

## BAB II

### LANDASAN TEORI

#### 2.1 *Text Mining*

*Text mining* merupakan proses penambangan data teks dari format tidak terstruktur menjadi format terstruktur untuk mengidentifikasi pola yang ada [13]. Tujuan utama *text mining* adalah memperoleh dan mengekstrak informasi berguna dari teks untuk dipakai pada tugas lebih lanjut. Perolehan informasi tersebut didapatkan melalui penemuan pola maupun tren dengan cara pembelajaran pola, pemodelan topik, pemodelan bahasa statistik, dan lainnya [14]. Proses *text mining* dilakukan berdasarkan kombinasi kata, frase, serta melibatkan ekstraksi fitur yang menjadi kunci utama dalam pelatihan model [15].

*Text mining* memerlukan penataan pada teks yang dijadikan *input* karena memiliki format tidak terstruktur. Oleh karena itu, harus dilakukan *text pre-processing* yang merupakan proses untuk membersihkan dan mengubah data teks menjadi format yang dapat digunakan (format terstruktur). Tahapan *pre-processing* dibagi sebagai berikut:

a. *Case Folding*

Data teks memiliki keberagaman kapitalisasi yang dapat menjadi masalah saat mengklasifikasikan data berukuran besar [16]. Pendekatan umum untuk menangani kapitalisasi yang tidak konsisten adalah dengan menyamaratakan seluruh karakter ke dalam penggunaan huruf yang sama, yaitu huruf kecil [16]. Selain itu, proses penghapusan tanda baca, angka, spasi berlebih, serta karakter *single* diperlukan untuk mengurangi *noise*.

b. *Tokenization*

Tokenisasi adalah proses pemecahan teks kalimat yang berbentuk panjang menjadi kata-kata yang biasa disebut *token* [13]. Proses ini menyelidiki setiap kalimat dan membuat daftar *token* yang dapat digunakan sebagai masukan untuk algoritma selanjutnya [17]. Tujuan utama dari langkah ini adalah menyelidiki kata-kata dalam sebuah kalimat [16].

c. *Normalization*

Proses ini dilakukan untuk menormalisasi bahasa yang tidak baku seperti bahasa gaul atau kata *slang* dan kata-kata singkat menjadi baku. Normalisasi yang dimaksud adalah pengembalian bentuk penulisan menjadi kata yang sesuai pada Kamus Besar Bahasa Indonesia [18].

d. *Filtering*

Proses filterisasi mencakup tahapan seperti penghapusan *stopword* yang merupakan kata-kata yang tidak memiliki informasi atau tidak penting dalam analisis. Dengan ini, dimensionalitas teks dapat berkurang tanpa mengurangi isi teks tersebut [17].

e. *Stemming*

Proses *stemming* merupakan pencarian *stem* atau kata dasar dengan mentransformasi kata yang memiliki imbuhan maupun akhiran dan mencocokkannya dengan kata dasar [17].

Adapun tugas *text mining* yang umum dilakukan adalah *text categorization*, *text clustering*, *concept/entity extraction*, *sentiment analysis*, *document summarization*, dan juga *entity-relation modeling* [14]. Pada tugas *sentiment analysis*, penggunaan *text mining* dimanfaatkan untuk menganalisis teks dan menentukan apakah teks tersebut memiliki sentimen positif, negatif, netral, dan lainnya.

## 2.2 *Text Classification*

Klasifikasi teks, yang juga dikenal sebagai kategorisasi teks, adalah penggunaan umum dalam *Natural Language Processing* (NLP) yang bertujuan untuk menetapkan label atau *tag* ke unit tekstual seperti kalimat, kueri, paragraf, dan dokumen [19]. Adapun NLP merupakan cabang kecerdasan buatan (*artificial intelligence*) yang memungkinkan komputer mampu memahami teks maupun kata-kata yang diucapkan [20]. Beranjak dari terobosan NLP dan *text mining*, banyak peneliti tertarik untuk mengembangkan aplikasi yang memanfaatkan metode klasifikasi teks [16]. Klasifikasi teks biasanya diaplikasikan untuk deteksi spam, analisis sentimen, kategorisasi berita, klasifikasi topik, dan sebagainya [19].

Pada [16], secara umum klasifikasi teks dapat dilakukan pada 4 level sebagai berikut:

1. Level dokumen

Pada level ini algoritma memperoleh kategori yang relevan dari dokumen secara lengkap.

2. Level paragraf

Pada level paragraf, algoritma memperoleh kategori yang relevan dari satu paragraf (sebagian dari isi dokumen).

3. Level kalimat

Pada level kalimat, diperoleh kategori yang relevan dari satu kalimat (sebagian dari paragraf).

4. Level sub-kalimat

Pada level sub-kalimat, algoritma memperoleh kategori sub-ekspresi yang relevan dalam sebuah kalimat (sebagian kalimat).

