

BAB II

KAJIAN LITERATUR

2.1. Pemutihan Pajak

Pemutihan pajak kendaraan bermotor merupakan program pemerintah daerah berupa penghapusan sanksi administrasi atau denda keterlambatan pembayaran Pajak Kendaraan Bermotor (PKB). Dalam program ini, masyarakat umumnya hanya diwajibkan membayar pokok pajak tanpa dikenakan denda keterlambatan. Kebijakan ini bertujuan meningkatkan kepatuhan wajib pajak, mengurangi tunggakan pajak kendaraan, serta meningkatkan pendapatan asli daerah (PAD) [10].

Pajak Kendaraan Bermotor (PKB) sendiri merupakan pajak atas kepemilikan dan/atau penguasaan kendaraan bermotor yang dipungut oleh pemerintah provinsi. Pengaturan mengenai PKB tercantum dalam Undang-Undang Nomor 1 Tahun 2022 tentang Hubungan Keuangan antara Pemerintah Pusat dan Pemerintah Daerah (UU HKPD), yang menggantikan sebagian ketentuan sebelumnya pada Undang-Undang Nomor 28 Tahun 2009 tentang Pajak Daerah dan Retribusi Daerah [10], [11].

Selain mengatur PKB, UU HKPD juga memperkenalkan kebijakan opsen Pajak Kendaraan Bermotor yang mulai berlaku sejak 5 Januari 2025 [12], [13]. Opsen PKB merupakan pungutan tambahan pajak oleh pemerintah kabupaten/kota atas pokok PKB dengan tarif sebesar 66% dari pajak terutang, dengan tujuan memperkuat pendapatan daerah dan pemerataan fiskal tanpa menambah beban bagi wajib pajak [12], [14].

Pelaksanaan program pemutihan pajak kendaraan bermotor di Indonesia tidak bersifat nasional, melainkan ditentukan oleh masing-masing pemerintah provinsi melalui kebijakan daerah. Oleh karena itu, bentuk keringanan, periode pelaksanaan, maupun jenis denda yang dihapus dapat berbeda di setiap daerah [10].

2.2. Analisis Sentimen

Analisis sentimen adalah bidang ilmu yang mempelajari cara untuk menganalisis opini, pendapat, dan penilaian suatu pihak tentang objek tertentu yang dapat berupa produk, layanan, atau suatu masalah. Analisis sentimen dapat diimplementasikan pada berbagai tingkat yaitu teks yang berupa dokumen atau kalimat [15]. Analisis sentimen adalah teknik *Natural Language Processing* (NLP) dan *Machine Learning* (ML) untuk mengevaluasi secara otomatis teks berdasarkan perasaan yang dirasakan penulis. Ada tiga jenis utama analisis sentimen, yaitu [16]:

1. *Multimodal sentiment analysis*

Multimodal sentiment analysis adalah jenis analisis sentimen yang mempertimbangkan data seperti teks, video, dan audio untuk menganalisis emosi yang diekspresikan dalam data tersebut. Tanda visual dan pendengaran seperti ekspresi wajah dan nada suara dipertimbangkan yang memberikan spektrum sentimen yang luas.

2. *Aspect-based sentiment analysis*

Dalam analisis berbasis aspek, metode NLP digunakan untuk menganalisis dan mengekstraksi emosi dan pendapat yang terkait dengan aspek atau fitur tertentu dari produk dan layanan. Misalnya, peneliti dapat mengekstraksi sentimen terkait makanan, layanan, suasana, dan lainnya dalam ulasan restoran.

3. *Multilingual sentiment analysis*

Setiap bahasa memiliki tata bahasa, sintaksis, dan kosa kata yang berbeda, dan sentimen diekspresikan secara berbeda di setiap bahasa. Oleh karena itu, *Multilingual sentiment analysis* menggunakan kemampuan setiap bahasa secara khusus untuk mengekstrak sentimen dari teks yang dipelajari.

