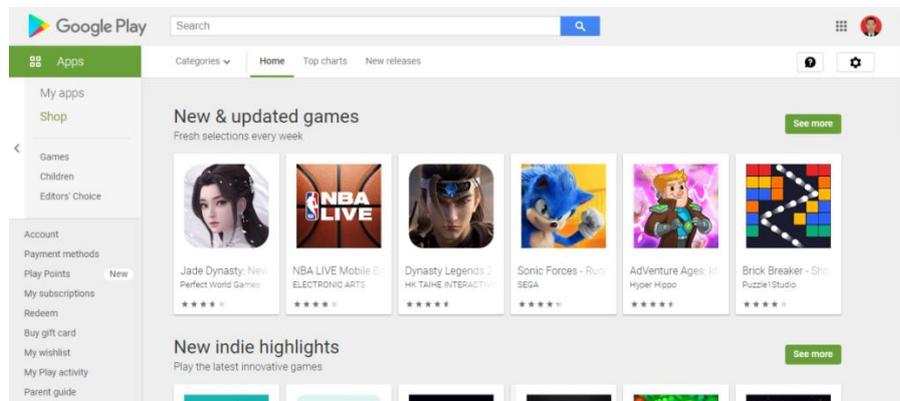


BAB II

LANDASAN TEORI

2.1 *Google Play Store*

Google Play Store adalah sebuah layanan konten digital yang dimiliki oleh *Google* yang berisi produk-produk digital seperti aplikasi, musik atau lagu, buku, *game* maupun pemutar media berbasis *cloud* (Saputra et al., 2019). Dilansir dari situs resmi *Google Play Store* tercatat pada tahun lalu sudah terjadi lebih dari 140 Miliar proses *download* konten yang ada di *Google Play Store*, lebih dari 2,5 Miliar pengguna aktif setiap bulan dan *Google Play Store* juga sudah tersedia di 190 negara di dunia (Google, 2022). Gambar 2.1 merupakan tampilan *Google Play Store* yang diakses menggunakan *browser*.



Gambar 2.1 Tampilan *Google Play Store* Pada *Browser*

2.2 *Aplikasi PeduliLindungi*

Aplikasi PeduliLindungi adalah sebuah aplikasi yang dikembangkan guna membantu instansi pemerintah dalam melakukan proses pelacakan untuk menghentikan penyebaran COVID-19. *Aplikasi PeduliLindungi* dalam melakukan

pelacakan sangat mengandalkan partisipasi dari masyarakat dengan cara membaca data lokasi masyarakat saat bepergian sehingga dapat dilakukan penelusuran riwayat kontak dengan penderita COVID-19 (Pedulilindungi.id, 2022). Aplikasi PeduliLindungi pada *Google Play Store* memiliki total *download* lebih dari 50 juta *download*. Gambar 2.2 merupakan tampilan dari aplikasi PeduliLindungi yang diakses menggunakan perangkat *android*.



Gambar 2.2 Tampilan Aplikasi PeduliLindungi

Aplikasi PeduliLindungi memiliki fitur untuk melihat sertifikat vaksin bagi pengguna yang telah melakukan vaksin COVID-19. Kemudian pengguna juga bisa melihat riwayat hasil *test* COVID-19 jika *test* tersebut dilakukan di lab yang terfasilitasi oleh KEMENKES. Lalu terdapat informasi untuk aturan perjalanan, pelayanan kesehatan, pencarian kamar pada rumah sakit dan juga informasi terkait statistik COVID-19.

2.3 Analisis Sentimen

Analisis sentimen adalah bidang ilmu untuk menganalisis opini orang, sentimen, evaluasi, penilaian, sikap dan emosi terhadap entitas produk, layanan, organisasi, individu, masalah, peristiwa, topik dan atributnya. Terdapat beberapa penamaan terhadap bidang ini, yaitu analisis sentimen, penambangan opini atau *opinion mining*, ekstraksi opini atau *opinion extraction*, penambangan sentimen atau *sentiment mining* dimana saat ini semuanya masuk dalam ranah analisis sentimen (Liu, 2012).

Analisis sentimen merupakan proses untuk mencari informasi berupa polaritas sentimen yang digunakan untuk mengungkapkan suatu makna yang sama (Giachanou & Crestani, 2016). Analisis sentimen dapat dilakukan di berbagai tingkat perincian yaitu pada tingkat dokumen, paragraf, kalimat atau klausa (Fink et al., 2011).

Metode yang bisa digunakan untuk proses analisis sentimen antara lain berbasis pembelajaran mesin atau *machine learning*, berbasis *lexicon* dan metode pendekatan *hybrid* (Sharma et al., 2020). Untuk metode dengan berbasis pembelajaran mesin sendiri dibagi menjadi *supervised learning* dan *unsupervised learning* (Medhat et al., 2014). Proses klasifikasi pada analisis sentimen pada dasarnya adalah permasalahan pada klasifikasi teks, oleh karena itu metode *supervised learning* dapat langsung digunakan dan diimplementasikan seperti dengan menggunakan metode *Support Vector Machine (SVM)* atau *Naïve Bayes Classifier (NBC)* (Liu, 2015). Pada penelitian kali ini analisis sentimen akan

dilakukan dengan metode pembelajaran mesin (*machine learning*) dengan berbasis *supervised learning* pada tingkat kalimat.

