BAB II

LANDASAN TEORI

2.1 Penelitian Terkait

Tabel 2.1 Penelitian terkait dalam lima tahun terakhir

Penulis dan tahun	Judul	Jurnal	Metodologi dan hasil	Analisis kritis
Nur Syahirah Wan Min W, Zareen Zulkarnain N (2020)	Comparative Evaluation of Lexicons in Performing Sentiment Analysis [9]	Journal of Advanced Computing Technology and Application (JACTA)	VADER memiliki akurasi 79% dan TextBlob 73%	Proses pelabelan data dilakukan dengan *experts* hanya pada 300 sampel dari keseluruhan 7.997 *tweets*.
Ulya D, Kunaefi A, Rolliawati D, Nugroho B (2023)	Unpacking Public Perceptions of Qris with Twitter Data: A Vader and LDA Methodology [10]	Jurnal ELTIKOM	Akurasi label sentimen VADER 81,66%. Pada topik sentimen positif, *coherence score* 0,488. Sedangkan topik sentimen negatif sebesar 0,383	Sistem lainnya yang diluncurkan pemerintah belum dibahas
Bergamini Gomes G, Attux R (2023)	Contributions to Social Media Analysis Based on Topic Modeling [11]	Anais do XI Symposium on Knowledge Discovery, Mining and Learning	Rata-rata *topic coherence* LDA 0,565; BTM 0,609; NMF 0,651; BERTopic 0,778	Eksplorasi pemodelan topik pada dataset dalam tema berbeda dengan BERTopic
Baird A, Xia Y, Cheng Y (2022)	Consumer perceptions of telehealth for mental health or substance abuse: a Twitter-based topic modeling analysis [12]	JAMIA Open	*Coherence score* BERTopic mencapai 0,8 pada data pre pandemi dan lebih dari 0,6 pada data setelah pandemi.	Batasan berupa penggunaan kata kunci yang berkaitan dengan telehealth

Penulis dan tahun	Judul	Jurnal	Metodologi dan hasil	Analisis kritis
Imamah, Rachman F (2020)	Twitter Sentiment Analysis of Covid-19 using Term Weighting TF-IDF and Logistic Regression [13]	Information Technology International Seminar	Logistic Regression dengan akurasi 94,71%; *precision* 95%; *recall* 95%; *f1-score* 95%	Penelitian ini memiliki potensi keterbatasan karena hanya mengandalkan pembobotan dengan TF-IDF pada dataset Twitter yang membutuhkan detail lainnya.
Setiawan J, Gousander V, Prasetiawan I (2023)	Unmasking the Sentiments of Labuan Bajo: An Instagram-based Analysis for Tourism Insights through VADER Sentiment Analysis [14]	G-Tech: Jurnal Teknologi Terapan	VADER dengan akurasi sebesar 72,13%; presisi 97,43%; recall 98,1%; f-measure 97,76%	Meningkatkan jumlah dataset agar data *test* dapat merepresentasikan hasil yang dapat diandalkan.
Arya V, Mishra A, González-Briones A (2022)	Sentiment Analysis of Covid-19 Vaccine Tweets using Machine Learning and Vader Lexicon Method [21]	Advances in Distributed Computing and Artificial Intelligence Journal	Akurasi dengan VADER 64%; Random Forest 90%; Logistic Regression 92%	Tidak disebutkan label untuk menghitung akurasi dan belum ada penanganan *imbalanced data*.
Fajri F, Tutuko B, Sukemi S (2022)	Membandingkan Nilai Akurasi BERT dan DistilBERT pada Dataset Twitter [22]	Jurnal Sistem Informasi	Analisis sentimen BERT akurasi 87%; *precision* 91%; *recall* 91%; f1-score 89% DistilBERT akurasi 97%; *precision* 99%; *recall* 99%; f1-score 99%	Eksplorasi model DistilBERT pada dataset dan domain pengetahuan lainnya untuk analisis sentimen media sosial.
Egger R, Yu J (2022)	A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts [23]	Frontiers in Sociology	NMF dan BERTopic paling cocok untuk pemodelan topik pada data Twitter daripada LDA dan Top2Vec.	Memanfaatkan strategi *parameter tuning* agar hasil BERTopic yang tidak terlalu luas