Tugas utama analisis sentimen adalah mengelompokkan teks yang terdapat dalam sebuah kalimat atau dokumen dan menentukan apakah pendapat yang diungkapkan dalam kalimat atau dokumen tersebut bersifat positif, negatif atau netral. Informasi mengenai sentimen pengguna dapat dimanfaatkan untuk membuat strategi bisnis yang lebih baik, mengalokasikan sumber daya dengan lebih optimal, serta merencanakan langkah-langkah pengembangan produk berdasarkan preferensi dan harapan pengguna [6]. Analisis sentimen dapat dikategorikan menjadi tiga kelas sentimen, yaitu [17]:

1. Sentimen Positif

Sentimen Positif didefinisikan dalam Kamus Besar Bahasa Indonesia (KBBI) sebagai tindakan atau sikap yang meningkatkan nilai seseorang atau sesuatu.

2. Sentimen Negatif

Sentimen negatif menurut Kamus Besar Bahasa Indonesia (KBBI) adalah sikap atau respon yang menurunkan nilai seseorang atau sesuatu. Kalimat dengan sentimen negatif mengubah perspektif tentang sesuatu yang menyebabkan tren turun.

3. Sentimen Netral

Menurut Kamus Besar Bahasa Indonesia (KBBI), kata "netral" berarti "tidak berpihak" dan kalimat bersentimen netral adalah kalimat yang tidak memiliki sifat positif atau negatif.

Sebelumnya, sejumlah studi telah dilakukan terkait analisis sentimen publik di media sosial. Analisis sentimen di beberapa media sosial termasuk X terhadap kenaikan Pajak Pertambahan Nilai (PPN) 12% menggunakan IndoBERT memperoleh nilai akurasi sebesar 84,94% [17], serta penelitian lain yang memanfaatkan IndoBERT untuk *word embedding* yang bersifat kontekstual dan mengekstraksi fitur secara efektif [5] Jadi, untuk penelitian ini akan digunakan model IndoBERT agar bisa memahami konteks kalimat dari teks data dan mengekstraksi fitur secara efisien sehingga menghasilkan nilai akurasi yang tinggi.

2.3. *Natural Language Processing (NLP)*

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan (*Artificial Intelligence*) yang berfokus pada kemampuan komputer untuk memahami, memproses, dan menghasilkan bahasa manusia secara alami [18]. NLP menggabungkan konsep linguistik komputasional, statistika, dan pembelajaran mesin untuk mengubah teks menjadi bentuk representasi numerik yang dapat dipahami oleh sistem [19]. Dalam konteks deteksi komentar spam, NLP berperan penting untuk memahami struktur dan makna kalimat dalam komentar pengguna, sehingga sistem mampu membedakan antara komentar yang relevan dengan konten dan komentar yang bersifat promosi atau mengandung unsur spam [20].

Tahapan umum dalam NLP mencakup proses *text preprocessing* yang meliputi *case folding*, *tokenization*, *stopword removal*, dan *stemming*. Langkah-langkah ini berfungsi untuk menyederhanakan teks agar dapat dianalisis oleh model secara lebih efisien [19], [21] Sebagai contoh, tahap *tokenization* digunakan untuk memecah kalimat menjadi unit-unit kata, sedangkan *stopword removal* menghapus kata umum seperti "yang", "dan", atau "di" yang tidak memiliki makna kontekstual signifikan. Proses *stemming* juga sangat penting untuk bahasa Indonesia karena memiliki variasi imbuhan seperti "bermain", "memainkan", dan "dimainkan" yang semuanya memiliki akar kata yang sama, yaitu "main" [18].