2.4 *Multinomial Naïve Bayes (MNB)*

Multinomial Naïve Bayes (MNB) yang merupakan variasi lain dari *Naïve Bayes*, *Multinomial Naïve Bayes* merupakan metode *Probabilistic Reasoning* yang bertujuan untuk melakukan klasifikasi data pada kelas tertentu (Bunga et al., 2018). Metode ini mengasumsikan setiap atribut data tidak bergantung satu dengan lainnya, tetapi asumsi tersebut berlawanan dengan keadaan sebenarnya karena dokumen ataupun teks harus memiliki keterhubungan di setiap katanya agar dokumen tersebut memiliki makna. Meskipun demikian, metode ini dalam klasifikasi memiliki hasil yang cukup memuaskan (Sabrani et al., 2020).

Metode MNB adalah hasil dari perkalian *prior probability* dan *conditional probability* yang mana hasil akhirnya merupakan sebuah *posterior probability* dengan nilai paling besar untuk suatu kelas tertentu dengan rumus (Christopher et al., 2008):

$$c_{map} = \arg \max_{c \in C} P(c) \prod_{1 \leq k \leq n_d} P(t_k | c) \quad (2.1)$$

Keterangan:

- $\arg \max$: Fungsi untuk mencari *posterior probability* terbesar suatu kelas
- $P(t_k | c)$: *Conditional probability*, peluang kemunculan kata k dalam suatu kelas tertentu
- $P(c)$: *Prior probability*, peluang kemunculan sebuah kelas dari seluruh pengamatan yang dilakukan

Nilai probabilitas sebuah dokumen d berada dalam kelas c dinyatakan dengan rumus:

$$P(c) = \frac{N_c}{N'} \quad (2.2)$$

Keterangan:

N_c : Jumlah dokumen di kelas c

N' : Jumlah total dokumen *training*

Untuk menghitung nilai *conditional probability* dapat menggunakan rumus :

$$P(t|c) = \frac{T_{ct}}{\sum t' \in VT_{ct'}} \quad (2.3)$$

Keterangan:

T_{ct} : Frekuensi suatu dalam kelas c pada dokumen *training* termasuk yang berulang

$\sum t' \in VT_{ct'}$: Jumlah total kata dalam suatu kelas c

Dalam suatu kelas seringkali terdapat suatu data yang tidak terlihat pada data *training* hal tersebut menyebabkan peluang kata dalam kelas bernilai 0. Jika *conditional probability* bernilai 0 maka hasil perkalian *posterior probability* juga akan bernilai 0. Untuk mencegah peluang bernilai 0 maka digunakan *add-one smoothing (Laplace Smoothing)*. Cara kerjanya adalah dengan menambahkan 1 angka pada *numerator* dan jumlah kosakata pada *denominator*, dapat dilihat pada rumus :

$$P(t|c) = \frac{T_{ct} + 1}{\sum t' \in VT_{ct'} + B'} \quad (2.4)$$

Keterangan:

B' : Total kosakata atau kata unik pada seluruh kelas dalam dokumen *training*

Dengan menggunakan metode Multinomial Naïve Bayes dapat melakukan proses klasifikasi dengan baik dan memiliki rata-rata tingkat akurasi yang cukup tinggi (Bunga et al., 2018).

2.5 Decision Tree (DT)

Decision Tree (DT) merupakan suatu representasi sederhana untuk klasifikasi, proses pada *Decision Tree* adalah mengubah data dalam bentuk tabel menjadi bentuk pohon atau *tree* lalu mengubah bentuk pohon tersebut menjadi aturan atau *rule* (Rizkia et al., 2019).

Dalam penelitian ini bahasa pemrograman yang digunakan adalah bahasa pemrograman *python*. Pada pemrograman *python* untuk penerapan algoritma biasanya menggunakan *library sklearn*, termasuk untuk algoritma DT juga menggunakan *library sklearn*. Dilansir dari *website* resminya *Sklearn* sendiri untuk algoritma DT mengimplementasikan versi CART (*Classification And Regression Trees*) yang di optimasi (Scikit-learn.org, 2022), sehingga pada penelitian ini menerapkan algoritma DT dengan versi CART (*Classification And Regression Trees*).

CART merupakan salah satu versi dari algoritma DT, CART adalah suatu metode yang menggambarkan hubungan antara variabel dependen dengan satu atau lebih variabel independen (Hartati et al., 2012). CART adalah metode non parametrik yang berfungsi untuk mendapatkan sebuah kelompok data akurat

sebagai penanda suatu proses klasifikasi. CART merupakan metode yang terdiri dari metode pohon klasifikasi dan pohon regresi, jika variabel dependen yang dimiliki mempunyai tipe kontinu atau numerik maka yang dihasilkan CART adalah pohon regresi sedangkan jika variabel dependen yang dimiliki bertipe kategorial maka pohon yang dihasilkan oleh CART adalah pohon klasifikasi (Indah P. et al., 2019).