Klasifikasi teks dapat dilakukan dengan anotasi secara manual atau dengan pelabelan secara otomatis. Seiring meningkatnya skala data teks dalam industri, klasifikasi teks otomatis menjadi pilihan utama [19]. Model yang dipakai pada klasifikasi teks secara umum dibagi ke 2 kategori yaitu *machine learning* dan *deep learning* [21].

Banyak penelitian yang menggunakan model *machine learning* mendapatkan bahwa algoritma yang dipakai memiliki efisiensi dan stabilitas yang tinggi, namun masih terbatas pada *dataset* yang berskala besar [21]. Adapun pada model *deep learning*, dicapai kinerja yang sangat baik dalam skala data yang besar meskipun cenderung relatif lambat selama pelatihan maupun pengujian modelnya [21].

### 2.3 Aspect-Based Sentiment Analysis (ABSA)

*Aspect-Based Sentiment Analysis* atau ABSA, adalah jenis analisis sentimen yang dapat menentukan sentimen di setiap aspek yang ditentukan [10]. ABSA merupakan solusi dari masalah analisis sentimen yang tidak dapat mengklasifikasikan ulasan ke dalam kelas positif atau negatif berdasarkan kategori

aspek [9]. ABSA memproses informasi pada level sub-kalimat atau level aspek. Hingga saat ini, ABSA dapat dilakukan dengan berbagai metode.

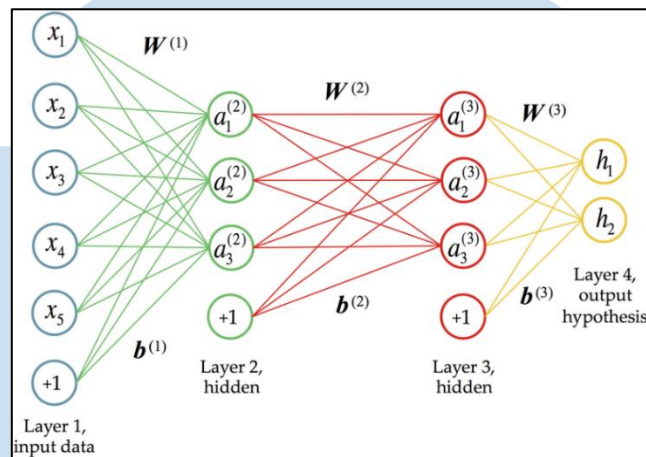
Pada beberapa penelitian yang dilakukan, proses dalam ABSA dibagi menjadi dua tugas yaitu tugas ekstraksi aspek dan mengestimasi polaritas/*rating* [8]. Tugas pertama yaitu pengekstrakan aspek, bertujuan untuk mengekstrak kata/aspek dari ulasan produk dan mengelompokkan sinonim dari setiap aspeknya dikarenakan tiap orang dapat menggunakan frasa yang berbeda yang merujuk pada aspek yang sama [8]. Tugas kedua yaitu estimasi *rating*/polaritas yang bertujuan untuk menentukan sentimen pada suatu aspek, apakah positif, negatif, atau netral [8]. Pada metode ini, ABSA mendeteksi polaritas dan mengekstrak aspek terlebih dahulu dan kemudian mengklasifikasikannya sebagai positif atau negatif [22].

Selain menggunakan metode pengekstrakan aspek, ABSA juga dilakukan dengan cara lain dikarenakan proses pengekstrakan aspek membutuhkan banyak sumber daya [10]. Proses ABSA dapat dilakukan dengan pengklasifikasian aspek dan juga sentimen. Pada metode ini, ABSA berperan dalam mengklasifikasikan teks ulasan ke dalam kelas positif atau negatif berdasarkan aspeknya, di mana model akan mengklasifikasikan dokumen teks ke dalam aspek kategori kemudian kecenderungan sentimennya [9]. Misalnya, dalam kalimat ulasan “Harga makanannya cukup mahal”, model ABSA mampu mengklasifikasikan kalimat tersebut ke dalam aspek harga dan kelas sentimen negatif [9]. Adapun metode ini memerlukan pemodelan dari data teks berlabel yang kemudian dapat diterapkan pada data tidak berlabel.

#### **2.4 Deep Learning**

*Deep learning* adalah salah satu teknik *machine learning* di bidang *artificial intelligence* yang menyempurnakan algoritma *machine learning* dengan kemampuannya dalam memahami suku kata dan gambar [23]. *Deep learning* menggunakan strategi *supervised* dan/atau *unsupervised* dalam mempelajari representasi arsitektur untuk klasifikasi [23]. Terinspirasi oleh pengamatan biologis pada mekanisme otak manusia untuk memproses sinyal alami, *deep learning* menarik banyak perhatian komunitas akademik dan ilmiah karena kinerjanya yang

baik dan canggih di banyak domain penelitian seperti *speech recognition*, *computer vision* dan juga pada *natural language processing* [23].