Dalam pengembangan sistem deteksi komentar spam, hasil dari pra-pemrosesan teks kemudian diubah menjadi representasi numerik melalui berbagai pendekatan seperti *Bag-of-Words (BoW)*, *Term Frequency-Inverse Document Frequency (TF-IDF)*, dan *word embeddings*. Pendekatan BoW menghitung frekuensi kemunculan kata dalam teks, sedangkan TF-IDF memberikan bobot lebih tinggi pada kata yang jarang muncul namun penting dalam konteks tertentu [22]. Sementara itu, representasi berbasis *word embeddings* seperti *Word2Vec* atau *GloVe* menghasilkan vektor yang menggambarkan makna semantik

antar kata sehingga model dapat mempelajari hubungan makna dalam teks secara lebih mendalam [19].

Klasifikasi teks merupakan salah satu aplikasi utama NLP yang berfungsi untuk mengelompokkan dokumen atau kalimat ke dalam kategori tertentu, seperti spam dan non-spam. Pendekatan tradisional menggunakan statistical text classification, di mana teks direpresentasikan dalam bentuk fitur numerik dan diproses menggunakan algoritma pembelajaran mesin seperti Naïve Bayes atau Support Vector Machine (SVM) [20]). Namun, metode ini memiliki keterbatasan dalam memahami konteks semantik yang kompleks, terutama pada teks dengan gaya penulisan non-formal.

2.4. Machine Learning

Machine learning adalah kumpulan teknik yang bermanfaat untuk menangani dan memprediksi data yang berskala besar dengan cara menggunakan algoritma pembelajaran yang diterapkan pada data tersebut [19]. Machine learning juga dapat didefinisikan sebagai penggunaan komputer dan algoritma matematika untuk membuat prediksi di masa depan dengan belajar dari data yang sudah ada [20]. Machine learning terbagi menjadi empat jenis, yaitu:

1. Supervised Learning

Teknik supervised learning digunakan untuk mengidentifikasi hubungan di antara atribut input dan atribut target dimana klasifikasi adalah salah satu kategori untuk memprediksi keanggotaan grup di dalam contoh data, klasifikasi merupakan teknik data mining. Logistic Regression, Support Vector Machine, Random Forest Classifier, Naive Bayes Bernoulli adalah algoritma dari supervised learning [21].

1. Unsupervised Learning

Unsupervised learning sangat baik untuk mengelola atau mengklasifikasi suatu pola dari banyaknya objek yang sejenis dimana objeknya tidak sepenuhnya sama. Tujuan dari unsupervised learning adalah untuk membuat penggunaanya dapat mengelompokkan objek objek yang memiliki nilai yang sama dalam ruang lingkup tertentu dimana clustering adalah proses membagi data ke dalam sejumlah kelompok tertentu [22]. Algoritma unsupervised learning adalah K-Means, Hierarchical Clustering, DBSCAN dan sebagainya.

2. Semi-supervised Learning

Semi-supervised learning adalah jenis machine learning dimana menggabungkan antara data yang mempunyai label dan data yang tidak memiliki label dan melatih model tersebut untuk melakukan tugas regresi dan klasifikasi. Dimana ini relevan di dalam situasi

dimana akan mendapatkan sejumlah besar data yang berlabel cukup sulit atau mahal, tetapi untuk mendapatkan sejumlah data yang berlabel lebih mudah [23].

3. Reinforcement Learning

Reinforcement learning mengumpulkan pengetahuan untuk memilih tindakan yang akan menghasilkan hasil terbaik yang diinginkan dimana dia memiliki banyak keunggulan ketika dibandingkan dengan jenis lainnya dimana keunggulannya untuk menjelajahi sendiri lingkungan yang sangat dinamis dan stokastik dan membuat kebijakan kontrol yang ideal dengan menggunakan umpan balik evaluasi dari lingkungan tersebut [24].

Berdasarkan pada pendapat Peter Harington yang menjelaskan beberapa alur kinerja machine learning, yaitu [25]:

1. Mengumpulkan data, seperti file Excel, Microsoft Access, file teks dan lainnya.
2. Mempersiapkan data dengan cara menentukan kualitas data dan setelahnya mengambil langkah-langkah untuk memperbaiki masalahnya contohnya seperti kehilangan data.
3. Melatih sebuah model dengan membagi data yang sudah disiapkan menjadi dua bagian, yaitu data training yang digunakan dalam pengembangan model dan juga data testing digunakan sebagai referensi.
4. Mengevaluasi model dengan cara menentukan pilihan algoritma berdasarkan pada hasil dari pengujian.
5. Meningkatkan kinerja dengan memilih model yang lain atau dengan menambahkan lebih banyak variabel dalam hal untuk meningkatkan efisiensi.

2.5. Transformer dan BERT

Model *Transformer* merupakan inovasi besar dalam bidang *Natural Language Processing* (NLP) karena mampu memproses konteks kalimat secara paralel dan efisien [25]. Tidak seperti model berbasis RNN atau LSTM yang membaca teks secara berurutan, *Transformer* menggunakan mekanisme *self-attention*, yang memungkinkan model memahami hubungan antara semua kata dalam kalimat sekaligus.

Secara umum, arsitektur *Transformer* terdiri atas dua komponen utama, yaitu *encoder* dan *decoder*. Bagian *encoder* berfungsi untuk memproses dan memahami konteks dari input teks, sedangkan *decoder* digunakan untuk menghasilkan output berupa teks atau prediksi berdasarkan representasi yang telah dibentuk oleh *encoder* [27]. Dalam setiap lapisan *encoder* terdapat tiga komponen penting: *self-attention layer*, *feed-forward neural network*, dan *normalization layer*. *Self-attention* memungkinkan model menentukan kata

mana yang relevan dengan kata lain dalam kalimat, sementara *feed-forward layer* memperkaya representasi *semantic*. *Decoder* memiliki struktur serupa tetapi menambahkan *masked self-attention*, agar prediksi setiap kata hanya mempertimbangkan kata sebelumnya, bukan kata yang akan datang [28].

Berdasarkan arsitektur *Transformer*, model BERT (*Bidirectional Encoder Representations from Transformers*) dikembangkan oleh *Google Research* untuk meningkatkan pemahaman konteks kalimat dari dua arah depan ke belakang dan sebaliknya [29]. Dengan pendekatan *bidirectional*, BERT dapat memahami makna kata berdasarkan posisi dan konteks sekitarnya, bukan hanya berdasarkan urutan [13].

Berbeda dari *Transformer* asli yang memiliki dua komponen (*encoder-decoder*). BERT hanya menggunakan bagian *encoder* dari *Transformer*, karena tujuannya bukan untuk menghasilkan teks baru seperti pada penerjemahan otomatis, melainkan untuk memahami dan merepresentasikan makna teks secara mendalam. *Encoder* dalam BERT terdiri atas beberapa lapisan *Transformer* (biasanya 12 atau 24), di mana setiap lapisan belajar memahami hubungan kontekstual antar kata melalui mekanisme *multi-head self-attention* [23].

2.6. IndoBERT

IndoBERT atau BERT dengan versi Indonesia adalah sebuah model *Deep Learning* untuk *Natural Language Processing* (NLP) yang dibangun dari model *Transformer* dimana setiap elemen output terhubung ke setiap elemen input. IndoBERT adalah *pre-trained model* yang dilatih dengan dataset yang besar yang berasal dari berbagai sumber [41]. BERT (*Bidirectional Encoder Representations from Transformers*), sebagai model bahasa berbasis *Transformer* yang memanfaatkan teknik pembelajaran *bidirectional* dan *pretraining* pada data tekstual yang besar, telah membuktikan keunggulannya dalam memahami konteks semantik yang kompleks melalui mekanisme *self-attention* dan representasi yang kontekstual, sehingga pengembangan IndoBERT yang menggunakan arsitektur serupa namun dilatih khusus dengan data berbahasa Indonesia menjadi pilihan yang sangat relevan untuk berbagai tugas dalam konteks bahasa Indonesia [42]. Mekanisme *self-attention* digunakan untuk menghitung seberapa besar perhatian yang diberikan satu *token* terhadap *token* lainnya dalam kalimat. Proses *self-attention* ini dilakukan melalui langkah-langkah berikut:

1. *Query*, *Key*, dan *Value*: Setiap *token* akan memiliki representasi dalam bentuk vektor *query*, *key*, dan *value*, yang dihasilkan dari vektor *embedding* mereka.

