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 sistem yang akan dibuat. Beberapa penelitian yang terkait dengan penggunaan metode *K-Nearest Neighb*or adalah sebagai berikut.

Penelitian analisis sentimen dilakukan oleh Ody Andrian (2014), yaitu melakukan penelitian membuat aplikasi sistem pendukung keputusan yang dapat memberikan rekomendasi pembelian laptop. Aplikasi tersebut dapat memberikan rekomendasi pemilihan laptop dari perhitungan manual dan hasil dari aplikasi.

Selanjutnya Yessivha, dkk., (2018), melakukan penelitian tentang klasifikasi dokumen Twitter untuk mengetahui karakter calon karyawan. Penelitian tersebut menghasilkan nilai akurasi sebesar 66% untuk k=4. Hasil tersebut adalah hasil dimana 53 data uji yang benar dan 27 data uji yang salah.

Penelitian lainnya dilakukan oleh Siti Ernawati dan Risa Wati (2018), yaitu melakukan penelitian tentang penerapan algoritma *K-Nearest Neighbor* pada analisis sentiment review agen travel menggunakan 100 *review* positif dan 100 *review* negatif dengan akurasi mencapai 87% dan titik AUC adalah 0,916.

Selanjutnya Akhmad Deviyanto dan M. Didik R. Wahyudi (2018), melakukan penelitian tentang implementasi *K-Nearest Neighbor* pada analisis

sentiment masyarakat terhadap kandidat Pilkada DKI 2017 dengan nilai akurasi terbesar adalah 67,2% ketika k=5, presisi tertinggi 56,94% ketika k=5, dan recall 78,24% dengan k=15.

Dalam penelitian Dwi Suci Ariska Yanti, dkk., (2019), meneliti tentang kebijakan ganjil genap kendaraaan bermotor di DKI Jakarta. Berdasarkan hasil pengujian, penelitian ini memiliki nilai *F-Measure* tertinggi sebesar 66,1% dan nilai akurasi sebesar 66,5%.

Tabel 2. 1 Tabel Perbandingan Tinjauan Pustaka

No	Nama Peneliti	Tahun	Objek	Hasil
1	Ody Andrian	2016	Jenis Laptop yang ada pada CV. Buana Jaya Komputer Yogyakarta	Rekomendasi laptop berdasarkan masukan kriteria/atribut dari konsumen.
2	Yessivha Imanuela Claudy, Rizal Setya Perdana, dan M. Ali Fauzi	2018	Tweet calon karyawan dari suatu perusahaan	Klasifikasi kepribadian atau karakter calon karyawan.
3	Siti Ernawati, Risa Wati	2018	Sentimen ulasan agen perjalanan	Sentimen positif dan negative.
4	Akhmad Deviyanto, Muhammad Didik Rohman Wahyudi	2018	Sentimen masyarakat terhadap kandidat Pilkada DKI 2017	Sentimen positif dan negative.
5	Dwi Suci Ariska Yanti, Indrianti, Putra Pandu Adikara	2019	Sentimen masyarakat tentang kebijakan ganjil genap kendaraan bermotor di DKI Jakarta	Sentimen positif dan negative.
6	Tiara Rahmadhani (2021)	2021	Opini masyarakat terhadap acara TV	Sentimen negatif, pososstif, dan metral pada tayangan televisi.

2.2 Dasar Teori

2.2.1 Twitter

Twitter adalah situs jejaring sosial besar yang mengarah pada komunikasi cepat. Lebih dari 140 juta pengguna aktif mempublikasikan lebih dari 400 juta karakter "*tweets*" setiap hari. Twitter telah menjadi media komunikasi yang penting bagi semua lapisan masyarakat. Twitter telah berperan penting dalam berbagai peristiwa sosial dan politik (Morstatter dan Liu, 2013). 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

Text Mining adalah proses mengambil informasi dari teks. Informasi biasanya diperoleh melalui peramalan pola dan kecenderungan pembelajaran pola statistik. *Text Mining* yaitu parsing, bersama dengan panambahan beberapa fitur linguistic turunan dan penghilangan beberapa diantaranya, dan penyisipan *subsequenst* ke dalam database, menentukan pola dalam data terstruktur, dan akhirnya mengevaluasi dan menginterpretasi output, *text mining* biasanya mengacu ke beberapa kombinasi relevansi, kebaruan, dan *interestingness*.

