan Aplication Programming Interface (API). Ada dua jenis API yang dapat digunakan REST API dan Streaming API. REST API digunakan untuk mengakses status dan user timeline. Streaming API digunakan untuk mengakses kata kunci, hashtags, ID pengguna, dan lokasi.

2.2.2 Text Mining

Banyaknya informasi yang ada di dunia maya membuat upaya-upaya pengembangan terhadap penggalian informasi dari basis data daring semakin pesat, salah satunya text mining. Text mining, yang juga disebut sebagai Teks Data

Mining (TDM) atau Knowledge Discovery in Text (KDT), secara khusus dikembangkan untuk proses ekstraksi informasi dari dokumen-dokumen teks tak terstruktur (unstructured). Text mining memiliki definisi menambang data berupa teks di mana sumber data biasanya didapatkan dari dokumen dan tujuanya adalah untuk mencari kata-kata yang dapat mewakili isi dari dokumen sehingga dapat dilakukan analisis keterhubungan antar dokumen.

Text mining mencoba memecahkan masalah kelebihan informasi (information overload) dengan menggunakan teknik-teknik dari bidang ilmu yang terkait. Text mining dapat dipandang sebagai perluasan dari data mining atau Knowledge Discovery in Database (KDD), yang bertujuan untuk menemukan pola-pola menarik dari basis data berskala besar.

2.2.3 Pembobotan TF-IDF

Pembobotan Pembobotan atau *term weighting* merupakan proses mendapatkan nilai dari *term* yang berhasil diekstrak dari proses sebelumnya. Metode yang digunakan untuk pembobotan ini adalah *Term Frequency – Inverse Document Frequency* (TF-IDF).

1. *Term Frequency* (TF)

   Term Frequency merupakan jumlah kemunculan frekuensi kata pada suatu dokumen (Xia & Chai, 2011). *Term Frequency* ($tf_{t,d}$) didefinisikan jumlah kemunculan term t pada dokumen d. pembobotan menggunakan TF dijelaskan pada Persamaan

$$w_{tf_{t,d}} = \{0\ ,\ \ jika\ tf_{t,d} = 01 + \ tf_{t,d}\ \ ,\ \ jika\ tf_{t,d} > 0 \text{ .....(1)}$$

Keterangan:

$$w_{tf_{t,d}} = Hasil\ pembobotan\ tf_{t,d}$$

$$tf_{t,d} = Banyaknya\ kemunculan\ kata\ t\ dalam\ dokumen\ d$$

2. *Invers Document Frequency* (IDF)

Invers Document Frequency merupakan frekuensi kemunculan term pada keseluruhan dokumen teks. *Term* yang jarang muncul pada keseluruhan dokumen teks memiliki nilai *Invers Document Frequency* lebih besar dibandingkan dengan *term* yang sering muncul (Rahmawati, Sihwi, & Suryanti, 2014). Pembobotan menggunakan *Invers Document Frequency* (IDF) dijelaskan pada Persamaan

$$idf_t = \ log\left(\frac{N}{df_{(t)}}\right) \quad \text{.....(2)}$$

Keterangan :

idf_t = Hasil *inverse* dari df_t

N = jumlah dokumen teks

$df_{(t)}$ = jumlah dokumen yang mengandung *term t.*

3. *Term Frequency – Invers Document Frequency* (TF-IDF)

Nilai tf-idf dari sebuah kata merupakan kombinasi dari nilai tf dan nilai idf dalam perhitungan bobot. Pembobotan TF-IDF dijelaskan pada Persamaan

$$W_{tf} = W_{tf\ \ t,d} * idf_t$$

$$= \ w_{tf_{t,d}} * \frac{N}{df_t} \text{ .....(3)}$$

Keterangan :

$W_{t,f}$ = pembobotan TF-IDF

W_{tf} $_{t,d}$ = $Hasil\ pembobotan\ tf_{t,d}$.

Idf_t = *Invers Document Frequency.*

N = jumlah dokumen teks

2.2.4 Analisis Sentimen

Analisis sentiment adalah sebuah teknik atau cara yang digunakan untuk mengidentifikasi bagaimana sebuah sentiment diekspresikan menggunakan teks dan bagaimana sentiment tersebut bisa dikategorikan sebagai sentiment positif maupun sentiment negative (Nasukawa & Yi, 2013). Pendapat yang hampir senada dikemukakan oleh (Cvijikj & Michahelles, 2011), dimana analisis sentiment digunakan untuk memahami komnetar yang diciptakan oleh pengguna internet dan menjelaskan bagaimana sebuah produk maupun *brand* diterima oleh mereka. Definisi analisis sentiment twitter sendiri merupakan bagian dari pendapat pada media twitter.