Gambar 2.1 Arsitektur pada Deep Learning [24]

*Deep learning* menggunakan beberapa *layer* pemrosesan untuk menemukan pola dan struktur dalam kumpulan data, di mana tiap *layer* akan mempelajari konsep dari data yang menjadi dasar untuk *layer* berikutnya; semakin tinggi tingkatannya maka semakin abstrak pula konsep yang dipelajari [25]. *Layer* pertama adalah *layer input* data, *layer* berikutnya yaitu *hidden layer* di mana pemrosesan utama terjadi, dan *layer* terakhir untuk memprediksi *output* [24].

*Deep learning* dianggap penting dalam konteks *big data* karena dapat mengekstrak informasi tingkat tinggi dari volume data yang sangat besar [25]. *Deep learning* secara efektif telah digunakan dalam industri yang memanfaatkan volume data yang besar seperti perusahaan Google, Apple, dan Facebook [23].

## 2.5 Convolutional Neural Network (CNN)

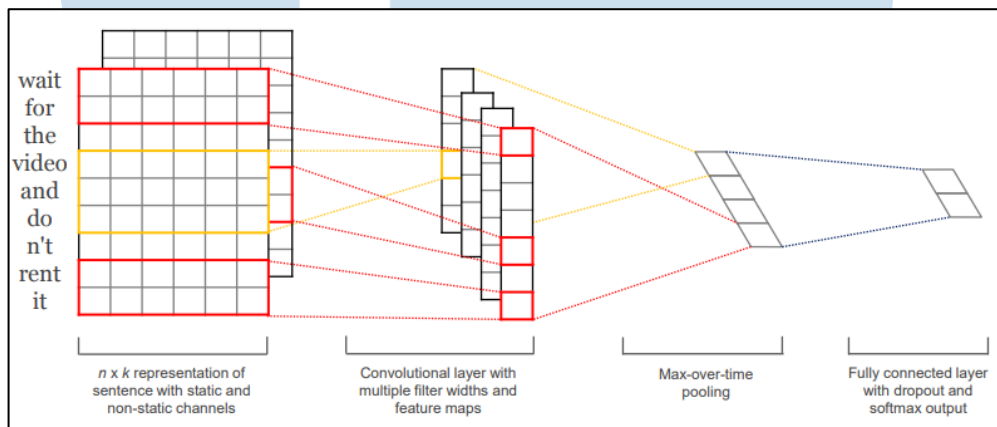
### 2.5.1 Algoritma CNN

*Convolutional Neural Network* atau CNN merupakan algoritma *deep learning* yang umum digunakan dalam *computer vision* dan pengolahan citra seperti klasifikasi citra, dan pendeteksian objek [10]. Model CNN kemudian terbukti efektif dalam *Natural Language Processing* (NLP) dan telah mencapai

hasil yang baik dalam *semantic parsing* (penguraian semantik), *sentence modeling* (pemodelan kalimat), dan *text classification* (klasifikasi teks) [26].

### 2.5.2 Layer CNN

Model CNN umumnya terdiri dari *layer convolutional*, *pooling*, dan *fully connected layer*. Namun untuk mengatasi masalah klasifikasi teks, model CNN perlu ditambahkan *embedding layer* [10]. CNN menggunakan *layer* dengan *filter* konvolusi yang diterapkan ke fitur lokal [26]. Gambar 2.2 menunjukkan arsitektur yang dipakai dalam melakukan klasifikasi teks menggunakan CNN.



Gambar 2.2 Arsitektur CNN pada Klasifikasi Teks [26]

Berikut penjelasan dari setiap *layer* yang digunakan pada CNN khususnya dalam melakukan klasifikasi teks:

1. *Embedding layer*

*Layer* ini berfungsi untuk memetakan *input* berbentuk indeks kosakata (*vocabulary word indices*) ke dalam vektor berdimensi rendah [27]. Pada dasarnya *layer* ini merupakan tabel pencarian yang dipelajari dari teks *input* yang diberikan. Adapun ukuran kosakata (*vocabulary*) ditentukan dari panjang kalimat maksimum. Setelah semua kata ditransformasi menjadi vektor, maka akan diumpangkan ke *convolutional layer* [27].

## 2. *Convolutional layer*

*Layer* ini merupakan *layer* pemrosesan utama model berlangsung, yang melakukan proses konvolusi terhadap *input* dan *filter* [11]. Ketika *input* masuk ke *layer* ini, operasi konvolusi yang melibatkan *filter* diterapkan pada jendela kata untuk menghasilkan fitur baru, kemudian *filter* diaplikasikan secara berulang pada setiap jendela kata dalam kalimat sehingga menghasilkan *feature maps* [26].