Kunci dari proses pada *text mining* adalah menggabungkan informasi yang berhasil diekstraksi dari berbagai sumber (Hearst, 2003). Sedangkan menurut (Harlian, 2006) *text mining* didefinisikan sebagai kata yang berupa text yang

biasanya sumber data didapatkan dari dokumen, dengan tujuan adalah mencari kata-kata yang dapat mewakili isi dari dokumen tersebut yang nantinya dapat dilakukan analisa hubungan antar dokumen. Proses *text mining* yang khas meliputi kategorisasi teks, *text clustering*, ektraksi konsep/entitas, produksi taksonomi granular, *sentiment analysis*, penyimpulan dokumen, dan pemodelan relasi entitas yaitu, pembelajaran hubungan antara entitas (Bridge, 2011).

Pendekatan manual *text mining* secara intensif dalam laboratorium pertama muncul pada pertengahan 1980-an, namun kemajuan teknologi telah memungkinkan ranah tersebut untuk berkembang selama decade terakhir. *Text mining* adalah bidang interdisipliner yang mengacu pada pencarian informasi, pertambangan data, pembelajaran mesin, statistik, dan komputasi linguistic. Dikarenakan kebanyakan informasi (perkiraa umum mengatakan lehih dari 80%) saat ini disimpan sebagai teks, *text mining* diyakini memiliki potensi nilai komersial tinggi (Bridge, 2011).

2.2.3 Algoritma Nazief & Adriani

Algoritma Nazief & Adriani menyimpulkan sebuah kata dasar dapat ditambahkan imbuhan berupa *derivation prefix* (DP) di awal dan/atau diakhiri secara berurutan oleh *derivation suffix* (DS), *possesive pronoun* (PP), dan *particle* (P). Perhitungan algoritma Nazief & Adriani dapat dilihat pada persamaan 2.1

DP + DP + DP + *root word* + DS + PP + P 2.1

Adapun langkah-langkah yang digunakan oleh algoritma Nazief dan Adriani yaitu sebagai berikut: (Nazief & Adriani, 1996)

1. Kata dicari di dalam daftar kamus. Bila kata tersebut ditemukan di dalam kamus, maka dapat diasumsikan kata tersebut adalah kata dasar sehingga algoritma dihentikan.
2. Bila kata di dalam langkah pertama tidak ditemukan di dalam kamus, maka diperiksa apakah sufiks tersebut yaitu sebuah partikel (“-lah” atau “-kah”). Bila ditemukan, maka partikel tersebut dihilangkan.
3. Pemeriksaan dilanjutkan pada kata ganti milik (“-ku”, “-mu”, “-nya”). Bila ditemukan, maka kata ganti tersebut dihilangkan.
4. Memeriksa akhiran (“-i”, “-an”). Bila ditemukan, maka akhiran tersebut dihilangkan. Hingga langkah ke-4 dibutuhkan ketelitian untuk memeriksa apakah akhiran “-an” merupakan hanya bagian dari akhiran “-kan”, dan memeriksa lagi apakah partikel (“-lah”, “-kah”) dan kata ganti milik (“-ku”, “-mu”, “-nya”) yang telah dihilangkan pada langkah 2 dan 3 bukan merupakan bagian dari kata dasar.
5. Memeriksa awalan (“se-“, ”ke-“, “di-“, “te-“, “be-“, “pe-“, “me-“). Bila ditemukan, maka awalan tersebut dihilangkan. Pemeriksaan dilakukan dengan berulang mengingat adanya kemungkinan multi-prefix.Langkah ke-5 ini juga membutuhkan ketelitian untuk memeriksa kemungkinan peluluhan awalan, perubahan prefix yang disesuaikan dengan huruf-awal kata, dan aturan kombinasi prefix-suffix yang diperbolehkan.

