BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian terkait ini penulis banyak mencari referensi dari penelitian-penelitian sebelumnya yang berkaitan. Sehingga penulis dapat melihat perbedaan antara penelitian terdahulu dengan penelitian sekarang. Tabel 2.1 di bawah ini menjelaskan secara ringkas dari beberapa penelitian yang ada sebelumnya:

Tabel 2.1 Perbandingan Hasil Penelitian

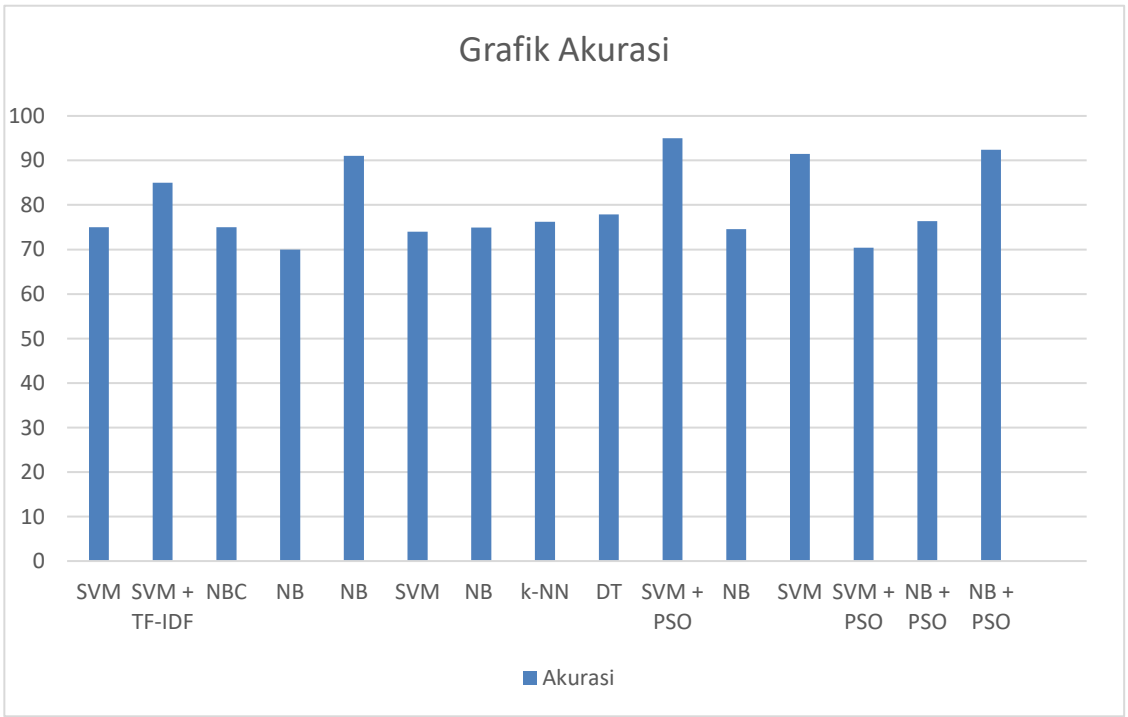
No.	PENELITI & TAHUN	DATA SET	OPTIMASI	PREPROCESSING	ALGORITMA	HASIL
1	(Yuniasari & Maspiyanti, 2021)	300	none	Case folding, tokenisasi, cleansing, stopword removal, convert emoticon, stemming	SVM	75%
2	(Gifari et al., 2022)	200	none	Normalization, Case folding, tokenisasi, cleansing, filtering, stemming	SVM, TF-IDF	85%
3	(Putri et al., 2022)	1546	none	Cleansing, Case folding, tokenisasi, stopword, splitting	NBC	75%
4	(Darwis et al., n.d.)	1179	none	casefolding, filtering, tokenisasi, slang replacement dan stopword removal	NB	69.97%
5	(Anggraini et al., 2021)	2646	none	casefolding, filtering, tokenisasi, slang replacement dan stopword removal	NB	91.06%
6	(Isnain et al., 2021)	2000	none	casefolding, cleansing, stemming, tokenisasi, stopword	SVM	74%
7	(Batlayeri et al., 2022)	1000	none	casefolding, cleansing, tokenisasi, stopword	NB, k-NN, DT	74.92% 76.22% 77.85%