## 3. *Pooling Layer*

*Layer* ini berfungsi untuk secara bertahap mengurangi jumlah parameter, kompleksitas komputasi model, dan mengontrol *overfitting* [28]. Operasi yang digunakan adalah *max-over-time pooling* yang diterapkan pada *feature maps*, dengan tujuan untuk mengambil fitur yang paling penting (fitur dengan nilai tertinggi) untuk setiap *feature maps* [26].

## 4. *Fully-connected Layer* atau *Dense Layer*

*Layer* ini merupakan *layer* CNN yang membentuk *neuron* satu dimensi, dan terdiri dari *neuron* yang saling berhubungan dengan *neuron* pada lapisan sebelumnya dan selanjutnya [11]. Di *layer* ini dapat dilakukan regularisasi dengan fungsi *dropout* yang menjaga *neuron* agar tetap berada dalam nilai probabilitas antara 0 dan 1, yang memudahkan penggolongan kelas *output* [29]. *Layer* ini juga akan menghasilkan *output* dari banyaknya kelas yang ditentukan menggunakan aktivasi *softmax*.

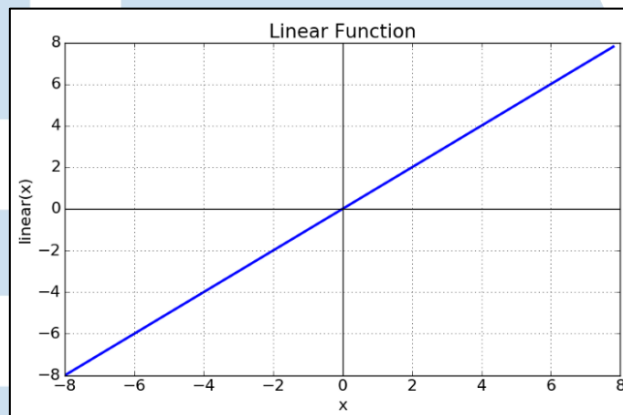
### 2.5.3 *Activation Function*

Di setiap *layer* pada *neural network* terdapat operasi yang disebut sebagai fungsi aktivasi. Fungsi aktivasi pada *hidden layer* digunakan untuk mengontrol seberapa baik model *neural network* belajar dari data pelatihan, sedangkan pada *output layer* fungsi aktivasi digunakan untuk mendefinisikan *output* dari *neural network*.

Fungsi aktivasi di bagi menjadi 2 tipe sebagai berikut:

### 1. *Linear Activation Function*

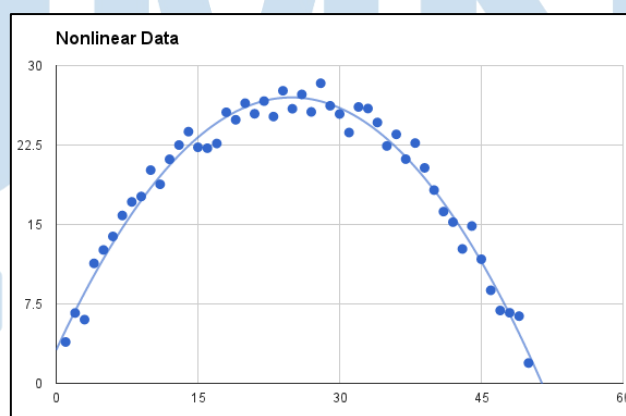
Fungsi ini merupakan fungsi *linear* di mana *output*-nya tidak dibatasi oleh rentang apapun sehingga tidak mendukung kompleksitas atau berbagai parameter data yang biasanya dipakai pada *neural network* [30].



Gambar 2.3 Fungsi Aktivasi Linear [30]

### 2. *Non-linear Activation Function*

Fungsi ini merupakan tipe fungsi aktivasi yang banyak digunakan pada *neural network* karena memudahkan adaptasi dengan berbagai data dan untuk membedakan antara *output* dengan memetakan *input* dan *output* pada *neural network* [30]. Berikut grafik yang dihasilkan fungsi aktivasi *non-linear*:



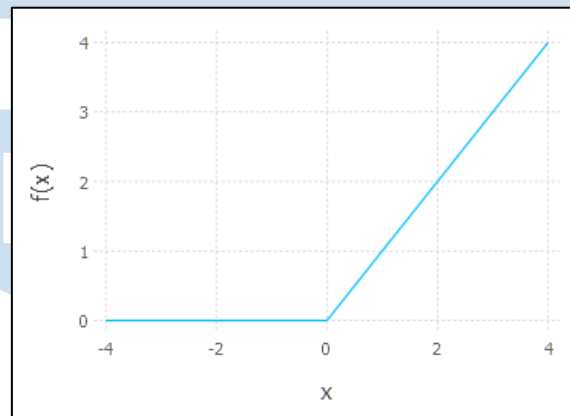
Gambar 2.4 Fungsi Aktivasi Non-Linear [30]