Langkah-langkah algoritma CART adalah sebagai (Astuti, 2018):

1. Mempersiapkan data yang akan diklasifikasikan.
2. Menentukan variabel prediktor sebagai variabel untuk dasar pengelompokan berdasarkan variabel tujuan (target).
3. Menentukan calon cabang (*candidate split*) kiri dan kanan.
4. Mengukur *goodness* atau kesesuaian dari calon masing-masing cabang s pada simpul keputusan t yang dihitung dengan rumus:

$$\Phi(s|t) = 2P_L P_R \sum_{j=1}^{jmlh\ kategori} |P(j|t_L) - P(j|t_R)| \quad (2.5)$$

Keterangan:

- t_L : Calon cabang kiri dari node keputusan t
- t_R : Calon cabang kanan dari node keputusan t
- P_L : Jumlah data catatan pada calon cabang kiri t_L / jumlah data catatan seluruhnya
- P_R : Jumlah data catatan pada calon cabang kiri t_R / jumlah data catatan seluruhnya
- $P(j|t_L)$: Jumlah data catatan berkategori j pada calon cabang kiri t_L /

jumlah data catatan pada *node* keputusan t

$P(j|t_R)$: Jumlah data catatan berkategori j pada calon cabang kanan
 t_R / jumlah data catatan pada *node* keputusan t

5. Tentukan calon cabang untuk *node* keputusan dengan memilih nilai terbesar, cabang ini tidak dihitung lagi selanjutnya.
6. Gambarkan cabang *node* keputusan dan *node* kejadian terminasi.
7. Ulangi langkah 4 sampai tidak terdapat lagi cabang *node* keputusan.

Berikut ini merupakan kelebihan dan juga kekurangan dari algoritma

CART (Timofeev, 2004). Kelebihan CART:

1. CART merupakan metode non parametrik, sehingga tidak membutuhkan spesifikasi dari setiap bentuk fungsional.
2. CART tidak perlu menentukan variabel di awal proses, karena CART akan dengan sendirinya memilih variabel yang akan paling signifikan dan membuang variabel yang tidak signifikan.
3. CART dapat dengan mudah menangani *outliers* (data yang memiliki karakteristik unik dan berbeda dengan hasil observasi).
4. CART tidak memiliki asumsi dan memiliki komputasi cepat.
5. CART fleksibel dan memiliki kemampuan untuk menyesuaikan sesuai kebutuhan.

Sedangkan kekurangan CART adalah :

1. CART memiliki kemungkinan akan menghasilkan *decision tree* yang tidak stabil.
2. CART membagi data hanya berdasarkan satu variabel.

2.6 AdaBoost (AB)

Pada awalnya *boosting* disebut hipotesis *boosting* karena mengacu pada metode *Ensemble* dengan jenis apa pun yang dapat menggabungkan beberapa *learner* yang lemah menjadi *learner* yang kuat, kebanyakan ide metode *boosting* adalah dengan cara berurutan melatih prediktor untuk mencoba memperbaiki prediktor pendahulunya (Géron, 2017). *Boosting* merupakan pengulangan serial, dimana kesalahan yang terjadi pada pengulangan sebelumnya digunakan untuk mempengaruhi klasifikasi pada pengulangan berikutnya dengan cara pembobotan kesalahan klasifikasi pada langkah-langkah selanjutnya (Unpingco, 2016). Saat ini banyak metode *boosting* yang tersedia, salah satu metode *boosting* tersebut adalah *AdaBoost*.

Adaboost (AB) merupakan kependekan dari *Adaptive Boosting* adalah sebuah algoritma iteratif yang diusulkan oleh Freund dan Schapir pada tahun 1997 yang berfungsi untuk membangun pengklasifikasi yang kuat, dengan ide utamanya adalah melatih pengklasifikasi lemah pada set pelatihan yang sama kemudian menggabungkan pengklasifikasi yang lemah tersebut menjadi pengklasifikasi akhir yang kuat. Secara sederhananya *Adaboost* adalah proses peningkatan sederhana dari algoritma klasifikasi yang lemah dan proses ini dapat meningkatkan klasifikasi data dengan mengurangi bias dan *varians* melalui pelatihan berkelanjutan (Wu et al., 2020).

Berikut ini merupakan langkah-langkah dari algoritma *AdaBoost* (Freund & Schapire, 1997):

Input: Sequence of N labeled examples $\langle (x_1, y_1), \dots, (x_N, y_N) \rangle$

Distribution D over the N examples

Weak learning algorithm WeakLearn

Integer T specifying number of iterations

Initialize the weight vector: $W_i^1 = D(i)$ for $i = 1, \dots, N$

Do for $t = 1, 2, \dots, T$

1. *Set*

$$P^t = \frac{W^t}{\sum_{i=1}^N w_i^t} \quad (2.6)$$

2. *Call WeakLearn, providing it with the distribution P^t ; get back a hypothesis $h_t: X \rightarrow [0,1]$.*

3. *Calculate the error of h_t : $\varepsilon_t = \sum_{i=1}^N P_i^t |h_t(x_i) - y_i|$.*

4. *Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$.*

5. *Set the new weights vector to be*

$$w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|} \quad (2.7)$$

Output the hypothesis

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T (\log 1/\beta_t) h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \log 1/\beta_t \\ 0 & \text{otherwise.} \end{cases} \quad (2.8)$$

2.7 TF-IDF

Feature Extraction adalah proses untuk mengekstrak atribut data mentah ke dalam statistik atau algoritma *machine learning*. Proses ini dikenal sebagai vektorisasi dikarenakan hasil dari proses ini adalah himpunan dalam bentuk vektor numerik. Hal tersebut diperlukan karena algoritma konvensional bekerja

pada vektor numerik dan tidak bisa bekerja langsung pada data teks mentah (Veluchamy et al., 2018). Salah satu *feature extraction* yang dapat digunakan adalah metode TF-IDF.