No.	PENELITI & TAHUN	DATA SET	OPTIMASI	PREPROCESSING	ALGORITMA	HASIL
8	(Sabrila et al., 2022)	1000	PSO	normalisasi, case folding, cleansing, tokenization, stopword removal, dan stemming	SVM + PSO	95%
9	(Petiwi et al., 2022)	5000	none	case folding, filtering, tokenizing, stopword removal, labelling	NB, SVM	74,6% dan 91,5%
10	(Astuti & Taufan, 2022)	470	PSO	transform case, removal annotation, tokenizing, filtering dan stemming	SVM, NB + PSO	70,43% dan 76,38%
11	(Pramukti et al., 2022)	302	PSO	cleansing, remove duplicate, sleksi data, normalisasi, transform case, tokenizing, filtering, stopword, stemming, dan pemberian label	NB + PSO	92,37%
12	(Limbong et al., 2022)	500	None	normalization, case folding, tokenizing, dan stopwords	NB, KNN	0,914 0,928
13	(Kundana, 2022)	391	None	cleansing, remove duplicate, sleksi data, normalisasi, transform case, tokenizing, filtering, stopword, stemming, dan pemberian label	NB, DT	?

Metode optimasi PSO yang dipadukan pada algoritma SVM terbukti meningkatkan akurasi yang baik pada penelitian analisis sentimen *tweet* tentang UU cipta kerja (Sabrila et al., 2022) hasil penelitian menggunakan 10 *k-fold cross validation* menggunakan algoritma SVM menghasilkan akurasi sebesar 92,99%, presisi sebesar 93,24%, dan *recall* sebesar 93%. Sementara pada algoritma SVM dan PSO menghasilkan akurasi sebesar 95%, presisi sebesar 95,08%, dan *recall* sebesar 94,97%. Hasil yang diperoleh menunjukkan bahwa metode optimasi *Particle Swarm Optimization* dapat mengatasi kelemahan algoritma *Support Vector Machine* dalam masalah pemilihan parameter dan berhasil meningkatkan

performa yang dihasilkan di mana SVM-PSO lebih unggul dibandingkan SVM biasa dalam analisis sentimen. Penelitian yang dilakukan (Astuti & Taufan, 2022) menunjukkan hasil pengujian pada algoritma klasifikasi menggunakan PSO lebih baik dibandingkan tanpa menggunakan PSO yaitu memiliki hasil akurasi algoritma NB dan SVM adalah 64,04% dan 72,55%, sedangkan hasil akurasi setelah menggunakan PSO pada algoritma SVM dan NB adalah 70,43% dan 76,38%.

Penelitian yang dilakukan oleh (Pramukti et al., 2022) menggunakan metode *Naïve Bayes* dan *Particle Swarm Optimization* sebagai *feature selection*, selain itu terdapat tahap *preprocessing* yang didalamnya meliputi *cleansing, remove duplicate*, seleksi data, normalisasi, *case folding, tokenizing, filtering, stopwords*, *stemming*, dan *labeling*. Hasil klasifikasi yang didapat 53,31% pengguna *twitter* setuju dan 46,69% pengguna *twitter* tidak setuju dengan perpanjangan kebijakan PPKM darurat. Nilai *accuracy* yang didapatkan meningkat sebanyak 15,21% dari 77,16% menjadi 92,37%, nilai *precision* yang didapatkan meningkat sebanyak 3,07% dari 87,33% menjadi 90,40%, dan nilai *recall* yang didapatkan meningkat sebanyak 30,96% dari 64,42% menjadi 95,38%. Data tersebut dapat dilihat pada gambar 2.1 berikut ini.



Gambar 2.1 Grafik Akurasi Penelitian Terkait

2.2 Analisis Sentimen

Analisis sentimen atau *opinion mining* merupakan deteksi sikap-sikap terhadap objek atau orang. Analisis sentimen bertujuan untuk melakukan penilaian terhadap emosi, sikap, pendapat, evaluasi yang disampaikan oleh seseorang pembicara atau penulis terhadap sebuah produk atau terhadap tokoh masyarakat (Afandi et al., n.d.; Lesmana & Nabyla, 2020; Priandi et al., 2021). Alasan tersebut menyebabkan beberapa penelitian terutama pada *review* produk didahului dengan menentukan elemen dari sebuah produk yang sedang dibicarakan sebelum memulai proses analisis sentimen atau *opinion mining* (Vinodhini & Chandrasekaran, 2012).

Analisis sentimen dapat digunakan untuk mendapatkan tiga persentase sentimen positif dan sentimen negatif terhadap seseorang, perusahaan, institusi, produk atau pada sebuah kondisi tertentu di masyarakat. Nilai dari analisis sentimen bisa dipecah menjadi 3 kategori yakni, sentimen positif, sentimen negatif dan sentimen netral atau diperdalam lagi sehingga dapat menemukan siapa atau kelompok yang menjadi sumber sentimen positif atau sentimen negatif.

2.3 Twitter

Twitter adalah media sosial yang memungkinkan penggunanya untuk mengirim dan membaca pesan *tweet* dalam bentuk teks, gambar, atau video. Media sosial *Twitter* berbeda dengan media sosial lainnya, terutama dalam hal membuat status dan penulisan *tweet*. Jejaring sosial selain *Twitter*, tidak ada batasan jumlah karakter yang dapat ditulis, sedangkan *Twitter* hanya menawarkan 280 karakter yang dapat ditulis sebagai status atau *tweet*. *Twitter* bersifat publik, artinya segala sesuatu yang ditulis atau dibagikan dapat dilihat oleh semua pengguna lain, tetapi pengguna *Twitter* dapat membatasi pengiriman *tweet* yang hanya dapat dilihat oleh teman atau biasa disebut *only followers*. *Twitter* memiliki fitur utama yakni dapat menuliskan status atau cuitan (*tweet*) serta dapat melakukan pengiriman pesan kepada pengguna lain, fitur lain dari sosial media *twitter* sebagai berikut:

1. *Following*

   Salah satu fitur utama jejaring sosial *Twitter* adalah *following*. Fitur ini memungkinkan pengguna untuk terhubung dengan pengguna lain atau bisa disebut pertemanan. Setiap *tweet* yang diunggah oleh pengguna yang diikuti dapat dilihat di beranda pengguna yang mengikutinya.

2. *Retweet*

   Fitur *retweet* adalah fitur yang memungkinkan pengguna untuk dengan mudah membagikan *tweet* dari pengguna lain sehingga muncul di beranda pribadi mereka.

3. *Hashtag*

   Hashtag atau biasa disebut tagar adalah fitur *twitter* yang membantu untuk mengelompokkan *tweet*. Dimana setiap *tweet* yang ditulis dapat ditambahkan *hashtag* berupa kata atau *keyword* dari *tweet* tersebut. Salah satu fungsi *hashtag* adalah untuk mengelompokkan atau mempermudah pencarian kata kunci dari *tweet*.

4. *Trending Topic*

   Fitur *trending topic* adalah fitur yang menampilkan topik atau *hashtag* yang sedang populer atau sering dibicarakan oleh pengguna *tweet*. Adanya *trending topic* membuat pengguna lebih mudah untuk mengetahui hal apa saja yang sedang viral di kalangan masyarakat.

2.4 Naïve Bayes

Naive Bayes merupakan sebuah pengklasifikasian atau penggolongan data yang menghitung kemungkinan dari dataset yang tersedia (Saleh, 2015). Sedangkan menurut (Bustami, 2014) *Naive Bayes* merupakan pengklasifikasian untuk memprediksi peluang masa depan dengan metode probabilitas dan statistik sesuai

dengan pengalaman di waktu sebelumnya. Kelebihan dari algoritma *Naive Bayes* adalah data yang di butuhkan untuk menetapkan perkiraan parameter dalam proses penggolongan dalam menggunakan metode ini hanya memerlukan jumlah data pelatihan yang kecil (Saleh, 2015).

Sedangkan menurut (Syarli & Muin, 2016) kelebihan *Naive Bayes* yaitu mudah diimplementasikan dan pada banyak kasus memberikan hasil yang baik, kemudian kekurangnnya yaitu tidak terkaitnya antar fitur atau bersifat independent, sedangkan pada kenyataannya keterkaitan itu harus ada dan tidak dapat dimodelkan oleh *Naive Bayesian Classifier*. Klasifikasi ini berdasarkan teori Bayes. *Naïve Bayes* berasumsi bahwa efek dari nilai atribut pada kelas tertentu independen dari nilai-nilai dari atribut lainnya.

Proses diawali dengan memasukkan data latih Rumus algoritma *Naive Bayes* ditunjukkan pada persamaan (1) berikut:

$$\mathrm{P(c)} = \frac{Nc}{\mathrm{N}doc} \tag{1}$$

Keterangan:

c : Kategori atau kelas

doc : Dokumen

Nc : Banyaknya kategori c pada dokumen latih

Ndoc : Banyaknya keseluruhan dokumen latih yang digunakan

Perhitungan selanjutnya dari probabilitas bahwa kata i termasuk dalam kategori atau kelas tertentu dapat dilakukan dengan menggunakan persamaan (2) berikut ini.

$$\mathrm{P(H|X)} = \frac{P(X|H)P(H)}{\mathrm{P(X)}} \tag{2}$$

Dimana :

X *:* Data sampel dengan kelas (label) yang tidak diketahui

H : Hipotesa bahwa X adalah data dengan peluang kelas (label) C

P(H|X) : Peluang bahwa hipotesa benar (valid) untuk data sampel X yang diamati

P(X|H) : Peluang data sampel X, bila diasumsikan bahwa hipotesa benar (valid)

P(H) : Peluang dari hipotesa H

P(X) : Peluang data sampel yang diamati

2.5 Decision Tree

Decision Tree adalah algoritma yang paling banyak digunakan untuk masalah klasifikasi (Oktafia & Pardede, 2008). Algoritma *Decision Tree* bersifat sangat kuat, populer, berbasis logika, dan mudah dipahami (Lakshmi et al., 2016). Hal yang menarik dari *Decision Tree* adalah penggunaan struktur pohon (*tree*) yang berfungsi untuk merepresentasikan aturan yang terbentuk dari hasil klasifikasi (Anam et al., 2018). *Decision Tree* sendiri menggunakan metode *supervised machine learning* yaitu proses pembelajaran dimana data baru diklasifikasikan berdasarkan *training samples* yang sudah ada.

Nilai *gain* adalah information *gain* yang digunakan untuk mencari variabel/atribut pada dataset (S) yang digunakan sebagai *root/node* dan *node* cabang, dan merupakan atribut dengan nilai *gain* tertinggi. Anda dapat menggunakan konsep *entropi*, *koefisien Gini*, dan kesalahan klasifikasi untuk mencari nilai perolehan (*information gain*). Nilai profit maksimum yang diperoleh dari atribut- atribut dataset (data pelatihan) pertama-tama digunakan untuk mencari atribut yang layak mendapatkan akar (pohon keputusan) dari pohon keputusan. Kemudian proses pencarian atribut yang akan menjadi cabang diulangi sampai menemukan daun yang menjadi label kelasnya. *Entropy* yang informasinya diperoleh dengan nilai entropi adalah ekspresi untuk menghitung keseragaman atribut (A) dari data sampel (S) menggunakan persamaan (3) berikut ini.

$$Entrophy(S) = \sum_{i=1}^{n} p_i log_2 p_i \quad (3)$$

S = Himpunan kasus dalam dataset

A = Fitur (atribut)

n = jumlah partisi atribut

S Pi = proposi dari Si terhadap S

Maka pada persamaan (4) berikut:

$$Entrophy(S) = \sum_{i=1}^{n} p(i|s) log_2 p(i|s) \quad (4)$$

Gain(*S, A*) adalah informasi dari attribute A pada koleksi contoh:

$$Gain(S, A) = Entrop\ y(S) \sum_{i=1}^{n} \frac{|Si|}{|S|} \quad (5)$$

$$Entrop\ y(S_i)$$

Information Gain dengan Nilai Gini Index

$$Ginit(t) = 1 \sum_{i=1}^{n} [p(i|s)^2 \quad (6)$$

$$Gain(S, A) = Entropy(S) \sum_{i=1}^{n} \frac{|Si|}{|S|} Gini(S_i) \quad (7)$$

Information Gain dengan *classification error*, nilai C. *Error* diperoleh dari nilai *value attribut* yang terkecil dari *class label*

$$C.\ Error(S) = 1 \quad max_i[p(i|s)] \quad (8)$$

$$Gain(S, A) = Error(S) \sum_{i=1}^{n} \frac{|Si|}{|S|} Error(S_i) \quad (9)$$

Layaknya sebuah pohon, Seperti halnya pohon, pohon keputusan memiliki akar (*root/node*), batang/cabang (*branch node*), dan daun (*leaf*).

2.6 Rapidminer

RapidMiner merupakan *software*/perangkat lunak untuk pengolahan data. Dengan menggunakan prinsip dan algoritma *data mining*, *RapidMiner* mengekstrak pola-pola dari data set yang besar dengan mengkombinasikan metode statistika, kecerdasan buatan dan *database*. *RapidMiner* memudahkan penggunanya dalam melakukan perhitungan data yang sangat banyak dengan menggunakan operator. Operator ini berfungsi untuk memodifikasi data. Data dihubungkan dengan node-*node* pada operator kemudian kita hanya tinggal menghubungkannya ke *node* hasil untuk melihat hasilnya. Hasil yang diperlihatkan *RapidMiner* pun dapat ditampilkan secara visual dengan grafik. Menjadikan *RapidMiner* adalah salah satu *software* pilihan untuk melakukan ekstraksi data dengan metode-metode *data mining* (Rahmat C.T.I. et al., 2017).