BAB 2
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada tinjauan pustaka ini akan membahasan beberapa referensi penelitian tentang analisis yang sudah pernah dibuat sebelumnya yang memiliki kesamaan dalam system yang akan dibuat. Referensi penelitian ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Tinjauan Pustaka

No	Nama Peneliti	Metode	Objek	Hasil Keluaran
1.	Christoffer Sitepu (2019)	Pendekatan Yurisdis Normatif dan Pendekatan Yurisdis Empiris	Praktik pungutan liar yang dilakukan oleh polisi lalu lintas	Menunjukkan bahwa pelaksanaan E-Tilang di wilayah Jakarta Selatan terlaksana dengan baikkarena sudah lebih dominan digunakan system E-Tilang ini untuk penyelesaian perkara pelanggaran lalu lintas.
2.	Abdul Malik Zuhdi, Ema Utami, Suwanto Raharjo (2019)	K-NN	Capres Indonesia 2019	Hasil analisis bernilai positif,netral dan negative dengan akurasi 83.3%
3.	Muhammad Syarifuddin (2020)	*Decision Tree, K-NN, Naïve Bayes*	Opini public terhadap efek PSBB	Hasil analisis menggunakan metode K-NN menunjukkan akurasi 80.8 %

4.	Syahfitri Kartika Lidya, Opim Salim Sitompul, Syahril Efendi (2015)	*Support Vector Machine* (SVM) dan K-*Nearest Neighbor* (K-NN)	Artikel berita diwebsite yang berbahasa Indonesia	Kesimpulan pengaruh nilai k pada K-*fold cross validation* dan k pada K-NN
5.	Akhmad Deviyanto dan M. Didik R. Wahyudi (2018)	K-*Nearest Neighbor* (K-NN)	Pemilihan gubernur Jakarta (PILKADA DKI) 2017	Sentimen positif dan negative.
6	Andriantika Dewi Kusuma (2024)	K-*Nearest Neighbor* (K-NN)	Penerapan E-Tilang di Indonesia	Sentimen Negatif, netral, dan Positif

Penelitian Crisstoffer Sitepu (2019), membuat analisis pelaksanaan E-Tilang dalam upaya pencegahan praktik pungutan liar yang dilakukan oleh polisi lalu lintas. Penelitian tersebut menggunakan metode analisis kualitatif dan penaikan kesimpulan yang dilakukan secara induktif, yaitu menarik kesimpulan berdasarkan hal – hal yang besifat khusus dengan permasalahan yang akan diteliti.

Penelitian Abdul Malik Zuhdi, Ema Utami, Suwanto Raharjo (2019), membuat analisis sentiment terhadap capres Indnonesia 2019 dengan metode K-NN. Penelitian ini menggunakan metode K-NN dengan menggunakan penggabungan pembobotan tekstual dan pembobotan nontekstual untuk meningkatkan akurasi. Akurasi ketika menggunakan pembobotan tekstual adalah 82,50%, ketika menggunakan pembobotan nontekstual adalah 60%, sedangkan penggabungan keduanya adalah 83.3 %.

Penelitian Muhammad Syarifuddin (2020), membuat analisis sentiment opini public terhadap efek PSBB pada twitter dengan *algoritma Decision tree*, K-NN, *Naïve Bayes*. Penelitian ini menggunakan metode K-NN dengan akurasi 80.80%, precision 82.72%, dan recall 74.41% ketika menggunakan formula. Ketika menggunakan aplikasi Rapidminer menghasilkan akurasi 80,80%, precision 80%, dan recall 74,41%.

Penelitan Syahfitri Kartika Lidya, Opim Salim Sitompul, Syahril Efendi (2015), membuat *sentiment analisys* pada teks Bahasa Indonesia menggunakan *Support Vector Machine* (SVM) dan K-*Nearest Neighbor* (K-NN). Peneliti ini menganalisis artikel berita berbahasa Indonesia diweb menggunakan nilai k yang menghasilkan tingkat akurasi yang berbeda. Nilai k yang terlalu kecil akan menghasilkan nilai akurasi yang rendah, sedangkan nilai k yang telalu besar menghasilkan nilai akurasi yang besar.

Penelitian Akhmad Deviyanto dan M. Didik R. Wahyudi (2018), membuat analisis sentiment pengguna twitter tentang topik Pilkada DKI 2017 menggunakan algoritma *KNN (K-Nearest Neighbor).* Dari hasil pengujian diketahui bahwa nilali akurasi terbesar adalah 67,2% ketika k=5.

Peneliti Andriantika Dewi K (2024), Membuat sentiment analisis terhadap penerapan E-Tilang di Indonesia menggunakan algoritma K-Nearest Neighbor dengan akurasi tertinggi dengan nilai k =3 yaitu 78%. Sedangkan nilai presisi tertinggi yaitu 72% dan nilai recall tertinggi yaitu 62%. Hasil analisis sentimen dari pemberlakuan system e-tilang di Indonesia mendapatkan respon positif sebesar 68%.