Penulis dan tahun	Judul	Jurnal	Metodologi dan hasil	Analisis kritis
Chiny M, Chihab M, Chihab Y (2021)	LSTM, VADER, and TF-IDF based Hybrid Sentiment Analysis Model [24]	IJACSA	Model *hybrid* LSTM, VADER, dan TF-IDF membantu meningkatkan *classifier* biner dengan rata-rata 9.51% pada data IMDB dan 12,58% pada data *US airline tweets*.	Memanfaatkan model *deep learning* berbasis *self attention* untuk meningkatkan performa klasifikasi sentimen
Jin Z, Lai X, Cao J (2020)	Multi-label Sentiment Analysis base on BERT with modified TF-IDF [25]	IEEE International Symposium on Product Compliance Engineering -Asia	Model BERT dengan TF-IDF menghasilkan akurasi 64%	Penambahan *feature* TF-IDF sedikit meningkatkan performa model *hybrid* ini sehingga memungkinkan penggabungan kedua *feature* ini belum cukup menangkap informasi tambahan untuk klasifikasi sentimen.

Peninjauan terhadap penelitian terkait dilakukan untuk mendapat informasi mengenai algoritma untuk analisis sentimen dan pemodelan topik terkait Coretax pada media sosial X. Proses peninjauan dilakukan dengan metode *Literature Review* pada *database* jurnal melalui aplikasi Publish or Perish. Kriteria pencarian jurnal menggunakan *kata kunci* berbahasa Inggris seperti *sentiment analysis*, pemodelan topik, dan Twitter. Rentang waktu dibatasi hanya pada tahun 2020 hingga 2025 agar penelitian yang ditemukan tetap relevan.

Tabel 2.1 menunjukkan beberapa penelitian terkait dalam lima tahun terakhir. Penelitian terkait menunjukkan ragam variasi metodologi dalam analisis sentimen. Metode berbasis BERT [22] terbukti lebih unggul dibandingkan VADER, Random Forest, dan Logistic Regression [13], [21] dalam klasifikasi sentimen X berkaitan Covid-19. Namun dalam konteks *slang*, singkatan kata, dan emoji VADER

memiliki akurasi tinggi dibandingkan TextBlob [9]. Selain itu, VADER juga menunjukkan kelebihannya pada klasifikasi sentimen dalam konteks persepsi publik pada destinasi wisata [14], sistem pembayaran [10] dan kebijakan pemerintah [16]. Performa model *hybrid* pada [24] menunjukkan adanya kesempatan untuk meningkatkan akurasi model VADER untuk analisis sentimen. Model *hybrid* lainnya antara BERT dan TF-IDF menghasilkan akurasi yang lebih tinggi dibandingkan dengan hanya memanfaatkan *feature* BERT pada Logistic Regression [25]. Pada sisi lain, terlihat bahwa metode LDA menjadi algoritma yang cukup umum digunakan dalam pemodelan topik. Namun LDA menghasilkan *coherence score* yang cukup rendah dalam konteks persepsi publik terhadap sistem pembayaran digital pemerintah [10]. Selain itu, LDA menunjukkan kelemahan dalam mengidentifikasi topik pada data multilingual dengan kinerjanya paling rendah [11]. Egger dan Yu dalam penelitiannya [23] mengungkapkan algoritma NMF dan BERTopic menjadi metode pemodelan topik yang paling cocok untuk data X. Namun dalam konteks persepsi publik, BERTopic menghasilkan *coherence* score yang cenderung tinggi [12] dibandingkan dengan metode NMF [11].