2. Perhitungan *Attention*: Misalnya, untuk *token* "rakyat", perhatian terhadap *token* "menuai" dan "kritik" dihitung dengan cara mengalikan vektor *query* Q untuk "rakyat" dengan vektor *key* K untuk "menuai" dan "kritik" dan membagi hasil perkalian tersebut dengan $\sqrt{d_k}$, dimana d_k adalah dimesi dari vector key (768). Hasil perkalian ini diproses untuk menghasilkan bobot perhatian yang menunjukkan seberapa relevan satu token terhadap lainnya, Rumusnya adalah sebagai berikut.

$$Attention(Q, K, V) = softmax = \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (1)$$

3. *Softmax*: Hasil dari perkalian Q-KT akan digunakan dalam operasi *softmax* untuk menghasilkan bobot perhatian yang nilainya berada di antara 0 dan 1. Bobot ini menentukan seberapa besar kontribusi masing-masing *token* terhadap *token* lainnya.
4. *Weighted Sum*: Vektor *value* V akan dipengaruhi oleh bobot perhatian yang dihasilkan, dan akan dihitung sebagai *weighted sum* untuk menghasilkan representasi *token* yang baru.

IndoBERT dilatih dengan menggunakan lebih dari 200 juta kata, sumber utamanya adalah Wikipedia Indonesia (74 juta kata), artikel berita dari Kompas, Tempo, dan Liputan6 (55 juta kata), dan *Indonesian Web Corpus* (90 juta kata) [43]. Beberapa tahapan yang harus dilakukan untuk persiapan dan pembersihan data teks agar dapat digunakan dalam model *machine learning* yaitu, *Preprocessing*, *Tokenizing*, *Stemming*, dan *Stopwords* [44]. Model IndoBERT lebih mampu membedakan antara pandangan positif atau negatif pada data *tweet* X berbahasa Indonesia dibandingkan dengan model analisis sentimen terdahulu lainnya [45].

IndoBERT digunakan juga sebagai *word embedding* kontekstual yang memanfaatkan model *pre-trained*. *Word embedding* merupakan proses mengonversi kata menjadi vektor. Metode terdahulu untuk menghasilkan embedding kata mencakup pendekatan *bag of words* dan *term frequency-inverse document frequency* (TF-IDF). Pada proses *embedding*, langkah pertama yang dilakukan adalah menambahkan *token* khusus di awal dan akhir kalimat, yaitu *token* [CLS] dan [SEP]. *Token* khusus ini berfungsi untuk memisahkan bagian awal (*prefix*) dan akhir (*suffix*) dari kalimat. Selanjutnya, dilakukan proses tokenisasi yang mengubah kalimat menjadi kata atau *token*. Proses tokenisasi ini menggunakan metode *word piece tokenization*, yang memanfaatkan kosakata yang dimiliki oleh model serta kosakata di luar

model [46]. Setelah menghasilkan *embedding* dari semua *token*, kita melakukan *pooling* untuk mengubah sekumpulan *vector* menjadi satu representasi tunggal. Ada beberapa jenis *pooling* yang sering digunakan dalam *embedding* model BERT, antara lain:

1. *CLS Pooling*, menggunakan *embedding* dari *token* [CLS], yaitu *token* khusus yang ditambahkan di awal kalimat. Model BERT dirancang sedemikian rupa sehingga *embedding* [CLS] di akhir pemrosesan akan mengandung informasi dari seluruh kalimat.
2. *Mean Pooling*, *Mean Pooling* menggabungkan semua *embedding token* dalam kalimat dengan menghitung rata-ratanya. Dengan demikian, setiap *embedding token* memberikan kontribusi yang merata terhadap representasi akhir kalimat. Rumus *Mean Pooling*:

$$E_{mean} = \frac{1}{n} \sum_{i=1}^n e_i \quad (2)$$

Dimana,

e_i adalah *embedding* dari *token* ke- i , dan n adalah jumlah total *token* dalam teks.

2.7. Convolution Neural Network

CNN pada awalnya dirancang untuk pemrosesan data dua dimensi seperti citra, bekerja melalui mekanisme *weight sharing* dalam mengekstraksi fitur hierarkis dari data berstruktur *grid* [24]. Namun, dalam konteks data teks maupun deret waktu, CNN satu dimensi (Conv1D) digunakan untuk memindai fitur lokal sepanjang sumbu urutan. Conv1D terbukti lebih efektif dalam mengekstraksi fitur lokal pada teks pendek dibandingkan model sekuensial seperti GRU, karena kemampuannya mendeteksi pola n -gram pada posisi mana pun dalam kalimat secara *translation-invariant* [25], [26]. Pendekatan ini juga telah dimanfaatkan secara luas dalam model *hybrid* CNN-LSTM untuk analisis teks berbahasa Indonesia, di mana CNN berperan mengekstraksi fitur teks sebelum diteruskan ke LSTM untuk memodelkan dependensi sekuensialnya. Operasi konvolusi diskrit satu dimensi yang diterapkan pada sinyal input x dengan filter (kernel) w didefinisikan sebagai berikut:

$$(x * w)[t] = \sum_k x[t + k] \cdot w[k] \quad (3)$$

Dimana,

t adalah posisi waktu,

k adalah indeks kernel,

$*$ menandakan operasi konvolusi.

Kernel bergeser sepanjang dimensi waktu dengan stride tertentu, menghasilkan representasi fitur yang disebut feature map.

Setelah operasi konvolusi, fungsi aktivasi diterapkan secara elemen. Fungsi aktivasi yang paling umum digunakan dalam lapisan konvolusi adalah *Rectified Linear Unit* (ReLU), yang didefinisikan sebagai:

$$f(z) = \max(0, z) \quad (4)$$

Dimana,

z adalah nilai input,

Jika $z > 0$, maka nilai output = z ,

Jika $z \leq 0$, maka nilai output = 0,

ReLU mempercepat konvergensi pelatihan dan mengurangi masalah vanishing gradient pada lapisan konvolusi. Beberapa varian seperti *Leaky ReLU* dan *ELU* juga digunakan dalam kasus tertentu.

Dengan F filter yang masing-masing berukuran K , output dari lapisan Conv1D pada posisi t untuk filter ke- f dapat dituliskan sebagai:

$$h_f[t] = \text{ReLU}(\sum_{k=0}^{K-1} x[t \cdot s + k] \cdot w_f[k] + b_f) \quad (5)$$

Dimana,

s adalah *stride*,

w_f adalah bobot filter ke- f ,

b_f adalah bias filter ke- f .

Output ini membentuk *feature map* berdimensi (T', F) dengan T' bergantung pada panjang input, ukuran *kernel*, *stride*, dan *padding*.

2.8. Operasi Pooling

Lapisan *pooling* mengikuti lapisan konvolusi untuk mengurangi dimensi spasial *feature map* sekaligus memberikan invariansi translasi parsial. Dua jenis *pooling* yang paling umum adalah *max pooling* dan *average pooling*. *Max pooling* mengambil nilai maksimum dalam jendela *pooling* berukuran p :

$$\text{MaxPool}(h)[t] = \max\{h[t \cdot p], h[t \cdot p + 1], \dots, h[t \cdot p + p - 1]\} \quad (6)$$

Sementara, *average pooling* menghitung rata-rata:

$$\text{AvgPool}(h)[t] = \left(\frac{1}{p}\right) \cdot \sum_{i=0}^{p-1} h[t \cdot p + i] \quad (7)$$

Max pooling lebih umum digunakan dalam konteks deret waktu karena lebih efektif dalam mempertahankan fitur dominan (seperti puncak atau titik perubahan tren) yang relevan untuk pemodelan temporal.

2.9. Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) diperkenalkan oleh Hochreiter & Schmidhuber pada tahun 1997 [27], [28] sebagai solusi atas masalah *vanishing gradient* pada RNN konvensional. Arsitektur LSTM memperkenalkan konsep *cell state* (C_t) sebagai "memori jangka panjang" yang mengalir sepanjang sekuens dengan modifikasi minimal, dikontrol oleh tiga mekanisme gerbang: *input gate*, *forget gate*, dan *output gate* [27]. Setiap gerbang menggunakan fungsi sigmoid untuk menghasilkan nilai antara 0 (tutup penuh) dan 1 (buka penuh), sehingga secara selektif mengatur aliran informasi yang masuk, dipertahankan, maupun dikeluarkan dari *cell state* [28], [29].

2.9.1. Forget Gate

Forget gate menentukan seberapa banyak informasi dari *cell state* sebelumnya yang akan dipertahankan atau dilupakan:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (8)$$

Dimana,

σ adalah fungsi sigmoid,

W_f adalah matriks bobot *forget gate*,

h_{t-1} adalah *hidden state* dari timestep sebelumnya,

x_t adalah input pada *timestep* t ,

b_f adalah bias.

Jika nilai f_t mendekati 0 berarti informasi lama dilupakan; sebaliknya apabila mendekati 1 berarti dipertahankan.

2.9.2. Input Gate

Input gate menentukan informasi baru apa yang akan ditambahkan ke *cell state*, sementara kandidat *cell state* menghasilkan nilai baru yang mungkin disimpan:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (9)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (10)$$

Dimana,

i_t adalah *gate* yang mengontrol seberapa banyak informasi baru dimasukkan, $\tilde{C}_t$ adalah kandidat nilai baru yang dihitung menggunakan fungsi *tanh* sehingga nilainya berada dalam rentang $[-1, 1]$.

2.9.3. Cell State

Cell state diperbarui dengan menggabungkan *forget gate* (melupakan informasi lama) dan input *gate* (menambahkan informasi baru):

$$C_t = f_t \odot C_{\{t-1\}} + i_t \odot \tilde{C}_t \quad (11)$$

Simbol $\odot$ menandakan perkalian elemen per elemen (*Hadamard product*). Persamaan ini adalah keunggulan LSTM, *cell state* dapat mempertahankan informasi jangka panjang karena gradien mengalir melalui operasi penjumlahan tanpa terdegradasi.

2.9.4. Output Gate & Hidden Gate

Output gate menentukan bagian mana dari *cell state* yang akan menjadi *output* (*hidden state*):

$$o_t = \sigma(W_t \cdot [h_{\{t-1\}}, x_t] + b_o) \quad (12)$$

$$h_t = o_t \odot \tanh(C_t) \quad (13)$$

Dimana,

h_t adalah *hidden state* pada timestep t yang berfungsi sebagai representasi sekuens hingga titik tersebut. Nilai ini diteruskan ke timestep berikutnya dan juga menjadi output dari sel LSTM pada timestep t .

2.10. Model Hibrida

Model hibrida (*hybrid model*) merupakan pendekatan yang menggabungkan dua atau lebih metode, algoritma, atau arsitektur pembelajaran untuk memanfaatkan keunggulan masing-masing metode sekaligus mengurangi kelemahannya [23]. Konsep ini berkembang karena tidak ada satu algoritma yang mampu memberikan performa optimal untuk seluruh jenis permasalahan — sejalan dengan prinsip *No Free Lunch Theorem* dalam pembelajaran mesin [23]. Oleh karena itu, integrasi beberapa teknik sering menghasilkan akurasi, generalisasi, dan efisiensi yang lebih baik dibandingkan model tunggal [30].