2.2 Dasar Teori

2.2.1 Analisis Sentimen

Analisis sentimen atau juga dikenal sebagai *opini mining* atau *emotion artificial intelligence* merupakan penggunaan pemrosesan bahasa alami, analisis teks, komputasi *linguistic* dan *biometric* untuk mengidentifikasi, mengesktrak, menghitung dan mempelajari suatu informasi subjektif secara sistematis (Rizal, 2017). 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, 2003)

Analisis sentiment adalah sebuah topik untuk mendeteksi opini terhadap suatu subyek (misalnya individu, organissi ataupun produk) dalam sebuah kumpulan data (Nusukawa & Yi, 2003). Analisis sentimen digunakan untuk memahami komentar yang diciptakan oleh pengguna internet dan menjelaskan bagaimana sebuah produk maupun brand diterima oleh mereka (Cvijikj, 2013). Dari beberapa pendapat diatas bisa diambil kesimpulan bahwa analisis sentiment adalah sebuah proses untuk menentukan sentiment atau opini dari seseorang yang diwujudkan dalam bentuk teks dan bisa dikategorikan sebagai sentiment positif atau negative (Abdul M Z, Ema U & Suwanto R, 2019).

2.2.2 Text Mining

Text Mining merupakan proses untuk melakukan penambangan data dari dokumen atau data-data yang tidak terstruktur. *Text mining* berusaha untuk menghasilkan infomasi yang tersirat secara implisit atas informasi yang dengan otomatis diekstrak daari dokumen (Feldman, 2007). Data yang dikumpulkan kemudian dilakukan dilakukan *preprocessing* untuk menghindari data yang belum siap untuk diolah seperti terdapat gangguan (*noise*) dan data yang tidak konsisten (Hemalatha, Varma, & Govardhan, 2012). *Preprocessing* merupakan poses yang menjadikan data teks yang sudah didapat menjadi lebih teratur dan mudah untuk melalui tahap klasifikasi (Aris Tri, 2015). Adapun tahap *preprocessing* ini adalah sebagai berikut:

- *Text Preprocessing*

a. Case folding

Case folding merupakan tahapan mengubah semua huruf yang ada didalam dokumen menjadi huruf kecil.

b. Cleansing

Cleansing merupakan tahapan untuk membersihkan dokumen dari kata-kata yang tidak diperlukan untuk mengurangi *noise* (Noviah & Edi, 2014). Ada beberapa tahap *cleansing* yang diperlukan terhadap sebuah dokumen yang dijelaskan sebagai berikut:

- *Cleansing URL* yaitu proses pembersihan *url* atau *link* yang mengarah pada situs lainnya.

- *Cleansing mention*. *Mention* meupakan cara untuk memanggil akun lain atau membalas pesan tehadap akun yang di *mention*, menghapus *mention* diperlukan untuk mengurangi *tweet* yang berulang akibat penggunaan *mention* tersebut.
- *Cleansing hashtag*. *Hashtag* adalah tanda (#) untuk mengumpulkan topik tertentu berdasarkan pada *hashtag* tersebut.
- *Cleansing delimiter* yaitu proses menghilangkan karakter tanda baca yang tidak berguna seperti titik (.), koma (,), spasi, dll.
- *Cleansing emoticon*. Penggunaan *emoticon* akan mengganggu proses analisis sentiment karena seseorang salah atau kurang tepat dalam penggunaan *emoticon* (Buntoro et al, 2014).

c. Tokenizing

Tokenizing merupakan poses penghilangan tanda baca pada kalimat yang ada dalam dokumen sehingga menghasilkan kata-kata ynag berdiri masing-masing (Karyono & Utomo, 2012).

- *Feature Selection*

Feature pada *text mining* adalah berupa kumpulan kata atau frase yang bermakna sebagai kata opini bisa berupa opini positif ataupun opini negative (Haddi et al, 2013). Untuk mempermudah dan meningkatkan akurasi sentiment pada sebuah kalimat perlu dilakukan 2 tahap yaitu *stopword removal* dan *stemming*.

a. Stopword removal

Stopword removal meupakan proses menghilangkan kata yang memiliki pengaruh sangat kecil atau tidak penting didalam text. Pada tahap ini kata-kata harus dihapus untuk memudahkan tugas selanjutnya dan mempercepat proses *text processing* (Raulji & Saini, 2016). Dalam Bahasa Indonesia, misalnya kata : dan, atau, mungkin, ini, itu, dll.

b. Stemming

Stemming merupakan proses pemetaan dan penguraian bentuk (*variants*) dari suatu kata menjadi bentuk dasarnya (*stem*) (Kurniawan et al, 2017). Tujuan dari proses *stemming* adalah menghilangkan imbuhan berupa prefiks, sufiks maupun konfiks yang ada pada setiap kalimat.