Penelitian ini menggunakan metode dan fokus utama yang membedakannya dengan studi terdahulu. Penelitian ini berfokus pada analisis opini publik terhadap Coretax yang belum pernah dieksplorasi dalam literatur sebelumnya. Selain itu, penelitian ini menggunakan *automatic labeling* menggunakan Distilled Bidirectional Encoder Representations from Transformers (DistilBERT). Pemilihan DistilBERT didasarkan pada performanya yang tinggi dalam klasifikasi sentimen pada penelitian [22]. Berbeda dengan penelitian terkait, penelitian ini mengimplementasikan *spellchecking* pada tahap *preprocessing* untuk meningkatkan kualitas data untuk analisis sentimen. Pendekatan analisis sentimen dilakukan secara *hybrid* untuk mengurangi keterbatasan pada masing-masing algoritma mengacu variasi performa yang ditunjukkan Tabel 2.1. Model *hybrid* ini menggabungkan VADER, TF-IDF, dan BERT Embeddings dengan Logistic Regression sebagai *classifier*. Sementara itu penerapan pemodelan topik dilakukan

dengan metode BERTopic karena performanya ditunjukkan dengan *coherence score* yang tinggi.

2.2 Teori Penelitian

2.2.1 Analisis sentimen

Analisis sentimen merupakan sebuah studi mengenai opini publik, perasaan, sentimen, dan sikap terhadap suatu objek tertentu menggunakan tulisan [26]. Klasifikasi sentimen bertujuan untuk memberikan *label* pada data teks sebagai positif, negatif, atau netral [27]. Dalam konteks layanan publik, mengetahui sentimen publik terhadap layanan dapat membantu proses dalam pengambilan keputusan terhadap layanan tersebut. Dengan demikian diharapkan Coretax dapat berfungsi sebagaimana tujuan awal pengembangan sistem tersebut ditetapkan.

2.2.2 Pemodelan Topik

Pemodelan Topik merupakan sebuah pendekatan *unsupervised machine learning* dengan tujuan untuk mengkategorikan data dalam jumlah besar dengan mengidentifikasi topik yang muncul dalam dokumen [28]. Kata-kata dalam sebuah dokumen umumnya berkaitan dengan tema-tema tertentu sehingga dapat merepresentasikan topik utamanya [29]. Misalnya dalam sebuah dokumen yang membahas mengenai resep memasak berisi kata-kata "sendok," "aduk," dan "matang."

2.2.3 Pre-processing

Proses ini seringkali bersifat iteratif dan dinamis, sehingga memerlukan teknik pemrosesan data dengan menghapus data yang tidak relevan, *text filtering*, *translate* bahasa Inggris [30], *tokenization* [31], *case folding* [16], dan *lemmatization* [32], dan *remove stopwords* [33].

1. *Text filtering*. Tahap ini terdiri dari *tasks* yang bertujuan untuk menghasilkan teks secara utuh secara makna kontekstual. Pada *text filtering* dilakukan *cleaning* untuk menghapus URL, *mention username,* karakter spesial, tanda baca [10], [21], [22], simbol dan

angka pada data teks. Kemudian menghapus data duplikat, spam, dan *tweet* kosong.