Model hibrida menggabungkan teknik optimasi dan *machine learning* untuk meningkatkan kemampuan pencarian solusi, estimasi parameter, serta menghindari *local*

optimum. Pendekatan ini banyak diterapkan pada klasifikasi, *clustering*, dan prediksi karena mampu mengatasi keterbatasan algoritma tunggal [31].

Dalam bidang *Deep Learning*, model hibrida dapat dikategorikan menjadi tiga jenis utama, yaitu [30], [32]:

1. *Spatial Hybrid Model*

- a. Menggabungkan beberapa arsitektur untuk mengekstraksi fitur spasial.
- b. Contoh: CNN-VGG, CNN-ResNet.
- c. Banyak digunakan pada klasifikasi citra dan deteksi objek.

2. *Temporal Hybrid Model*

- a. Menggabungkan model yang mampu menangani data berurutan atau *time series*.
- b. Contoh: CNN-LSTM, CNN-GRU.
- c. Cocok untuk pengenalan aktivitas, video, dan analisis sentimen.

3. *Spatial-Temporal Hybrid Model*

- a. Mengombinasikan ekstraksi fitur spasial dan temporal secara bersamaan.
- b. Contoh: CNN-LSTM-Attention.
- c. Digunakan pada *video analysis*, *gesture recognition*, dan sistem cerdas lainnya.
- d. Pada bidang *Natural Language Processing* (NLP), model hibrida semakin populer karena mampu menggabungkan kemampuan representasi konteks dari *Transformer* dengan kemampuan pemodelan urutan dari RNN atau LSTM. Beberapa penelitian menunjukkan bahwa kombinasi model seperti BERT-LSTM, BERT-CNN, maupun CNN-LSTM menghasilkan performa yang lebih baik pada tugas analisis sentimen, klasifikasi teks, dan *Named Entity Recognition* dibandingkan model tunggal.

2.11. *Confusion Matrix*

Tahapan evaluasi merupakan proses menganalisis program untuk memberikan penilaian dari hasil keputusan program agar dapat meningkatkan keefektifan dari program tersebut [33]. Langkah ini merupakan kunci dalam NLP karena mengukur kinerja model yang telah dibangun dengan menggunakan metrik evaluasi yang tepat. Metrik evaluasi dapat bervariasi tergantung pada tugas atau masalah yang dihadapi dan fase pengembangan model [34]. Salah satu teknik pengujian untuk klasifikasi kelas pada analisis sentimen dapat menggunakan *confusion matrix*.

Pada *confusion matrix*, *True Positive* (TP) menunjukkan jumlah kalimat yang diklasifikasikan dengan benar sebagai sentimen positif oleh model *machine learning*, *False Positive* (FP) menandakan jumlah kalimat yang secara keliru dikategorikan sebagai sentimen

negatif oleh model *machine learning*, *True Negative* (TN) mewakili jumlah kalimat yang secara akurat dimasukkan ke dalam kategori sentimen negatif yang sebenarnya, dan *False Negative* (FN) menunjukkan jumlah kalimat yang secara tidak akurat ditempatkan ke dalam kategori sentimen positif [33], [35]. Untuk menentukan hasil model klasifikasi menggunakan *confusion matrix* dapat diukur menggunakan [33], [35]:

1. *Accuracy* menunjukkan proporsi prediksi benar dari seluruh data, yang dapat di hitung menggunakan persamaan ini:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (14)$$

2. *Precision* menggambarkan seberapa tepat model dalam mengidentifikasi apakah model sudah benar memberi nilai sentimen yang tepat, yang dapat di hitung menggunakan persamaan ini:

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

3. *Recall* mengukur kemampuan model menemukan seluruh data yang tepat termasuk dalam suatu kelas, yang di hitung menggunakan persamaan ini:

$$Recall = \frac{TP}{TP+FN} \quad (16)$$

4. F1-Score merupakan harmonisasi antara *precision* dan *recall*, digunakan ketika *dataset* tidak seimbang, yang dapat di hitung menggunakan persamaan ini:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (17)$$