Berikut beberapa fungsi aktivasi non-linear:

a. ReLU (*Rectified Linear Unit*)

ReLU merupakan fungsi aktivasi yang paling banyak digunakan untuk memperkenalkan non-linearitas ke dalam *neural network*. ReLU mempunyai rentang nilai dari 0 sampai bilangan positif tak terhingga [31]. Fungsi ini efisien dan menjadikan proses konvergensi lebih cepat, serta dapat diterapkan pada *convolutional layer* dan *fully connected layer*. Berikut grafik fungsi aktivasi ReLU:



Gambar 2.5 Fungsi Aktivasi Non-linear ReLU [31]

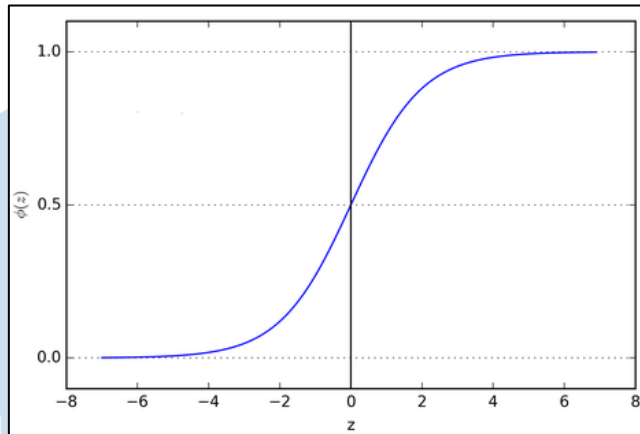
Rumus dari fungsi ReLU adalah sebagai berikut:

$$f(x) = a = \max(0, x) \quad (2.1)$$

$$f'(x) = \{1; \text{if } z > 0, 0; \text{if } z < 0, \text{undefined if } z = 0\} \quad (2.2)$$

b. *Sigmoid*

*Sigmoid* merupakan fungsi aktivasi yang digunakan untuk memprediksi probabilitas pada *output*. Rentang nilai *sigmoid* adalah 0 sampai 1, di mana jika *output* lebih besar atau sama dengan 0.5 akan menghasilkan nilai 1, dan jika lebih kecil dari 0.5 akan menghasilkan 0. Berikut grafik fungsi aktivasi *sigmoid*:



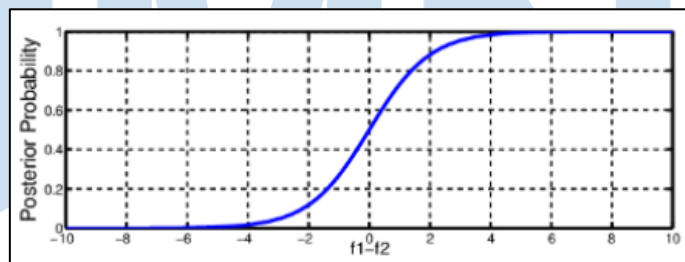
Gambar 2.6 Fungsi Aktivasi Non-linear *Sigmoid* [30]

Rumus dari fungsi *sigmoid* adalah sebagai berikut:

$$\phi(z) = \frac{1}{1 + e^{-z}} \quad (2.3)$$

c. *Softmax*

*Softmax* merupakan fungsi aktivasi yang dipakai untuk mendapatkan hasil klasifikasi ke beberapa kelas (lebih dari dua kelas) yang ditentukan dan dipakai pada *output layer* pada *fully connected layer*. *Softmax* dipakai untuk menghasilkan *output* berupa *multi-class* dan menghasilkan nilai berupa probabilitas untuk tiap kelasnya [29]. Berikut grafik fungsi aktivasi *softmax*:



Gambar 2.7 Fungsi Aktivasi Non-linear *Softmax* [31]

Rumus dari fungsi *softmax* adalah sebagai berikut:

$$f(x) = e^{x_i} / (\sum_{j=0} e^{x_j}) \quad (2.4)$$

$$S_j = P(y = j|x) \quad (2.5)$$

#### 2.5.4 Optimization

Untuk meningkatkan akurasi dan mengurangi kerugian (*loss*) pada model, perlu dilakukan *optimization*. *Optimizer* akan memperbarui parameter bobot untuk meminimalkan *loss function* dan meningkatkan akurasi. Tujuan dari *optimization* adalah untuk mencapai model terbaik terhadap serangkaian kriteria dengan memanfaatkan bobotnya dan mengurangi kesalahan antara yang *output* yang dikomputasi dan *output* yang diinginkan [32].