2. *Translate* bahasa Inggris. Seluruh *tweet* diterjemahkan menjadi bahasa Inggris dengan *library Google Translate* pada Python. Tahap ini sangat penting dilakukan karena leksikon VADER dibuat berdasarkan *corpus* bahasa Inggris.
3. *Case folding.* Dalam sebuah data tekstual terdapat kata-kata dengan huruf besar dan kecil yang bisa saja tidak beraturan. Data tersebut kemudian diubah seluruhnya ke huruf kecil untuk normalisasi bentuk kalimat.
4. *Tokenization*. Memecah sebuah teks menjadi kata-kata terpisah yang umumnya disebut sebagai token.
5. *Lemmatization*. Proses mengubah sebuah kata kembali ke bentuk dasarnya sesuai *entry* dalam kamus dengan mempertimbangkan konteks dan *part of speech* (POS). Dibandingkan dengan *stemming* yang hanya menghilangkan *affix* dari kata tersebut. Misalnya kata "*better*" diubah menjadi "*good*" oleh *lemmatization*, sedangkan *stemming* mengubahnya menjadi "*bett*." Keunggulan *lemmatization* terletak pada akurasi linguistik, namun proses ini meningkatkan kompleksitas komputasi.
6. *Remove stopwords*. Dalam data tekstual umumnya terdapat kata yang frekuensi kemunculannya tinggi namun tidak memiliki nilai informatif dalam analisis teks, seperti kata ganti, kata depan, dan konjungsi. Tujuan tahap ini yaitu agar dapat berfokus pada kalimat yang memiliki nilai dan makna pada teks sehingga dapat membantu meningkatkan performa analisis dan ekstraksi informasi.

2.3 Framework dan Algoritma Penelitian

2.3.1 KDD

Dalam *data mining* umumnya tedapat beberapa jenis kerangka kerja yang digunakan seperti *Cross-Industry Standard Process for Data Mining* (CRISP-DM); *Sample*, *Explore*, *Modify*, *Model*, *Assess* (SEMMA); dan *Knowledge Discovery in Database Process* (KDD). Tabel 2.2 menunjukkan perbandingan *framework* yang umumnya digunakan dalam penelitian berbasis *data science*.

Tabel 2.2 Perbandingan framework CRISP-DM, SEMMA, dan KDD

Pembeda	CRISP-DM	SEMMA	KDD
Fokus utama	Menekankan pada pemahaman tujuan dan kebutuhan bisnis serta *domain knowledge*.	Fokusnya eksplorasi dan modifikasi data untuk model prediktif.	Berfokus pada keseluruhan proses *knowledge discovery*.
Alur proses	*Iterative*	*Sequential*	*Iterative*
Tahap	1. Business understanding 2. Data understanding 3. Data preparation 4. Modeling 5. Evaluation 6. Deployment	1. Sample 2. Explore 3. Modify 4. Model 5. Assess	1. Selection 2. Cleaning 3. Transformation 4. Data Mining 5. Interpretation/ Evaluation

Berdasarkan perbandingan yang ditunjukkan pada Tabel 2.2, implementasi *framework* KDD menjadi yang paling tepat dengan tujuan penelitian. Proses yang berfokus pada analisis sentimen dan pemodelan topik pada media sosial X selaras dengan fokus utama metode tersebut. Pada penelitian ini dilakukan analisis sentimen dengan *features* VADER, TF-IDF, dan BERT dalam *classifier* Logistic Regression dan algoritma BERTopic pada pemodelan topik.

2.3.2 TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) merupakan salah satu metode statistik yang umum digunakan dalam analisis teks dan NLP. Metode ini menggabungkan Term Frequency (TF) dan Inverse Document

Frequency (IDF). TF menghitung frekuensi relatif kemunculan suatu kata dalam sebuah dokumen tertentu, sedangkan IDF mengukur tingkat kelangkaan kata tersebut dalam seluruh *corpus* [18]. Kombinasi kedua metrik ini menentukn bobot relatif yang merepresentasikan seberapa penting kata tersebut dalam dokumen secara relatif terhadap keseluruhan *corpus* [34]. Dengan TF-IDF, setiap dokumen kemudian diwakili oleh *vector* dengan dimensi sesuai jumlah kata-kata yang unik dalam *corpus*. Setiap elemen dalam *vector* ini terdiri dari nilai TF-IDF dari data terkait [35]. Secara matematis, pembobotan TF-IDF menggunakan rumus sebagai berikut.

$$tf(t,d) = \frac{\text{jumlah kemunculan } t \text{ pada } d}{\text{jumlah kata pada } d}$$

Rumus 2.1 Rumus *term frequency*

$$idf(t,D) = \log\left(\frac{\text{jumlah } d \text{ dalam } D}{\text{jumlah } d \text{ yang mengandung } t}\right)$$