Pesan twitter, pada kenyataannya, lebih mudah untuk menganalisis karena penulisan yang dibatasi dibanding forum diskusi. Hal ini berbeda pada forum diskusi yang lebih sulit, dikarenakan pengguna dapat mendiskusikan apapun dan berinteraksi satu sama lain. Kalimat seringkali memuat pendapat tunggal, meskipun tidak bersifat mutlak bahwa setiap kalimat berisi pendapat tunggal. Dalam kasus lain terdapat kalimat dengan pendapat lebih dari satu pada suatu kalimat namun ini hanya sebagian kecil (Liu, 2012).

Pada dasarnya sentimen analisis merupakan tahapan klasifikasi. Namun tahapan klasifikasi sentiment pada twitter (tidak terstruktur) sedikit lebih sulit

disbanding dengan klasifikasi dokumen terstruktur. Dalam kasus analisis sentimen twitter yang merupakan gambaran dari kalimat, langkah pertama (Liu, 2012) adalah untuk mengklasifikasikan apakah kalimat mengungkapkan pendapat atau tidak. Langkah kedua adalah mengklasifikan kalimat-kalimat pendapat menjadi positif dan kelas negatif.

2.2.5 Confusion Matrix

Confusion matrix adalah tabel yang berfungsi sebagai perbandingan kategori aktual dengan kategori prediksi (Nathania, Indriarti dan Bachtiar, 2018).

Tabel 2.2 Confusion Matrix

		Hasil Aktual	
		N	P
Hasil Prediksi	N	TP	FP
	P	FN	TN

Keterangan :

- *True Positive* (TP), yaitu sejumlah data yang benar sesuai kelas setiap kategori diprediksi benar oleh sistem.
- *True Negative* (TN), yaitu sejumlah banyaknya data yang salah sesuai kelas setiap kategori dipresdiksi benar oleh sistem.
- *False Positive* (FP), yaitu sejumlah banyaknya data yang salah kelas kategori diprediksi salah oleh sistem.
- *False Negative* (FN), yaitu sejumlah banyaknya data yang benar kelas kategori diprediksi salah oleh sistem.

Precision adalah tingkat keakuratan untuk mengetahui hasil kategori data yang diklasifikasikan sesuai dengan kategori sebenarnya. Persamaan *precision* dapat dilihat pada Persamaaan

$$Precision = \frac{TP}{TP+FP} \quad \text{.....(4)}$$

Recall adalah parameter untuk mengetahui tingkat keberhasilan sistem untuk mengenali sebuah kategori. Persamaan *recall* dapat dilihat pada Persamaaan

$$Recall = \frac{TP}{TP+FP} \quad \text{.....(5)}$$

F-measure merupakan gambaran pengaruh antara *precision* dan *recall* (Puspitasari, Santoso, Indriarti, 2018). Persamaan *F-measure* dapat dilihat pada Persamaaan

$$F = \frac{2 \times recall \times precision}{recall + precision} \quad \text{.....(6)}$$

Akurasi merupakan tahapan membandingkan nilai proses prediksi yang sudah dihitung pada pengujian nilai aktual. Persamaan akurasi dapat dilihat pada Persamaaan

$$\text{akurasi} = \frac{TP+TN}{TP+FN+FP+TN} \times 100 \quad \text{.....(11)}$$

2.2.6 K-Nearest Neighbor

K-Nearest Neighbor (KNN) adalah salah satu metode paling sederhana untuk memecahkan masalah klasifikasi (Wei, & Yongquan, 2016). Algoritma ini sering digunakan untuk klasifikasi teks dan data (Delima, & Rachmat, 2014). Pada metode ini dilakukan klasifikasi terhadap obyek berdasarkan data yang jaraknya paling dekat dengan obyek tersebut (Hardiyanto & Rahutomo, 2016).

Klasifikasi teks menggunakan metode KNN akan menghasilkan nilai yang lebih optimal jika menggunakan rumus *cosine similarity* untuk pembobotan tiap-tiap kata pada dokumen teks yang akan diproses. Untuk persamaan dari *cosine similarity* dapat dilihat pada Persamaan 2.8

$$CosSim(q, d_j) = \frac{d_j.q}{|d_j|.|q|} = \frac{\sum_{i=1}^{t} (w_{ij}.w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} . \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \quad \text{..... (12)}$$

Keterangan :

CosSim(q,d_j) : Nilai kemiripan antara dokumen uji (q) dengan dokumen latih ke j (d_j)

t : Jumlah *term* (kata)

d : Dokumen

q : *Query* (kata kunci)

w_{ij} : Bobot *term* (kata) ke i pada dok.latih j

W_{iq} : Bobot *term* (kata) ke i pada dok.uji q

Setelah hasil nilai *cosine similarity (CosSim)* didapatkan, maka hasil perhitungannya akan diurutkan secara menurun untuk setiap kategori. Setelah itu dilakukan penentuan nilai k dan selanjutnya akan dilakukan perhitungan untuk mendapatkan nilai k baru (n). Nilai k baru (n) dapat dihitung menggunakan persamaan

$$n = \frac{k * N(cm)}{Maks\,[N(cm)]j=1...Nc} \quad \text{.....(13)}$$

Keterangan :

N : Nilai k baru

K : Nilai k yang ditetapkan

N(cm) : Jumlah dokumen latih dikategori/ kategori m

Maks[N(cm)]j=1…Nc : Jumlah dokumen latih terbanyak pada semua kategori

Kemudian dilakukan perhitungan terhadap peluang dari dokumen uji X termasuk dengan dokumen latih dj sebanyak nilai n tetangga untuk setiap kategori pada dokumen X pada dokumen latih dj sebanyak nilai n tetangga untuk *training set*. Persamaan 2.10 dapat digunakan untuk menghitung peluang dari dokumen uji X pada kategori m.

$$p(x, c_m) = argmax_m \frac{\sum_{d_j \in top\_n\_kNN_{cm}} sim(x,d_j) y(d_j.c_m)}{\sum_{d_j \in top\_n\_kNN_{cm}} sim(x,d_j)} \quad (14)$$

Keterangan :

p(x,cm) : Probabilitas dokumen X menjadi anggota kategori cm

sim(x,dj) : Kemiripan antara dokumen X dengan dokumen latih dj

top_n_Knn : Top n tetanggga

y(dj.cm) : Fungsi atribut yang memenuhi pesamaan.

2.2.7 Naïve Bayes

Algoritma *Naïve Bayes* merupakan salah satu pengklasifikasi statistik, dimana pengklasifikasi ini dapat memprediksi probabilitas keanggotaan kelas suatu data yang akan masuk ke dalam kelas tertentu, sesuai dengan perhitungan probabilitas. Pengklasifikasi bayes didasari oleh *theorema bayes* yang ditemukan oleh Thomas Bayes pada abad ke-18. *Teorema Bayes* adalah teorema yang digunakan dalam statistika untuk

menghitung peluang suatu hipotesis. Bayes merupakan teknik prediksi berbasis *probalistic* sederhana yang berdasar pada penerapan *theorema Bayes* dengan asumsi independensi yang kuat (Eko Prasetyo, 2012). Definisi lain mengatakan *Naive Bayes* merupakan pengklasifikasian dengan metode probabilistik dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan dengan pengalaman di masa lalu (Alfa, 2015).

Keunggulan Naïve Bayes adalah sifatnya yang efektif dan cepat untuk mengolah data berjumlah besar. Karena kelebihannya itu, Naïve Bayes biasa digunakan untuk aplikasi seperti *spam filtering* (pendeteksi pesan sampah) dan deteksi anomali di jaringan komputer. Algoritma Naïve Bayes bahkan dianggap sebagai standar *de facto* untuk penerapan klasifikasi teks, misalnya sentiment analysis (menentukan apakah penulis suatu komentar bernada positif, negatif atau netral).

Dalam penelitian ini yang menjadi data uji adalah data headline dari berita yang diunggah dan komentar dari setiap *tweets*. Ada dua tahap proses pada klasifikasi dokumen. Tahap pertama adalah pelatihan terhadap dokumen yang sudah diketahui kategorinya, selanjutnya pada tahap dua adalah proses klasifikasi data yang belum diketahui kategorinya.(Dios Kurniawan, 2021).

Naïve Bayes merupakan salah satu contoh metode *supervised document classification* yang membutuhkan data latih dalam melakukan klasifikasi. Selanjutnya yang perlu dilakukan adalah menghitung nilai

probabilitas masing-masing katanya dengan persamaan 1.

Rumus untuk Naive bayes classifier ada pada persamaan berikut:

$P(C_i) = \frac{f_d(c_i)}{|d|}$ …….(15)

Keterangan:

$P(ci)$ = Probabilitas ci yang merupakan kategori kelas

$fd(ci)$ = Jumlah dokumen ci

$|D|$ = Jumlah data latih / dokumen

Setelah mendapatkan probabilitas dari setiap kelas, selanjutnya yaitu menghitung probabilitas dari setiapfitur pada setiap kelas dengan persamaan:

$$P(w_k|c_i) = \frac{f(w_{ki},C)+1}{P(C_i)+|W|} \quad \text{…….(16)}$$

Keterangan:

$P(wk|ci)$ = Peluang kemunculan kata ada sebuah kelas, wk adalah kata yang muncul pada sebuah kategori

$f(wki, C)$ = Nilai kemunculan kata wki pada kelas ci

$P(Ci)$ = Jumlah keseluruhan kemunculan kata pada kelas ci

$|W|$ = Jumlah data latih/dokumen

Kemudian langkah selanjutnya adalah menentukan probabilitas data uji dari setiap kelas berdasarkan dari proses pembelajaran. Nilai probabilitas yang paling tinggi akan terpilih:

$$Vmap = \underset{\{kelas\,0, kelas\,1\}}{argmax} \prod_{i=1}^{n} P(w_k|c_i)\; x \;\; P(c_i) \quad (17)$$

Keterangan:

$P(wk|ci)$ = Probabilitas kemunculan kata-kata pada sebuah kelas,

wk = Merupakan kata yang muncul pada sebuah kategori

$P(ci)$ = Menentukan probabilitas ci yaitu kategori kelas