*Optimizer* yang banyak digunakan adalah *Adaptive Moment Estimation* (Adam). Adam menghitung *learning rate* di setiap *step* berdasarkan 2 vektor yaitu *order moments* pertama dan kedua (*mean* dan *variance*), yang didefinisikan secara berulang menggunakan gradien dan kuadrat gradien masing-masing [33]. *Order moments* pertama dan kedua didefinisikan sebagai berikut:

$$m_{t,i} = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.6)$$

$$v_{t,i} = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.7)$$

#### 2.5.5 Loss Function

Selama mengoptimasi model, setiap *error* pada model harus diestimasi berulang kali. *Loss function* dapat digunakan untuk memperkirakan kerugian (*loss*) pada model sehingga bobot dapat diperbarui terus-menerus untuk mengurangi kerugian (*loss*) pada evaluasi berikutnya [34].

Terdapat beberapa jenis dari *loss function* sebagai berikut:

##### 1. Binary cross-entropy loss

Fungsi ini dipakai pada masalah klasifikasi yang menggunakan *binary* atau target *value*-nya adalah 0 atau 1.

##### 2. Multi-class cross-entropy loss

Fungsi ini dipakai pada masalah klasifikasi *multi-class* atau di mana target *value*-nya adalah 0, 1, 2, ..., n, dan tiap kelasnya memiliki *value integer* yang unik.

## 2.6 Confusion Matrix

*Confusion matrix* adalah tabel yang digunakan untuk mendefinisikan kinerja dari algoritma klasifikasi, dengan memvisualisasikan dan merangkum kinerja algoritma klasifikasi yang dipakai [35]. *Confusion matrix* menghasilkan nilai aktual dan nilai prediksi setelah proses klasifikasi, di mana nilai-nilai tersebut disajikan dalam bentuk matriks [36].

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	<b>TP</b> (True Positive)	<b>FP</b> (False Positive) <i>Type I Error</i>
	0 (Negative)	<b>FN</b> (False Negative) <i>Type II Error</i>	<b>TN</b> (True Negative)

Gambar 2.8 *Confusion Matrix* [9]

Setelah *confusion matrix* dibuat, performa dari algoritma klasifikasi dapat diukur dengan parameter *accuracy*, *precision*, *recall*, dan *F1 score*.

### 1. Accuracy

*Accuracy* direpresentasikan sebagai rasio data yang diklasifikasikan dengan benar (TP+TN) dengan jumlah keseluruhan data (TP+TN+FP+FN) [35].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2.8)$$

### 2. Precision

*Precision* direpresentasikan sebagai rasio data yang diklasifikasikan dengan benar (TP) dengan jumlah keseluruhan data yang diklasifikasikan positif (TP+FP) [35].

$$Precision = \frac{TP}{TP+FP} \quad (2.9)$$

### 3. *Recall*

*Recall* didefinisikan sebagai rasio data yang terklasifikasi benar (TP) dibagi dengan jumlah keseluruhan data aktual yang positif [35]. *Recall* memperlihatkan seberapa berhasil data aktual positif terklasifikasikan positif.

$$Recall = \frac{TP}{TP+FN} \quad (2.10)$$

### 4. *F1 score*

*F1 score* juga dikenal sebagai *F Measure* menyatakan keseimbangan antara *precision* dan *recall* yang biasa dipakai jika kelas data tidak seimbang [35].

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2.11)$$

## 2.7 *Tools*

### 2.7.1 *Python*

Python merupakan salah satu bahasa pemrograman tingkat tinggi yang bersifat *open source*. Hingga saat ini Python digunakan secara luas dikarenakan memiliki beragam *library* standar, dapat dijalankan di berbagai *platform*, *developer-friendly* karena mudah dipahami, serta kapasitasnya untuk berintegrasi dengan bahasa pemrograman lain dengan tetap menggunakan fitur-fiturnya [37]. Python dapat digunakan dalam melakukan *scripting*, *web scraping*, maupun pembuatan *dataset*.

Adapun Python terkenal di komunitas ilmiah dan banyak digunakan pada komputasi numerik dan ilmiah karena banyaknya *library* yang kaya akan fungsi dan modul, mulai dari tugas mudah seperti data manipulasi hingga pembentukan model *deep learning*.

## 2.7.2 TensorFlow

TensorFlow adalah *open-source framework* yang dikembangkan oleh peneliti Google dalam menjalankan tugas *machine learning*, *deep learning*, serta analisis statistik dan deskriptif lainnya [38]. TensorFlow mampu memproses representasi abstrak model secara otomatis serta dilengkapi dengan alat pendukung yaitu TensorBoard untuk visualisasi mendalam tentang *progress* pelatihan model.

Salah satu API (*Application Programming Interface*) tingkat tinggi pada TensorFlow yaitu Keras, juga menunjang kinerja pemodelan dan pelatihan model dengan menyediakan cara sederhana untuk membangun *neural network* mulai dari *layer* yang standar [38].