Rumus 2.2 Rumus *inverse document frequency*

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D)$$

Rumus 2.3 Rumus TF-IDF

Keterangan:

1. t merupakan kata-kata yang ada
2. d merupakan dokumen
3. D merupakan kumpulan dokumen (*corpus*)

2.3.3 SMOTE

SMOTE (Synthetic Minority Over-sampling Technique) merupakan teknik *oversampling* untuk menangani ketidakseimbangan kelas [36]. Mekanisme SMOTE bekerja dengan cara mengidentifikasi *k-nearest neighbors* dari setiap sampel pada *class* minoritas, lalu membuat data sintetik baru berdasarkan interpolasi linier dalam *feature space* [36], [37]. Proses ini meningkatkan representasi *class* minoritas tanpa sekadar melakukan duplikasi, sehingga mengurangi resiko *overfitting* pada metode *oversampling* [37].

Dalam konteks penelitian ini, implementasi SMOTE bertujuan untuk meningkatkan performa klasifikasi dengan menyeimbangkan distribusi kelas pada dataset [38].

2.3.4 BERT

BERT (Bidirectional Encoder Representations from Transformers) merupakan model yang dibangun dalam arsitektur model transformer yang dikembangkan oleh Devlin et al. [19]. Model ini menerapkan mekanisme *multi head self-attention* yang dapat menganalisis urutan dalam sebuah teks. [19], [39], [40]. Model ini terdiri dari *encoder* yang memiliki beberapa lapisan, masing-masing menggabungkan *self-attention* dan *neural networks*. Arsitektur ini memungkinkan BERT secara efektif menangkap hubungan kontekstual dalam sebuah teks [40]. BERT memiliki bentuk representasi berupa *vector* yang telah melalui proses *tokenization* dengan Wordpiece Tokenizer. *Tokenizer* ini memecah kata-kata dan memberi token khusus seperti [CLS] dan [SEP]. Penempatan masing-masing token tersebut berada pada awal dan akhir kalimat. Setiap token kemudian direpresentasikan dalam 3 jenis embedding, yaitu token embedding, segment embedding, dan position embedding. Ketiga jenis embedding ini nantinya dijumlahkan untuk membentuk *input* ke dalam model BERT [19]. Dalam penelitian ini, *vector* embedding dari BERT ini digunakan sebagai input untuk algoritma klasifikasi *Logistic Regression* dalam kerangka model hybrid untuk analisis sentimen.

2.3.5 DistilBERT

DistilBERT (Distilled Bidirectional Encoder Representations from Transformers) secara arsitektur lebih kecil 40% daripada BERT, dengan lapisan dan parameter yang lebih sedikit sambil tetap mempertahankan 97% kemampuan pemahaman pada BERT [41]. DistilBERT menggunakan mekanisme *multi-head self-attention* dari BERT, yang memungkinkan model untuk memahami koneksi kontekstual di antara kata-kata dalam kalimat [41]. Mekanisme ini memiliki peran penting dalam membedakan sentimen pada data

tekstual. Lapisan *multi-head attention* layer memungkinkan model untuk berfokus pada berbagai elemen teks secara bersamaan, sehingga dapat mengklasifikasikan sentimen dengan performa yang tinggi [22], [42].

2.3.6 VADER

VADER (Valence Aware Dictionary and sEntiment Reasoner) adalah salah satu metode analisis sentiment berbasis leksikon yang dioptimalkan untuk media sosial. Leksikon ini secara efektif mengatasi kompleksitas konten media sosial seperti *slang*, singkatan kata, dan emoji. [17], [43]. Fungsi utama VADER yaitu untuk menentukan polaritas sentimen dalam sebuah kata berdasarkan kombinasi fitur leksikal dan tata bahasa [17]. Kategori sentimen dengan leksikon VADER dihitung berdasarkan jumlah seluruh *valence score* setelah dinormalisasikan dalam rentang -1 hingga +1. VADER memiliki *rule* yang memperhitungkan penegasan kata dan urutan kata yang mempengaruhi intensitas sebuah kata. Tabel 2.3 menunjukkan skor kata-kata dalam leksikon VADER yang memiliki intensitas sentimen dalam rentang -4 hingga +4.