Term Frequency-Inverse Document Frequency (TF-IDF) merupakan pendekatan untuk memberi bobot pada jumlah kata dengan ukuran seberapa sering kata tersebut muncul pada sebuah dokumen. Pendekatan ini digunakan untuk mengatasi permasalahan dimana jumlah kata mentah yang mengarah ke fitur terlalu membebani kata-kata yang sering muncul, hal tersebut dapat menyebabkan *suboptimal* pada beberapa algoritma klasifikasi (VanderPlas, 2016). *Term Frequency* (TF) adalah frekuensi kemunculan kata atau kelompok kata-kata pada sebuah dokumen, disebut juga model *Bag of Words*. Masing-masing dokumen direpresentasikan sebagai vektor 0s dan 1s, dimana jika sebuah kata ada dalam sebuah dokumen maka posisi yang sesuai dalam vektor dikodekan sebagai “1” dan jika tidak dikodekan sebagai “0”. Rumus TF adalah (Veluchamy et al., 2018):

$$TF(\text{Word}) = \frac{\text{Frequency of Word in the Document}}{\text{Number of Word in the Document}} \quad (2.9)$$

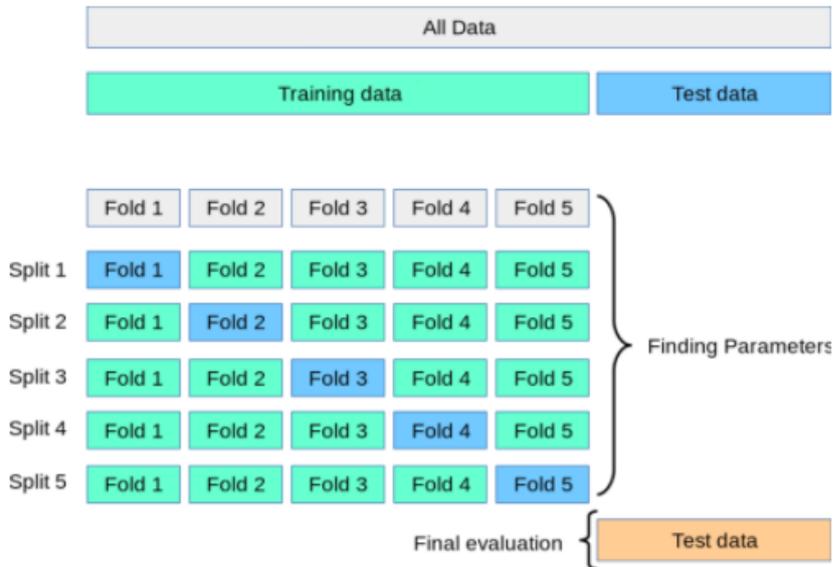
Inverse Document Frequency (IDF) dari sebuah kata adalah ukuran dari kepentingan relatif dari kata tersebut di seluruh *corpus*. Rumus IDF adalah :

$$IDF(\text{Word}) = \log \left(\frac{\text{Total Number of Documents}}{\text{Number of Documents Containing the Word}} \right) \quad (2.10)$$

Penggunaan TF-IDF ini dapat mengurangi dampak dari fitur yang kurang informatif pada sebuah dokumen, meskipun fitur tersebut sering muncul pada dokumen.

2.8 K-Folds Cross Validation

K-Folds Cross Validation merupakan metode yang dapat digunakan untuk melakukan pengujian pola klasifikasi. Pada *K-Folds Cross Validation* data yang diuji dibagi tergantung banyaknya nilai k, data yang dibagi diasumsikan sebagai D1, D2... Dk dimana setiap D memiliki jumlah data yang sama (Wahyuningsih & Utari, 2018). Gambar 2.3 merupakan ilustrasi dari *K-Folds Cross Validation* yang memiliki nilai k=5, dimana proses pengulangan pengujian dilakukan sebanyak 5 kali dan juga semua data yang ada dibagi menjadi 5 bagian pada pengujian yang dilakukan.



Gambar 2.3 Ilustrasi *K-Folds Cross Validation*

Sumber : <https://vitalflux.com/k-fold-cross-validation-python-example/>

2.9 Confusion Matrix

Confusion Matrix merupakan sebuah tabel yang memiliki dimensi $n \times n$ digunakan untuk merangkum keakuratan dari model prediksi klasifikasi. Pada *confusion matrix* n mewakili jumlah kelas, secara sederhana *confusion matrix* merupakan korelasi antara label aktual dan label prediksi dari suatu model (Bhatia, 2019). *Confusion matrix* ini sangat penting, karena *confusion matrix* menjadi parameter pengujian dalam perhitungan akurasi (Novantirani et al., 2015). Tabel 2.1 merupakan bentuk dari *confusion matrix* yang terdiri dari label *actual* dan *predicted*.

Tabel 2.1 *Confusion Matrix*

<i>Actual</i> \ <i>Predicted</i>	<i>Negative</i>	<i>Positive</i>
<i>Negative</i>	<i>True Negative (TN)</i>	<i>False Positive (FP)</i>
<i>Positive</i>	<i>False Negative (FN)</i>	<i>True Positive (TP)</i>

Keterangan :

- True Positive (TP)* : Tupel positif yang dilabeli dengan benar
True Negative (TN) : Tupel negatif yang dilabeli dengan benar
False Positive (FP) : Tupel negatif yang salah dilabeli dengan positif
False Negative (FN) : Tupel positif yang salah dilabeli dengan negatif

2.10 Performance Evaluation Measure

PEM (*Performance Evaluation Measure*) atau pengukuran evaluasi performa merupakan suatu tahapan yang berfungsi untuk mengukur performa dari

suatu sistem. PEM ini digunakan dalam proses *training data* yang bertujuan untuk melakukan evaluasi model yang sudah dibuat (Imron, 2019). Pengukuran evaluasi ini terdiri dari 4 bagian yaitu (Veluchamy et al., 2018):