## 2.8 Penelitian Terdahulu

Tabel 2.1 menunjukkan beberapa penelitian terdahulu mengenai klasifikasi teks dalam melakukan *Aspect-Based Sentiment Analysis* maupun sentimen analisis pada umumnya.

Tabel 2.1 Penelitian Terdahulu

No	Jurnal	Penulis	Algoritma	Hasil dan Simpulan
1	<i>Deep learning for aspect-based sentiment analysis on Indonesian hotels reviews, Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control Journal, Vol. 6, No. 3, 2021.</i>	Siwi Cahyaningtyas, Dhomas Hatta Fudholi, Ahmad Fathan Hidayatullah	RNN, LSTM, GRU, BiLSTM, CNN, CNN-LSTM, CNN-BiLSTM	Penelitian menggunakan data teks ulasan hotel dengan target klasifikasi aspek yaitu <i>price, hotel, room, location, service, dan restaurant</i> , serta target klasifikasi sentimen yaitu positif dan negatif. Pada klasifikasi aspek, model LSTM memiliki akurasi tertinggi yaitu 92.6%, sedangkan pada klasifikasi sentimen, model terbaik yaitu CNN dengan akurasi 90.4%. Adapun model CNN pada klasifikasi aspek juga mendapatkan akurasi yang cukup tinggi yaitu sebesar 91.8%.
2	<i>Aspect-Based Sentiment Analysis of Online Marketplace Reviews Using Convolutional Neural</i>	Mhd. Theo Ari Bangsa, Sigit Priyanta, Yohanes Suyanto	CNN, <i>Naïve Bayes</i>	Penelitian menggunakan data teks ulasan pada <i>online marketplace</i> dengan target klasifikasi aspek yaitu <i>accuracy, quality, service,</i>

No	Jurnal	Penulis	Algoritma	Hasil dan Simpulan
	<i>Network, IJCCS (Indonesian Journal of Computing and Cybernetics Systems), Vol.14, No.2, 2020.</i>			<i>price, packaging, dan delivery, serta target klasifikasi sentimen yaitu positif dan negatif. Model CNN pada klasifikasi aspek memiliki akurasi 91.98% dan pada klasifikasi sentimen memiliki akurasi 93.07%. Sewaktu dijalankan secara sequential, model CNN mendapat akurasi yang tinggi yaitu 85.54% dibanding Naïve Bayes 62.53%.</i>
3	<i>Aspect-Based Sentiment Analysis Approach with CNN, Proceeding of Electrical Engineering, Computer Science and Informatics (EECSI), 2018.</i>	Budi M Mulyo, Dwi H Widyantoro	SVM, Naïve Bayes, KNN, CNN	Penelitian menggunakan 3 jenis data ulasan yaitu ulasan restoran, laptop, dan hotel. Dari ketiga jenis data ulasan tersebut, klasifikasi aspek dan sentimen dengan model CNN mendapatkan rata-rata F1 yang paling tinggi yaitu 71%. Kekurangannya adalah data <i>training</i> yang cukup kecil.
4	<i>Aspect Based Sentimen Analysis Opini Publik Pada Instagram dengan Convolutional Neural Network, Journal of Intelligent Systems and Computation, Vol.1, No.2, 2019.</i>	Muhammad Arief Rahman, Herman Budianto	CNN	Penelitian menggunakan data teks ulasan operator telekomunikasi dengan target klasifikasi aspek yaitu layanan, kecepatan, <i>coverage</i> , dan promosi, serta target klasifikasi sentimen yaitu positif dan negatif. Hasil evaluasi model CNN menggunakan F1 <i>score</i> dimana klasifikasi aspek sebesar 87.6% dan klasifikasi sentimen sebesar 95.6%, dengan rata-rata yaitu 91.62%.
5	<i>Aspect-Based Sentiment Analysis on Indonesian Restaurant Review Using a Combination of Convolutional Neural Network and Contextualized Word Embedding, IJCCS (Indonesian Journal of Computing and Cybernetics Systems), Vol.15, No.3, 2021.</i>	Putri Rizki Amalia, Edi Winarko	CNN	Penelitian menggunakan data teks ulasan restoran dengan target klasifikasi aspek yaitu <i>food, price, place, dan service</i> , serta target klasifikasi sentimen yaitu positif dan negatif. Penelitian menggunakan tiga jenis model CNN yaitu BERT-CNN, ELMo-CNN, dan Word2vec-CNN. Klasifikasi aspek dengan ELMo-CNN menghasilkan micro-avg f1 <i>score</i> tertinggi yaitu 86%, sedangkan klasifikasi sentimen dengan BERT-CNN sebesar 91%.