Tabel 2.3 Contoh *valence score* dalam leksikon VADER

Kata	***Valence Score***
great	3.1
kudos	2.3
good	1.9
horrible	-2.5
crisis	-3.1

2.3.7 Logistic Regression

Logistic Regression merupakan sebuah metode dalam analisis statistik untuk menjelaskan hubungan variabel dependen biner dengan satu atau lebih variabel independen [44]. Dalam konteks *machine learning*, algoritma ini dikategorikan sebagai *supervised learning* untuk klasifikasi biner. Model *logistic regression* dilatih untuk memprediksi probabilitas suatu data untuk masuk sebagai satu dari dua *target class* berdasarkan fitur yang ada [45]. Dibandingkan dengan *linear regression*, output yang dihasilkan berupa

probabilitas antara 0 dan 1 melalui fungsi sigmoid [46]. Berikut merupakan rumus fungsi *sigmoid* yang menghubungkan teks menjadi kategori sentimen.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Rumus 2.4 Rumus sigmoid

Keterangan:

1. $\sigma(z)$ merupakan probabilitas antara 0 dan 1.
2. e merupakan bilangan Euler untuk membentuk kurva S
3. z terdiri dari fungsi linier $B_0 + B_1X_1 + \ldots + B_nX_n$
4. $B_0, B_1, \ldots B_n$ sebagai koefisien model yang diestimasi
5. $X_1, \ldots X_n$ sebagai variabel prediktor

2.3.8 BERTopic

BERTopic merupakan salah satu teknik dalam pemodelan topik secara *unsupervised* yang memanfaatkan *embeddings* berbasis *transformers* . Metode ini memetakan teks ke dalam *vector* menggunakan *sentence embeddings*, dan melakukan *dimension reduction* dengan Uniform Manifold Approximation and Projection (UMAP), mengelompokkan dokumen melalui Hierarchical Density-Based Spatial Clustering of Application with Noise (HDBSCAN), dan mengekstrak representasi topik dengan pendekatan *class based* TF-IDF [20]. Arsitektur dari BERTopic inilah yang mendukung implementasinya dalam berbagai jenis dataset [11], [12], [47] dalam mengelola topik dalam jumlah banyak dan teks tidak terstruktur [23], [48].

2.3.9 Confusion Matrix

Confusion Matrix merupakan tabel yang merepresentasikan performa model dalam mengklasifikasikan data. Tabel 2.4 menunjukkan 4 kuadran yang membandingkan jumlah *class* hasil prediksi dan *class* yang sebenarnya. *Confusion matrix* pada penelitian ini berfungsi sebagai alat ukur yang terdiri

dari *accuracy*, *recall*, *precision*, dan *f1-score*. Berikut penjelasan dan rumus perhitungan masing-masing metrik [49].

Tabel 2.4 Tabel Confusion Matrix

		Predicted class	
		Positive	***Negative***
Actual class	***Positive***	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

1. *Accuracy* merupakan rasio prediksi benar pada *class* positif dan negatif terhadap keseluruhan dataset. Meskipun metrik ini umum digunakan untuk mengevaluasi performa model klasifikasi, interpretasi hasilnya menjadi tidak representatif pada data yang tidak seimbang karena gagal mempertimbangkan distribusi *class*-nya. Berikut adalah rumus metrik *accuracy*.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Rumus 2.5 Rumus metrik *accuracy*

2. *Recall* merupakan rasio prediksi benar pada *class* positif terhadap jumlah seluruh data aktual pada *class* positif. Tujuan utama penggunaan *recall* ialah untuk memastikan kemampuan model dalam mengidentifikasi seluruh *class* positif secara maksimal. Rumus matematis dari metrik *recall* yaitu sebagai berikut.