1. *Accuracy* adalah proporsi dari jumlah total prediksi yang benar, dinyatakan dengan rumus:

$$Accuracy = \frac{True\ Positive\ (TP) + True\ Negative\ (TN)}{Total\ Number\ Of\ Observations} \quad (2.11)$$

2. *Precision*, adalah ukuran seberapa baik pengklasifikasi mengklasifikasikan ulasan sebagai positif sentimen, dinyatakan dengan rumus:

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (2.12)$$

3. *Recall* adalah ukuran seberapa baik pengklasifikasi mengklasifikasikan ulasan sebagai negatif sentimen, dinyatakan dengan rumus:

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (2.13)$$

4. *F1-Score* adalah metric yang menggabungkan *trade-off* dari *precision* dan *recall*, dinyatakan dengan rumus:

$$F1 - Score = \frac{2 * True\ Positive\ (TP)}{2 * True\ Positive\ (TP) + False\ Positive\ (FP) + False\ Negative\ (FN)} \quad (2.14)$$

2.11 Penelitian Terkait

Sebelumnya para peneliti sudah melakukan beberapa penelitian terkait bidang penelitian analisis sentimen dengan memasukkan unsur-unsur tambahan yang dapat memenuhi kebutuhan dengan fokus Analisis Sentimen. Tabel 2.2 merupakan penjelasan mengenai hasil penelitian disajikan dalam bentuk *state of the art*.

Tabel 2.2 *State Of The Art*

No.	Penulis	Judul	Metode	Hasil
1	(Doloksaribu & Samuel, 2022)	KOMPARASI ALGORITMA DATA MINING UNTUK ANALISIS SENTIMEN APLIKASI PEDULILINDUNGI	<i>Naïve Bayes, SVM, TF-IDF</i>	<p>Hasil pengujian algoritma <i>Naïve Bayes</i> dengan <i>count vectorizer</i> menghasilkan nilai akurasi 82,57%. Nilai <i>Class Precision</i> Negatif 88,68%, <i>Class Precision</i> Positif 64,27%, <i>Class Recall</i> Negatif 88,14% dan <i>Class Recall</i> Positif 65,48%.</p> <p>Hasil pengujian Algoritma <i>Naive Bayes</i> dengan <i>TF IDF Vectorizer</i> menghasilkan nilai akurasi 82,57%. Nilai <i>Class Precision</i> Negatif 88,63%, <i>Class Precision</i> Positif 64,33%, <i>Class Recall</i> Negatif 88,21% dan <i>Class Recall</i> Positif 65,27%.</p> <p>Hasil pengujian Algoritma <i>Support Vector Machine</i> dengan <i>Count Vectorizer</i> menghasilkan nilai akurasi 86,99%. Nilai <i>Class Precision</i> Negatif 88,56%, <i>Class Precision</i> Positif 80,32%, <i>Class Recall</i> Negatif 95,02% dan <i>Class Recall</i> Positif 62,34%.</p> <p>Hasil pengujian Algoritma <i>Support Vector Machine</i> dengan <i>TF IDF Vectorizer</i></p>

No.	Penulis	Judul	Metode	Hasil
				<p>menghasilkan nilai akurasi 89,05%. Nilai <i>Class Precision</i> Negatif 88,37%, <i>Class Precision</i> Positif 92,60%, <i>Class Recall</i> Negatif 98,43% dan <i>Class Recall</i> Positif 60,25%.</p> <p>Dapat dilihat dari hasil pengujian bahwa algoritma <i>Support Vector Machine</i> yang dibantu <i>TF IDF</i> dalam kasus ini menjadi algoritma terbaik dalam tingkat akurasi dibandingkan dengan menggunakan dengan algoritma <i>Naïve Bayes</i>.</p>
2	(Puspita & Widodo, 2021)	Perbandingan Metode <i>KNN</i> , <i>Decision Tree</i> , dan <i>Naïve Bayes</i> Terhadap Analisis Sentimen Pengguna Layanan BPJS	<i>KNN</i> , <i>Decision Tree</i> , <i>Naïve bayes</i>	Hasil dari penelitian tentang perbandingan <i>KNN</i> , <i>Decision Tree</i> dan <i>Naïve Bayes</i> terhadap analisis sentimen pengguna layanan BPJS, dengan hasil nilai tingkat akurasi <i>Decision Tree</i> lebih tinggi dari algoritma lainnya yaitu dengan nilai 96,13% untuk <i>Decision Tree</i> , 95,58% untuk <i>KNN</i> dan 89,14% untuk <i>Naïve Bayes</i> .
3	(Illia et al., 2021)	<i>Sentiment Analysis on PeduliLindungi Application Using TextBlob and VADER</i>	<i>Vader</i> , <i>TextBlob</i>	<i>Library VADER</i> secara khusus ditujukan untuk menganalisis sentimen di media sosial. Hal tersebut

No.	Penulis	Judul	Metode	Hasil
		<i>Library</i>		menunjukkan bahwa perpustakaan <i>tweet VADER</i> didominasi oleh sentimen positif, netral, dan negatif, masing-masing dan perbedaan sentimen pada kedua periode pengambilan data tidak terlalu besar.
4	(Zaidah, 2021)	ANALISIS SENTIMEN BERBASIS ASPEK PADA APLIKASI PEDULILINDUNGI MENGGUNAKAN <i>LATENT DIRICHLET ALLOCATION</i> DAN <i>SUPPORT VECTOR MACHINE</i>	<i>LDA, SVM</i>	Hasil penelitian analisis sentimen aplikasi PeduliLindungi yang telah dilakukan memperoleh hasil model akurasi sebesar 70,46%. Berdasarkan hasil penelitian yang sudah dilakukan aspek vaksin, data, aplikasi lebih dominan sentimen negatif dari pada sentimen positif.
5	(Salma & Silfianti, 2021)	<i>Sentiment Analysis of User Reviews on COVID-19 Information Applications Using Naive Bayes Classifier, Support Vector Machine, and K-Nearest Neighbor</i>	<i>Naïve Bayes, SVM, KNN</i>	Hasil klasifikasi pada penelitian ini menghasilkan rata-rata netral dengan perbandingan tipis tiap kategori. Nilai akurasi tiap algoritma, hasil SVM memiliki akurasi 76,5%, diikuti oleh NBC 72,3%, dan KNN memiliki akurasi 59,1%. Perlu metode lain dalam melakukan analisis sentimen dan klasifikasi lebih lanjut untuk meningkatkan hasil

No.	Penulis	Judul	Metode	Hasil
				akurasi.
6	(Villavicencio et al., 2021)	<i>Twitter Sentiment Analysis towards COVID-19 Vaccines in the Philippines Using Naïve Bayes</i>	<i>Naïve Bayes</i>	<p>Penelitian ini menganalisis terkait vaksin COVID-19 di Filipina pada sosial media <i>twitter</i>. <i>Tools</i> yang digunakan adalah <i>rapidminer</i>. Pengujian performa yang dilakukan pada penelitian ini menggunakan <i>K-Fold Cross Validation</i> dengan nilai <i>K</i> yang digunakan adalah 10.</p> <p>Nilai sentimen untuk tiap kategori pada penelitian ini adalah 83% positif, 9% netral dan 8% negatif. Kemudian untuk hasil klasifikasi dengan menggunakan algoritma <i>naïve bayes</i> adalah 81,77%.</p>
7	(Shofiya & Abidi, 2021)	<i>Sentiment Analysis on COVID-19-Related Social Distancing in Canada Using Twitter Data</i>	<i>SVM</i>	<p>Data yang dianalisis pada penelitian ini terkait <i>social distancing</i> di Kanada yang didapatkan dari <i>twitter</i>. Klasifikasi sentimen yang dilakukan dengan menggunakan algoritma <i>SVM</i> menghasilkan nilai akurasi sebesar 71% untuk 3 kategori yaitu positif, netral dan negatif.</p> <p>Saat kategori pada data diubah hanya menjadi positif dan negatif, nilai akurasi juga</p>

No.	Penulis	Judul	Metode	Hasil
				meningkat menjadi 81%. Lalu saat data tes dikurangi sebesar 10%, nilai akurasi meningkat menjadi 87%.
8	(Neogi et al., 2021)	<i>Sentiment analysis and classification of Indian farmers' protest using twitter data</i>	<i>Bag of Words, TF-IDF, Naïve Bayes, Decision Tree, Random Forest, SVM</i>	<p>Data yang digunakan pada penelitian ini berasal dari <i>twitter</i> terkait protes petani india. Data yang diperoleh sebanyak 18.000 <i>tweets</i> dengan periode pengambilan data selama 4 bulan.</p> <p>Hasil akurasi dengan menggunakan fitur ekstraksi <i>Bag of Words</i> adalah 72,9% <i>Naïve bayes</i>, 79,78% <i>Decision tree</i>, 96,62% <i>Random forest</i> dan 83,45% <i>SVM</i>.</p> <p>Kemudian nilai akurasi dengan menggunakan fitur ekstraksi <i>TF-IDF</i> adalah 71,33% <i>Naïve bayes</i>, 77,62% <i>Decision tree</i>, 95,51% <i>Random forest</i> dan 83,04% <i>SVM</i>.</p> <p>Dari hasil yang telah diperoleh, pada penelitian ini dapat disimpulkan bahwa penggunaan fitur ekstraksi <i>Bag of Words</i> memiliki nilai akurasi yang lebih tinggi untuk setiap algoritma</p>

No.	Penulis	Judul	Metode	Hasil
				dibandingkan dengan menggunakan fitur ekstraksi <i>TF-IDF</i> .
9	(Sabrani et al., 2020)	METODE <i>MULTINOMIAL NAÏVE BAYES</i> UNTUK KLASIFIKASI ARTIKEL <i>ONLINE</i> TENTANG GEMPA DI INDONESIA	<i>MULTINOMIAL NAÏVE BAYES</i>	Hasil dari penelitian yang telah dilakukan terkait klasifikasi artikel <i>online</i> tentang gempa di Indonesia dengan menggunakan metode <i>Multinomial Naïve Bayes</i> didapatkan akurasi yang tinggi yaitu dengan nilai <i>f-measure</i> mencapai 95,20% dan standar deviasi sebesar 1,58%.
10	(Wardani et al., 2020)	ANALISIS SENTIMEN PEMINDAHAN IBU KOTA NEGARA DENGAN KLASIFIKASI <i>NAÏVE BAYES</i> UNTUK MODEL <i>BERNOULLI</i> DAN <i>MULTINOMIAL</i>	<i>BERNOULLI NAÏVE BAYES, MULTINOMIAL NAÏVE BAYES</i>	Hasil dari penelitian ini adalah metode <i>Multinomial Naïve Bayes</i> lebih tinggi dibanding <i>Bernoulli Naïve Bayes</i> yaitu 93,45% untuk <i>Multinomial Naïve Bayes</i> dan 90,19% untuk <i>Bernoulli Naïve Bayes</i> .
11	(Syarifuddin, 2020)	ANALISIS SENTIMEN OPINI PUBLIK TERHADAP EFEK PSBB PADA <i>TWITTER</i> DENGAN ALGORITMA <i>DECISION TREE-</i>	<i>DECISION TREE, KNN, NAÏVE BAYES</i>	Hasil dari penelitian ini adalah <i>Decision Tree</i> memiliki nilai akurasi tertinggi dibandingkan dengan <i>KNN</i> dan <i>Naïve Bayes</i> yaitu dengan nilai akurasi 83,3 % untuk <i>Decision Tree</i> , 80,80% untuk <i>KNN</i> dan

No.	Penulis	Judul	Metode	Hasil
		<i>KNN-NAÏVE BAYES</i>		80,03% untuk <i>Naive Bayes</i> .
12	(Mustopa et al., 2020)	<i>Analysis of User Reviews for the PeduliLindungi Application on Google Play Using the Support Vector Machine and Naive Bayes Algorithm Based on Particle Swarm Optimization</i>	<i>SVM, Naive Bayes, PSO</i>	Berdasarkan percobaan yang dilakukan menggunakan <i>NB</i> dan <i>SVM</i> berbasis <i>PSO</i> dengan menggunakan 1364 data. Hasil <i>NB+PSO</i> menghasilkan nilai akurasi 69,00% dan nilai <i>AUC</i> = 0,859, sedangkan <i>SVM+PSO</i> memiliki nilai akurasi 93,0% dan nilai <i>AUC</i> = 0,977. Dapat disimpulkan pada penelitian ini algoritma <i>SVM+PSO</i> memiliki akurasi yang lebih tinggi dibandingkan algoritma <i>NB+PSO</i> .
13	(Fransiska & Irham Gufroni, 2020)	<i>Sentiment Analysis Provider by.U on Google Play Store Reviews with TF-IDF and Support Vector Machine (SVM) Method</i>	<i>SVM, TF-IDF</i>	Hasil analisis data menghasilkan kecenderungan positif meskipun perbedaanya tidak terlalu jauh, yaitu 54,6% positif dan 45,4% negatif. Hasil pengujian menggunakan <i>TF-IDF</i> dan <i>SVM</i> memiliki nilai yang cukup baik yaitu dengan rata-rata akurasi 84,7%, meskipun akurasi terbilang tinggi tetapi nilai tersebut tidak lebih baik dari penelitian sebelumnya.
14	(Guia et al., 2019)	<i>Comparison of Naive Bayes, Support Vector Machine, Decision</i>	<i>Naive Bayes, SVM, Decision Tree, Random</i>	Data yang digunakan adalah data <i>unlocked mobile phones</i> yang dijual di <i>Amazon.com</i> .

No.	Penulis	Judul	Metode	Hasil
		<i>Trees and, Random Forest on Sentiment Analysis</i>	<i>Forest</i>	<i>Naïve Bayes</i> menghasilkan nilai akurasi klasifikasi sebesar 83%. <i>Random Forest</i> menghasilkan nilai akurasi sebesar 88%. <i>SVM</i> menghasilkan nilai akurasi sebesar 89%. <i>Decision Tree</i> menghasilkan nilai akurasi klasifikasi 82%. Dari keempat algoritma yang dibandingkan, dapat disimpulkan bahwa <i>SVM</i> memiliki akurasi yang tinggi dibandingkan algoritma lainnya yang digunakan pada penelitian ini.
15	(Dini et al., 2015)	Integrasi Metode <i>Information Gain</i> Untuk Seleksi Fitur dan <i>Adaboost</i> Untuk Mengurangi Bias Pada Analisis Sentimen <i>Review</i> Restoran Menggunakan Algoritma <i>Naïve Bayes</i>	<i>Information gain, Adaboost, Naïve Bayes</i>	Hasil penelitian terkait analisis sentimen <i>review</i> restoran dengan menggunakan algoritma <i>Naïve Bayes</i> yang awalnya hanya memiliki nilai akurasi 70% meningkat menjadi 99,5% setelah menggunakan <i>AdaBoost</i> . Akurasi meningkat sebesar 29,5% dan <i>AUC</i> = 0,995.

Penelitian yang dilakukan memiliki keterkaitan dengan penelitian yang telah dilakukan sebelumnya yaitu mengenai analisis sentimen terkait ulasan atau opini masyarakat pada suatu entitas baik berupa produk, layanan, kebijakan

pemerintah maupun hal lainnya yang sedang hangat di internet baik pada sosial media seperti *twitter* ataupun pada layanan penyedia aplikasi seperti *Google Play Store*.

Penelitian ini bertujuan untuk melakukan pengujian perbandingan akurasi dari algoritma *Multinomial Naïve Bayes* dan *Decision Tree* dengan menerapkan *AdaBoost*, dimana penerapan *AdaBoost* pada penelitian yang sudah dilakukan sebelumnya dapat meningkatkan akurasi algoritma yang digunakan. Pemilihan algoritma *Multinomial Naïve Bayes* dan *Decision Tree* pada penelitian ini dikarenakan kedua algoritma tersebut memiliki nilai akurasi yang tinggi dibandingkan algoritma lainnya.

Pada penelitian terkait analisis sentimen yang telah dilakukan oleh (Wardani et al., 2020) mengenai pemindahan ibu kota baru dengan membandingkan metode klasifikasi *Bernoulli Naïve Bayes* dan *Multinomial Naïve Bayes*, dengan nilai performa *Multinomial Naïve Bayes* lebih tinggi dibanding *Bernoulli Naïve Bayes* yaitu 93,45% untuk *Multinomial Naïve Bayes* dan 90,19% untuk *Bernoulli Naïve Bayes*. Hasil pada penelitian tersebut membuktikan bahwa algoritma *Multinomial Naïve Bayes* memiliki nilai performa yang lebih tinggi dari algoritma *Bernoulli Naïve Bayes* yaitu dengan selisih 3,26%.

Kemudian pada penelitian yang dilakukan oleh (Puspita & Widodo, 2021) tentang perbandingan *KNN*, *Decision Tree* dan *Naïve Bayes* terhadap analisis sentimen pengguna layanan BPJS, dengan hasil nilai tingkat akurasi *Decision Tree* lebih tinggi dari algoritma lainnya yaitu dengan nilai 96,13% untuk *Decision Tree*, 95,58% untuk *KNN* dan 89,14% untuk *Naïve Bayes*. Hasil yang telah

diperoleh pada penelitian tersebut membuktikan bahwa algoritma *Decision Tree* memiliki akurasi yang lebih tinggi dibandingkan dengan algoritma *KNN* dan *Naïve Bayes*.

Ulasan terhadap aplikasi PeduliLindungi juga sudah pernah dianalisis dan diteliti salah satunya oleh (Zaidah, 2021) dengan menggunakan algoritma SVM. Hasil penelitian analisis sentimen aplikasi PeduliLindungi yang telah dilakukan memperoleh hasil model akurasi sebesar 70,46%. Pada penelitian lainnya terkait analisis sentimen terhadap ulasan pengguna aplikasi informasi COVID-19 yaitu PeduliLindungi dengan menggunakan metode *Naïve Bayes Classifier*, SVM dan KNN menghasilkan nilai akurasi untuk SVM sebesar 76,5%, diikuti oleh NBC 72,3%, dan KNN memiliki akurasi 59,1%, sehingga perlu metode lain dalam melakukan analisis sentimen dan klasifikasi lebih lanjut untuk ditingkatkan hasil akurasi (Salma & Silfianti, 2021).

AdaBoost merupakan salah satu metode untuk meningkatkan nilai akurasi pada proses klasifikasi, dibuktikan pada penelitian yang dilakukan oleh (Dini et al., 2015) terkait analisis sentimen *review* restoran dengan menggunakan algoritma *Naïve Bayes* yang awalnya hanya memiliki nilai akurasi 70% meningkat menjadi 99,5% setelah menggunakan *AdaBoost*.

Berdasarkan penelitian-penelitian yang telah dilakukan sebelumnya terkait analisis sentimen, penelitian ini akan membandingkan nilai akurasi dari algoritma *Multinomial Naïve Bayes* dan *Decision Tree* dengan menambahkan *Boosting AdaBoost* sebagai keterbaruan dari penelitian ini yang digunakan untuk

meningkatkan akurasi hasil klasifikasi dari algoritma yang digunakan pada analisis sentimen data ulasan aplikasi PeduliLindungi.

Tabel 2.3 merupakan matriks penelitian yang menjelaskan hubungan penelitian terdahulu dengan penelitian yang sedang dilakukan. Dalam matriks penelitian ini menunjukkan pendekatan yang berbeda-beda dalam menemukan sebuah solusi.

Tabel 2.3 Matriks Penelitian

No.	Penulis	Algoritma dan Metode														
		<i>Naïve Bayes</i>	<i>Multinomial Naïve Bayes</i>	<i>Bernoulli Naïve Bayes</i>	<i>Decision Tree</i>	KNN	<i>AdaBoost</i>	SVM	<i>Random Forest</i>	LDA	PSO	<i>VaderSentiment</i>	<i>Text Blob</i>	IG	TF-IDF	<i>Bag of Words</i>
1	(Rizki Setiawan, 2022)		✓		✓		✓					✓			✓	
2	(Doloksaribu & Samuel, 2022)	✓						✓							✓	
3	(Puspita & Widodo, 2021)	✓			✓	✓										
4	(Illia et al., 2021)										✓	✓				
5	(Zaidah, 2021)							✓		✓						
6	(Salma & Silfianti, 2021)	✓				✓		✓								
7	(Villavicencio et al., 2021)	✓														
8	(Shofiya & Abidi, 2021)							✓								
9	(Neogi et al., 2021)	✓			✓			✓	✓						✓	✓

No.	Penulis	Algoritma dan Metode														
		<i>Naïve Bayes</i>	<i>Multinomial Naïve Bayes</i>	<i>Bernoulli Naïve Bayes</i>	<i>Decision Tree</i>	KNN	<i>AdaBoost</i>	SVM	<i>Random Forest</i>	LDA	PSO	<i>VaderSentiment</i>	<i>Text Blob</i>	IG	TF-IDF	<i>Bag of Words</i>
10	(Sabrani et al., 2020)		✓													
11	(Wardani et al., 2020)		✓	✓												
12	(Syarifuddin, 2020)	✓			✓	✓										
13	(Mustopa et al., 2020)	✓						✓			✓					
14	(Fransiska & Irham Gufroni, 2020)							✓						✓		
15	(Guia et al., 2019)	✓			✓			✓	✓							
16	(Dini et al., 2015)	✓					✓							✓		