6. Setelah menyelesaikan semua langkah dengan sukses, maka algoritma akan mengembalikan kata dasar yang ditemukan.

2.2.4 Pembobotan TF-IDF

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 2.2.

$$w_{tf_{t,d}} = \begin{cases} 1+\log_{10} tf_{t,d} & , \ jika\ tf_{t,d}>0 \\ 0 & , \ jika\ tf_{t,d}=0 \end{cases} \qquad 2.2$$

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. Pembobotan menggunakan *Invers Document Frequency* (IDF) dijelaskan pada Persamaan 2.3.

$$idf_t = log\left(\frac{N}{df_{(t)}}\right) \quad 2.3$$

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 2.4

$$W_{tf} = W_{\substack{tf \\ t,d}} * idf_t \quad 2.4$$

$$= w_{tf_{t,d}} * \log_{10}\frac{N}{df_t}$$

Keterangan :

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

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

Idf_t = *Invers Document Frequency.*

N = jumlah dokumen teks

2.2.5 Analisis sentimen

Analisis sentiment adalah sebuah teknik atau cara yang digunakan untuk mengidentifikasi bagaimana sebuah sentimen diekspresikan menggunakan teks dan bagaimana sentimen tersebut bisa dikategorikan sebagai sentimen positif maupun sentimen 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.

2.2.6 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 2.5

$$Precision = \frac{TP}{TP+FP} \quad 2.5$$

Recall adalah parameter untuk mengetahui tingkat keberhasilan sistem untuk mengenali sebuah kategori. Persamaan *recall* dapat dilihat pada Persamaaan 2.6

$$Recall = \frac{TP}{TP+FP} \quad 2.6$$

F-measure merupakan gambaran pengaruh antara *precision* dan *recall* (Puspitasari, Santoso, Indriarti, 2018). Persamaan *F-measure* dapat dilihat pada Persamaaan 2.7

$$F = \frac{2 \times recall \times precision}{recall + precision} \quad 2.7$$

Akurasi merupakan tahapan membandingkan nilai proses prediksi yang sudah dihitung pada pengujian nilai aktual. Persamaan akurasi dapat dilihat pada Persamaaan 2.8

$$\text{akurasi} = \frac{TP+TN}{TP+FN+FP+TN} \times 100 \quad 2.8$$

2.2.7 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 2.9

$$CosSim(q, d_j) = \frac{d_j . q}{|d_j| . |q|} = \frac{\sum_{i=1}^{t} (w_{ij} . w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} . \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \quad 2.9$$

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-value.

Penentuan k-values yang tepat diperlukan agar didapatkan akurasi yang tinggi dalam proses kategorisasi dokumen uji. Algoritma Improved k-Nearest Neighbor melakukan modifikasi dalam penentuan k-values. Dimana penetapan k-values tetap dilakukan, hanya saja tiap-tiap kategori memiliki k-values yang berbeda. Perbedaan k-values yang dimiliki pada setiap kategori disesuaikan dengan besar-kecilnya jumlah dokumen latih yang dimiliki kategori tersebut. Sehingga ketika k-values semakin tinggi, hasil kategori tidak terpengaruh pada kategori yang memiliki jumlah dokumen latih yang lebih besar. Nilai k-value baru (n) dapat dihitung menggunakan persamaan 2.10

$$n = \frac{k * N(cm)}{Maks\ [N(cm)]j=1 \ldots Nc} \qquad 2.10$$

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 2.11 dapat digunakan untuk menghitung peluang dari dokumen uji X pada kategori m.

$$p(x, c_m) = argmax_m \frac{\sum_{d_j \in top_n_kNN_{cm}} sim(x,d_j) y(d_j.c_m)}{\sum_{d_j \in top_n_kNN_{cm}} sim(x,d_j)} \quad 2.11$$

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.