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2.6 Rumus *recall*

3. *Precision* merupakan hasil prediksi benar *class* positif terhadap jumlah seluruh data prediksi pada *class* positif. Rumus dari metrik *precision* yaitu sebagai berikut.

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

Rumus 2.7 Rumus *precision*

4. *F1-Score* merupakan rasio dari *harmonic mean* dari *precision* dan *recall*. Metrik ini memberikan gambaran terhadap performa model dalam *precision* dan *recall*. Rumus dari *f1-score* yaitu sebagai berikut.

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Rumus 2.8 Rumus *f1-score*

2.3.10 Coherence score

Coherence score merupakan metrik evaluasi yang digunakan untuk mengukur kualitas sebuah topik dengan menilai keterkaitan semantik di antara kata-kata kunci yang berkaitan erat dengan topik [48]. Salah satu metode *coherence score* yang paling banyak digunakan adalah C_v *coherence* karena menggabungkan pendekatan statistik dan pembobotan berbasis probabilitas, serta terbukti sesuai dengan persepsi manusia sebagai pembaca [20]. Berikut merupakan formulasi rumus *topic coherence*.

1. Segmentasi kata

$$P = \{(w_i, w_j) | 1 \leq i < j \leq n\}$$

Rumus 2.9 Rumus segmentasi kata *coherence score*

Keterangan:

a. $W = \{w_1, w_2, \dots, w_n\}$ daftar kata dalam topik
b. P himpunan pasangan kata unik

2. Pembentukan Vektor konteks kata

$$V(w_i) = [f(w_1, w_i), f(w_2, w_i), \dots, f(w_m, w_i)]$$

Rumus 2.10 Rumus pembuatan vektor konteks data

Keterangan:

a. $f(w_k, w_i)$ frekuensi kemunculan kata w_k dalam konteks kata w_i

b. $V(w_i)$ *vector* konteks kata w_i yang menangkap makna semantik

3. Penghitungan *cosine similarity*

$$\text{sim}(w_i, w_j) = \frac{V(w_i) \times V(w_j)}{\|V(w_i)\| \times \|V(w_j)\|}$$

Rumus 2.11 Rumus cosine similarity *coherence score*

Keterangan:

a. $\|V(w_i)\|$ Norma *vector* $V(w)$

b. Nilai $sim(w_i, w_j)$ berada dalam rentang antara 0 hingga 1

4. Agregasi *Coherence score*

$$Coherence = \frac{1}{|P|} \sum_{(w_i, w_j) \in P} \text{sim}(w_i, w_j)$$

Rumus 2.12 Rumus agregasi *coherence score*

Keterangan:

a. $|P| = \frac{n(n-1)}{2}$ jumlah pasangan kata unik

b. Skor akhir pada $Coherence$ berada pada rentang 0 hingga 1

2.4 Tools Penelitian

2.4.1 Google Colab

Google Colab merupakan versi *cloud-based tool* dari Jupyter Notebook yang dapat diakses melalui browser. Google Colab dapat melakukan komputasi *machine learning* di server Google. Software ini dipilih karena menyediakan akses gratis terhadap GPU yang dapat mempercepat proses model dan integrasi dengan layanan Google Drive. Penggunaan runtime GPU pada Google Colab sangat membantu dalam mengolah dataset dan menjalankan model *Deep Learning* seperti BERT dengan waktu yang lebih singkat. Bahasa pemrograman yang digunakan pun sama, yaitu Python. Dengan menggunakan Google Colab, maka proses analisis dan *modeling* data dapat dijalankan secara

efisien dan cepat. Salah satu kelebihan dari *tool* ini yaitu karena gratis dan tidak perlu mengunduh *software* tambahan lagi sehingga tidak bergantung pada performa komputer.