2.2.3 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) dan *Term Weighting* (Wtf)

$TF_{t,d}$ merupakan banyaknya kemunculan *terms*/token/kata *t* dalam dokumen *d*. TF juga dapat disebut dengan istilah frekuensi dalam satu dokumen (Xia & Chai, 2011). Sedangkan Wtf merupakan proses perhitungan atau pembobotan pada setiap

terms/token/kata *t*. Persamaan 2.1 merupakan persamaan perhitungan pembobotan dari TF:

$$w_{tf_{t,d}} = \begin{cases} 1+\log_{10} tf_{t,d} & , \ jika\ tf_{t,d}>0 \\ 0 & , \ jika\ tf_{t,d}=0 \end{cases} \qquad \text{Persamaan (2.1)}$$

Keterangan:

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

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

2. *Document Frequency* (DF_t) dan *Inverse Document Frequency* (IDF_t)

DF_t merupakan banyaknya dokumen ynag mengandung *terms*/token/kata *t*. Sedangkan IDF_t merupakan hasil *inverse* dari DF_t. Persamaan 2.2 merupakan persamaan untuk menghitung IDF_t yang diusulkan oleh Jones(1972) :

$$idf_t = \log_{10} \frac{N}{df_t} \qquad \text{Persamaan (2.2)}$$

Keterangan :

idf_t = Hasil *inverse* dari df_t

df_t = Banyaknya dookumen yang mengandung kata *t*

N = Banyaknya dokumen atau koleksi yang ada

3. TF-IDF *weighting*

Pembobotan TF-IDF dari suatu *terms*/token/kata *t* merupakan perkalian dari TF *weight* dengan IDF_t (Manning, Raghavan dan Schutze, 2009). Persamaan 2.3 adalah persamaan menghitung pembobotan TF-IDF.

$$w_{t,d} = w_{tf_{t,d}} * idf_t$$

$$= w_{tf_{t,d}} * \log_{10} \frac{N}{df_t} \qquad \text{Persamaan (2.3)}$$

Keterangan:

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

N = Banyaknya dokumen atau koleksi yang ada

Idf_t = Hasil *inverse* dari df_t

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

2.2.4 K-NN (K-Nearest Neighbor)

K-Nearest Neighbor (KNN) adalah salah satu metode paling sederhana untuk memecahkan masalah klasifikasi (Adeniyi, Wei, & Yongqua, 2016). Algoritma ini sering digunakan untuk klasifikasi teks dan data (Samuel, Delima, & Rachmat, 2014).

Rumus *cosine similarity (CosSim)* akan menghasilkan nilai yang lebih optimal untuk pembobotan tiap-tiap kata pada dokumen teks yang akan diproses pada klasifikasi teks menggunakan metode KNN. Pesamaan *cosine similarity (CosSim)*

digunakan untuk mengetahui angka similaritas dari dokumen tersebut. Persamaan 2.4 adalah persamaan dari *cosine similarity*.

$$CosSim(q, d_j) = \frac{d_j.q}{|d_j|.|q|} = \frac{\sum_{i=1}^{t}(w_{ij}.w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2}.\sqrt{\sum_{i=1}^{t} w_{iq}^2}} \qquad \text{Persamaan (2.4)}$$

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 2.6 (Herdiawan, 2015).

$$n = \frac{k * N(cm)}{Maks\ [N(cm)]j=1...Nc} \qquad \text{Persamaan (2.5)}$$

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 2.6 dapat digunakan untuk menghitung peluang dari dokumen uji X pada kategori m (Baoli, Shiwen, dan Qin, 2003).

$$p(x, c_m) = argmax_m \frac{\sum_{d_j \in top\_n\_kNN_{cm}} sim(x,d_j) y(d_j.c_m)}{\sum_{d_j \in top\_n\_kNN_{cm}} sim(x,d_j)} \quad \text{Persamaan (2.6)}$$

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.

2.2.5 Confusion Matrix

Confusion matrix merupakan jenis perhitungan yang digunakan untuk menghitung hasil akurasi dalam suatu penelitian dalam bentuk tabel. Berikut adalah tabel *confusion matrix* pada table 2.2.

Tabel 2. 2 Tabel *confusion matrix*

Kategori x	Hasil aktual	
Hasil Prediksi	TP	FP
	FN	TN

Keterangan :

- TP (*True Positive*) menunjukkan jumlah data yang benar sesuai kategori x dan setiap kategori diprediksi benar oleh sistem.
- TN (*True Negative*) menunjukkan jumlah data yang tidak sesuai dengan kategori x namun dianggap benar oleh sistem.
- FN (*False Negative*) menunjukkan jumlah data yang benar dengan kategori x dan diprediksi salah oleh system.
- FP (*False Positive*) menunjukkan jumlah data yang salah dengan kategori x dan diprediksi salah oleh sistem.

Precision digunakan untuk mengukur hasil kategori data yang diklasifikasikan sesuai kategorinya. Menghitung *precision* dapat menggunakan persamaan 2.7.

$$Precision = \frac{TP}{TP+FP} \qquad \text{Persamaan (2.7)}$$

Recall digunakan untuk mengetahui ukuran keberhasilan system untuk mengenali sebuah kategori. Menghitung recall dapat menggunakan persamaan 2.8.

$$Recall = \frac{TP}{TP+FN} \qquad \text{Persamaan (2.8)}$$

Akurasi digunakan untuk mengetahui jumlah dokumen yang diklasifikasikan dengan benar secara positif ataupun negatif. Menghitung nilai akurasi dapat menggunakan persamaan 2.9.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN}$$ Persamaan (2.9)