No	Jurnal	Penulis	Algoritma	Hasil dan Simpulan
6	<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, 2020 Fifth International Conference on Informatics and Computing (ICIC), 2020.</i>	Ali Mustopa, Hermanto, Anna, Eri Bayu Pratama, Ade Hendini, Deni Risdiansyah	Naive Bayes, SVM	Penelitian menggunakan data teks ulasan aplikasi PeduliLindungi pada Google Play Store sebanyak 1,364 data untuk mengklasifikasikan sentimen positif dan negatif ulasan. Model Naive Bayes memiliki tingkat akurasi sebesar 69%, sedangkan model SVM memiliki tingkat akurasi sebesar 93%.
7	Apakah Youtuber Indonesia Kena Bully Netizen?, ULTIMA InfoSys, Vol. XI, No. 2, 2020	Joviano Siahaan, Wella, Ririn I. Desanti	SVM	Penelitian menggunakan data teks komentar Instagram YouTuber populer, dan model algoritma SVM untuk mengklasifikasikan komentar ke kelas positif (tidak mengandung <i>cyberbullying</i> ) dan kelas negatif (mengandung <i>cyberbullying</i> ). Model SVM mencapai hasil akurasi sebesar 81.2% dalam mengklasifikasikan sentimen komentar.

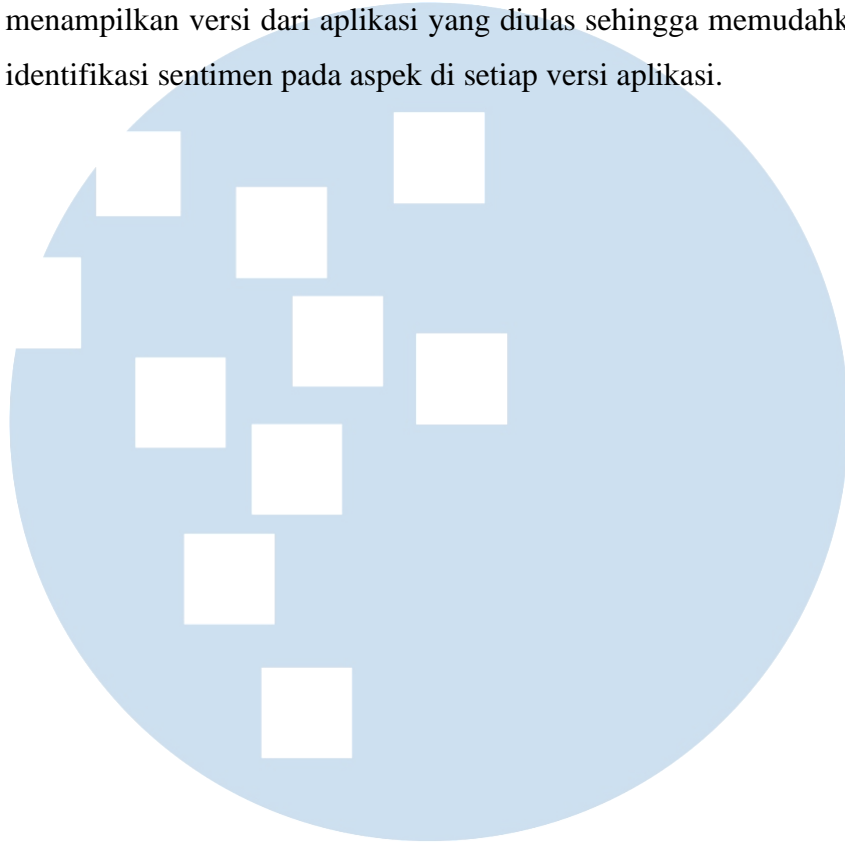
Dari tujuh penelitian terdahulu pada Tabel 2.1, dapat disimpulkan bahwa model dengan algoritma CNN terbukti memberikan hasil yang baik dalam melakukan klasifikasi aspek dan sentimen pada tugas *Aspect-Based Sentiment Analysis* (ABSA). Penelitian ini akan berfokus dengan penggunaan algoritma CNN, dan beberapa algoritma lain sebagai pembandingan yaitu SVM dan *Naive Bayes*.

Berikut beberapa perbandingan penelitian terdahulu dengan penelitian ini:

1. Menggunakan ABSA dalam menganalisis ulasan aplikasi PeduliLindungi, di mana penelitian ke-enam hanya sebatas analisis sentimen biasa.
2. Ketujuh penelitian terdahulu hanya sampai pada tahap pemodelan klasifikasi dan evaluasi *dataset*, sedangkan penelitian ini juga akan melakukan klasifikasi pada data yang tidak berlabel menggunakan model CNN yang sudah dibangun sehingga dapat dilakukan komparasi dan analisis pada hasil klasifikasinya.



3. Sumber data yang digunakan berasal dari Google Play Store yang juga menampilkan versi dari aplikasi yang diulas sehingga memudahkan dalam identifikasi sentimen pada aspek di setiap versi aplikasi.



# UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA