# BAB II
# TINJAUAN PUSTAKA

## 2.1 Penelitian Terkait

Berikut adalah penelitian terkait yang menjadi pertimbangan dan pembanding penulis melakukan penelitian ini :

**Tabel 1 Penelitian Terkait**

| Peneliti | Judul Penelitian | Tahun | Metode dan Hasil Penelitian | Kelebihan | Kekurangan |
|---|---|---|---|---|---|
| Mohammad Rezza Fahlevvi | Analisis Sentimen Terhadap Ulasan Aplikasi Pejabat Pengelola Informasi dan Dokumentasi Kementerian Dalam Negeri Republik Indonesia di Google Playstore Menggunakan Metode Support Vector Machine [6]. | 2022 | Metode pengumpulan data dengan menggunakan *Web Scrapping* dengan pada komentar *Google Play Store* di aplikasi PPID Kemendagri. Metode analisis text dengan SVM. Hasil Penelitiannya data ulasan yang berjumlah 700 data dengan label 85 positif dan 615 negatif. Dan hasil analisis menggunakan SVM menghasilkan rata-rata k-fold sebesar 88%, precision 94%, recall 100%, f-measure 97%, dan accuracy 97% | Melakukan klasifikasi data sentiment ulasan dengan menggunakan SVM yang termasuk dalam kategori *supervised learning*. | Data ulasan yang di gunakan masih tergolong sedikit yaitu berjumlah 700 data. Tidak di tunjukkan berapa annotator yang melakukan labelling. |
| Dina | Analisis | 2018 | Metode analisis | Keakuratan | Tidak di |

| | | | | | |
|---|---|---|---|---|---|
| Agustina, Fitri Rahmah | Sentimen pada Sosial Media Twitter terhadap MRT Jakarta Menggunakan Machine Learning [7]. | | yang di gunakan adalah Naïve Bayes. Hasil penelitian Diperoleh model analisis text mining dengan Naïve Bayes memiliki akurasi sebesar 76,21%. | analisis di ukut menggunakan confussion matrix. | sebutkan metode pengumpulan datanya dan akurasi yang di peroleh dari analisis menggunakan naïve bayes sebesar 76,21 %. |
| Ahmad Rifa'i, Herry Sujaini, Dian Prawira. | Sentiment Analysis Objek Wisata Kalimantan Barat Pada Google Maps Menggunakan Metode Naive Bayes [8]. | 2021 | Metode pengumpulan data dilakukan dengan *crawling* menggunakan *Google Maps API.* Menggunakan Metode pembobotan kata menggunakan TF-IDF. Dan Teknik klasifikasinya dengan Naïve Bayes. Hasil yang di peroleh Nilai akurasi tertinggi adalah 0,76 sedangkan terendah adalah 0,38. | Dibuatkan sebuah website untuk penyajian hasil dari analisis, agar dapat di akses oleh wisatawan. Menggunakan Metode pembo botan kata menggunakan TF-IDF | Penelitian ini lebih banyak berfokus pada perancangan system dan penyajian data pada system tersebut. Data uji yang di gunakan hanya 50 per tempat wisata dan ada 10 tempat wisata di wilayah Kalimantan Barat yang di analisis. |
| Ryo Kusnadi dkk. | ANALISIS SENTIMEN TERHADAP GAME GENSHIN IMPACT MENGGUNA | 2021 | Menggunakan metode CRoss-Industry Standard Process for Data Mining (CRISP- DM) Menggunakan | Menggunakan *hugging face AI* sebagai tambahan Langkah *data preparation* agar proses | Dengan di gunakan model BERTBASE dan klasifikasi data yang di |

| | | | | | |
|---|---|---|---|---|---|
| | KAN BERT [9]. | | model BERTBASE dengan dibuatkan klasifikasi sentiment di atasnya.. Hasil yang di dapat model lebih mudah mendeteksi nilai positif karena memiliki nilai presisi tertinggi yaitu sebanyak 0.86% | penyiapan data lebih baik. | lakukan, belum bisa menganalisa nilai netral secara efektif. |
| Raden Mas Rizqi Wahyu Panca Kusuma Atmaja, Wiyli Yustanti | Analisis Sentimen Customer Review Aplikasi Ruang Guru dengan Metode BERT (Bidirectional Encoder Representations from Transformers) | 2021 | Metode pengumpulan data ulasan dengan *scrapping*. Model yang di gunakan adalah pretrained model dari BERT. | Menggunakan pretrained model dari BERT kemudian membuat model sendiri sesuai dengan memasukkan data latih dan uji untuk keperluan klasifikasi. Dan menggunakan pendekatan metodologi CRISP-DM yang merupakan proses standar terbuka lintas industri untuk data mining. | Penggunaan pretrained model dari BERT sangat spesifik untuk data / domain yang di latih saja, jika di gunakan untuk konteks yang lebih lebar atau keluar dari data / domain latih, maka akan efisiensi dan akurasinya akan turun. |

## 2.2 Landasan Teori

### 2.2.1 Kekerasan Verbal

Kekerasan verbal adalah bentuk kekerasan pada seseorang melalui kata-kata berupa penghinaan, ataupun kata kata yang melecehkan. Tujuannya adalah menyalahkan, merusak mental, menghina, atau merendahkan korbannya sehingga si korban akan merasa tidak percaya diri, mulai mempertanyakan intelejensi, hingga merasa tidak memiliki harga diri. Kekerasan verbal bisa terjadi pada hubungan apa pun dan intensitasnya biasanya meningkat bila tidak segera diakhiri. Jika sudah parah, kekerasan ini juga bisa berujung pada kekerasan fisik dan meninggalkan efek yang buruk bagi korbannya.

Menurut S.Putra (2015) kekerasan verbal banyak terjadi tanpa di sadari atau tidak di sengaja, hal ini disebabkan orang-orang terkadang tidak menyadari bahwa apa yang dilakukannya adalah kekerasan karena menganggap hal itu sudah biasa dan sebatas gurauan semata [10].

### 2.2.2 NLP (Natural Language Preprocessing)

Pengolahan Bahasa Alami (Natural Language Processing/NLP) merupakan salah satu bagian dari Ilmu Kecerdasan Buatan yang memfokuskan pada upaya mengembangkan cara agar komputer dapat memahami, menafsirkan, dan memproses bahasa alami dalam bentuk tulisan atau percakapan manusia. NLP menganalisis bahasa manusia sedemikian rupa sehingga komputer dapat memiliki pemahaman yang serupa dengan manusia terhadap bahasa alami. NLP adalah salah satu bidang ilmu yang menggabungkan komputasi linguistik, ilmu komputasi, ilmu kognitif, dan kecerdasan buatan. Pada umumnya, NLP merupakan bidang lintas disiplin yang menggabungkan elemen-elemen dari

komputasi linguistik, ilmu komputer, ilmu kognitif, dan kecerdasan buatan. Dalam praktiknya, NLP digunakan dalam berbagai bidang, termasuk pengenalan suara, pemahaman bahasa lisan, sistem dialog, analisis leksikal, mesin penerjemah, pengembangan pengetahuan grafik, analisis sentimen, serta sistem cerdas dan ringkasan bahasa alami. Pendekatan NLP dimulai dari level kata untuk mengidentifikasi struktur dan sifat morfologis (seperti bagian ucapan atau arti) dari setiap kata; kemudian berpindah ke level kalimat untuk menentukan susunan kata, tata bahasa, dan makna seluruh kalimat. Ini kemudian diperluas hingga ke konteks dan keseluruhan domain. Kata-kata atau kalimat yang disediakan dapat memiliki interpretasi atau konotasi yang berubah tergantung pada konteksnya.

### 2.2.3 Analisis Sentimen

Analisis sentimen merupakan salah satu bidang penelitian dalam komputasi yang fokus pada eksplorasi opini, sentimen, dan emosi dalam teks. Bidang ini dikenal juga dengan sebutan opinion mining dan telah menjadi perhatian utama dalam NLP dan penambangan data. Tujuan utama analisis sentimen adalah mengolah, mengekstrak, merangkum, dan menganalisis informasi teks menggunakan berbagai metode. Hal ini bertujuan untuk mengungkapkan emosi dan pandangan penulis, serta mengidentifikasi kecenderungan emosional dalam teks melalui informasi subjektif yang terkandung di dalamnya. Sentimen sendiri merujuk pada sikap positif atau negatif individu atau kelompok terhadap suatu hal.

### 2.2.4 BERT

Bidirectional Encoder Representations from Transformers (BERT), yang dikembangkan oleh tim peneliti di Google *AI Language* pada tahun 2018 [11], adalah sebuah model representasi bahasa yang telah melalui pelatihan intensif. BERT muncul berdasarkan prinsip-prinsip deep learning dan berbagai teknik seperti semi-supervised learning, ELMo, ULMFiT, OpenAI Transformers, dan Transformers.

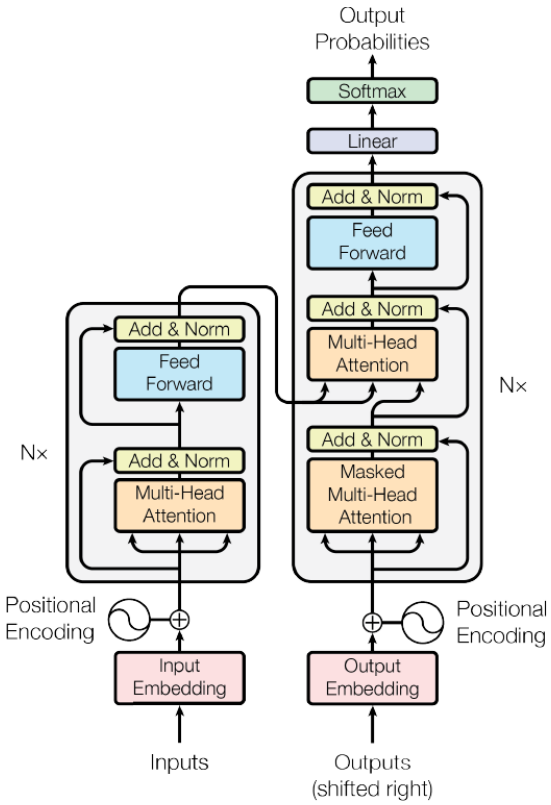
Sesuai dengan namanya, BERT menggunakan struktur Transformer yang memungkinkan pembelajaran konteks antara kata-kata dalam teks. Model Transformer ini memiliki kemampuan untuk memahami dan mengartikan pemahaman tersebut melalui mekanisme yang dikenal sebagai self-attention mechanism. Mekanisme self-attention ini memungkinkan Transformer untuk menghubungkan kata-kata dalam konteks yang lebih luas. Pada model Transformer, terdapat dua jenis mekanisme yang signifikan:

*a.* *Encoder*

*Encoder* berfungsi untuk membaca seluruh input teks sekaligus. *Encoder* terdiri dari stack (tumpukan) dari N = 6 *layers* yang identik. Setiap layer memiliki dua sub-layer yaitu *self-attention* layer dan *feed-forward neural network.* Dengan *self-attention* layer, encoder dapat membantu *node* untuk tidak hanya fokus kepada kata yang sedang dilihat tetapi juga untuk mendapatkan konteks semantik dari kata tersebut. Setiap posisi di *encoder* dapat menangani semua posisi di layer sebelumnya di *encoder*.

*b. Decoder*

*Decoder* berfungsi untuk menghasikan urutan *output* yang berupa prediksi. *Decoder* juga terdiri dari *stack* (tumpukan) dari N = 6 layers yang identif. Setiap *layer* terdiri dari dua sub-layer seperti yang ada pada *encoder*, dengan tambahan attention *layer* di antara dua layers tersebut untuk membantu *node* saat ini mendapatkan *key content* yang membutuhkan *attention* [12] dengan melakukan multi-head *attention* pada output dari *encoder*. Sama dengan di *encoder*, *self-attention* layer di *decoder* membuat setiap posisi di *decoder* dapat menangani semua posisi sebelumnya.




**Gambar 2.1 Encoder (kiri) dan Decoder (kanan)** [12]

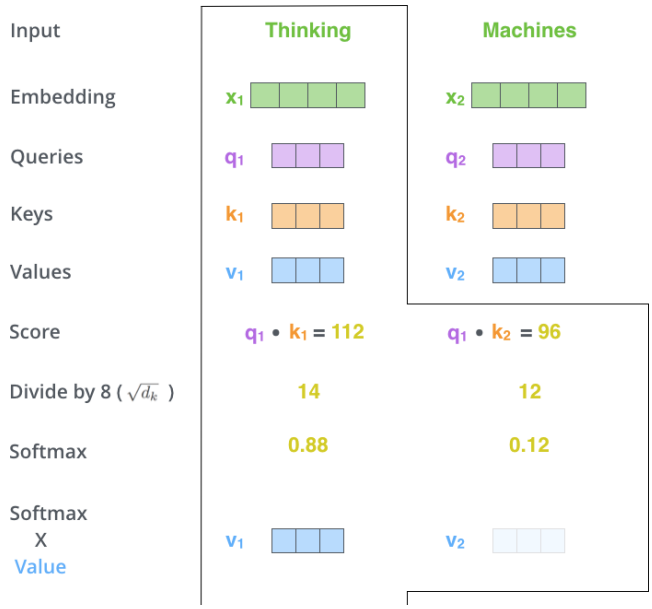
Langkah-langkah berikut menunjukkan proses yang terjadi pada *encoder* dan

*decoder* [13].

1. Setiap input kata yang memasuki *encoder* diubah menjadi sebuah *list vector* menggunakan *embeddings*. Karena *self-attention layer* tidak membedakan urutan kata-kata pada sebuah kalimat, *positional encoding* ditambahkan untuk menunjukkan posisi dari tiap kata. Tiap vektor dari input kata memiliki ukuran 512. Proses ini hanya terjadi di *encoder* yang berada paling bawah, sehingga*encoder* lainnya akan menerima *output* dari *encoder* yang pertama.

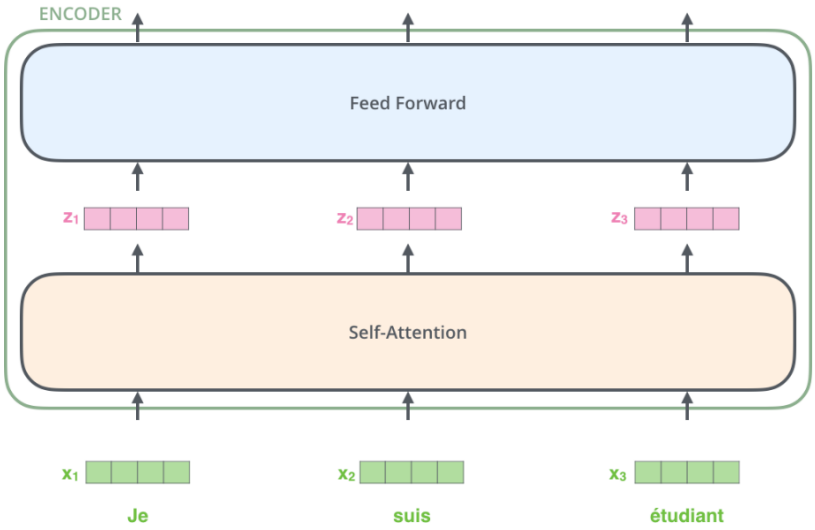
2. Input vektor melewati dua *layer* yang ada pada tiap *encoder* yaitu *self-attention laye*r dan *feed-forward neural network*. Pada *self-attention layer* dibuat tiga vektor dari masing-masing input vektor yaitu Query, Key, dan Value vector. Ketiga vektor ini dibuat dengan mengalikan *embedding*. Dimensi dari tiap vektor adalah 64. Setelah itu, nilai *self-attention* dari tiap kata dihitung dengan mengalikan *query vector* dan *key vector*.

   Kemudian, nilai *self-attention* dibagi 8 karena 8 adalah akar kuadrat dari dimensi tiap vektor yaitu 64. Nilai *self-attention* juga dihitung dengan *softmax* sehingga tiap *value vector* akan dikali dengan nilai dari *softmax*. Akhirnya *value vector* dijumlahkan dan menjadi output dari *self-attention layer*. Output




**Gambar 2.2 Proses pada *Self-attention Layer***

dari *self-attention layer* kemudian masuk ke *feed-forward* untuk tiap posisi seperti yang tertera pada Gambar 2.2.

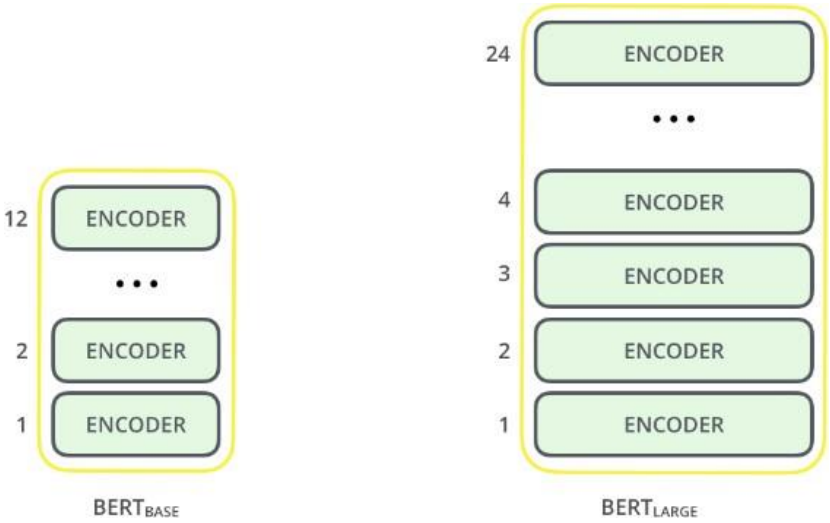



**Gambar 2. 3 Proses pada *Encoder***

3. Setelah setiap proses pada *encoder* selesai, *output* dari *encoder*[13] yaitu vector *key* dan vector *value* kemudian memasuki *decoder*. Tiap input dan *output* dari *self- attention layer* dan *feed-forward neural network* di *encoder* dan *decoder* diproses oleh *layer add & norm* yang berisi struktur residual dan normalisasi *layer*. Proses yang terjadi pada *decoder* sama dengan *encoder* akan tetapi di antara *self-attention layer* dan *feed-forward neural network* terdapat *attention layer* yang membantu *decoder* untuk fokus pada bagian-bagian dari kata yang relevan. *Self-attention layer* di *decoder* hanya boleh untuk menghadiri posisi sebelumnya dari *output*. *Output* dari tiap langkah dimasukkan ke dalam *decoder* terus menerus dan hasil dari *decoder* sama seperti hasil dari *encoder*. Akhirnya, *output* dari tumpukan *decoder*

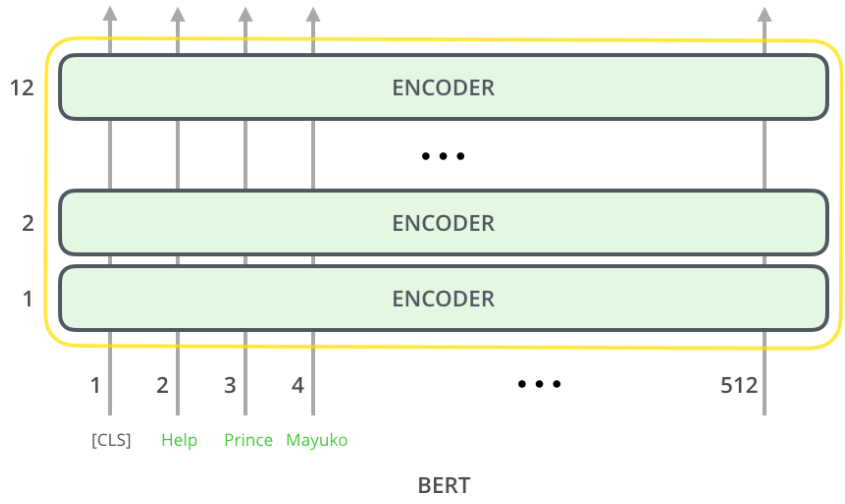
menghasilkan sebuah *vector* dengan *nilai float*. Untuk mengubahnya menjadi sebuah kata-kata, layer tambahan berupa *fully connected layer* dibutuhkan beserta *softmax layer*.

Arsitektur model BERT berupa *multi-layer bidirectional* Transformer seperti yang dilakukan pada implementasi asli Transformer tetapi hanya menggunakan proses sampai *encoder* saja. Pada implementasinya, terdapat dua ukuran model yang ada pada BERT, yaitu $BERT_{BASE}$ dan $BERT_{LARGE.}$ Kedua ukuran model BERT ini memiliki banyak lapisan *encoder* atau Transformer Blocks. $BERT_{BASE}$ memiliki *encoder* dengan 12 *layers*, 12 *self-attentions heads*, *hidden size* sebesar 768, dan 110M *parameters*. Sedangkan $BERT_{LARGE}$ terdapat 24 *layers*, 16 *self- attention heads*, *hidden size* sebesar 1024, dan 340M *parameters*. $BERT_{BASE}$ dilatih selama 4 hari menggunakan 4 cloud TPUs sedangkan $BERT_{LARGE}$ membutuhkan 4 hari menggunakan 16 TPUs.

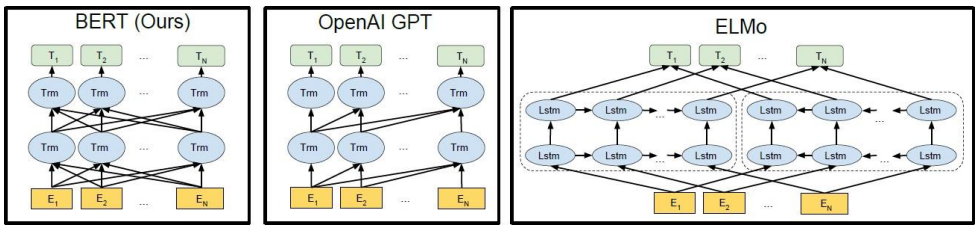



**Gambar 2.4 Perbedaan Ukuran $BERT_{BASE}$ dan $BERT_{LARGE}$**

Sesuai dengan namanya, BERT hanya menggunakan *encoder*. Sehingga arsitektur BERT terlihat seperti Gambar 2.5. BERT berbeda dengan model terarah (*directional*) yang melihat urutan teks dari kiri-ke-kanan, kanan-ke-kiri, atau gabungan dari kiri-ke-kanan dan kanan-ke-kiri. Model bahasa yang dilatih secara *bidirectional* dapat memiliki pemahaman yang lebih dalam tentang konteks daripada model bahasa satu arah. Gambar 2.6 menunjukkan perbandingan antara arsitektur BERT dengan OpenAI GPT dan ELMo. Di antara ketiga model arsitektur tersebut, hanya BERT yang secara bersamaan melihat kepada konteks kiri dan kanan di setiap layernya.




**Gambar 2.5 Arsitektur BERT**




**Gambar 2. 6 Perbedaan Arsitektur BERT Dengan OpenAI GPT dan ELMo**

BERT menggunakan WordPiece *embeddings* dengan 30,000 token

*vocabulary*. Token pertama dari tiap urutan selalu berupa token klasifikasi khusus yaitu [CLS] [11]. BERT dapat dilatih untuk memahami sebuah bahasa dan dapat pula disempurnakan (*fine-tune*) untuk mempelajari tugas-tugas tertentu. *Training* di BERT terdiri dari dua tahap, *pre-training* dan *fine-tuning*. Tahap pertama yaitu *pre-training* adalah tahap di mana BERT dibuat untuk memahami dan mempelajari bahasa dan konteksnya. BERT dapat memahami dengan *training* dengan dua tugas *unsupervised* yang dilakukan bersamaan yaitu Masked Language Model dan Next Sentence Prediction.

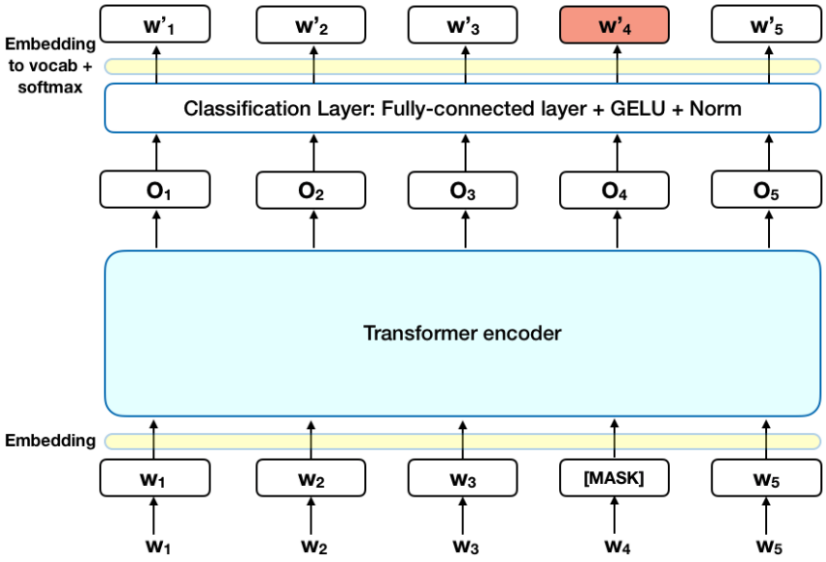
*1. Masked Language Modelling (Masked LM)*

Tujuan dari *Masked Language Modelling* adalah untuk memberi *mask* atau penutup ke kata secara acak pada kalimat dengan probabilitas yang kecil. Sebelum memasukkan urutan kata ke dalam BERT, 15% dari kata-kata di tiap urutan kata diganti dengan token [MASK]. Kemudian model akan mencoba untuk memprediksi nilai asli dari kata yang diberi [MASK] berdasarkan konteks yang diberikan oleh kata lain yang tidak ditutup dengan [MASK] di dalam urutan kata. Secara teknis, prediksi kata-kata *output*:

i) Membutuhkan lapisan klasifikasi di atas *output encoder*.

ii) Mengalikan *vector output* dengan matriks *embedding* kemudian mengubahnya menjadi *vocabulary dimension*.

iii) Menghitung probabilitas dari setiap kata di *vocabulary* dengan *softmax*.



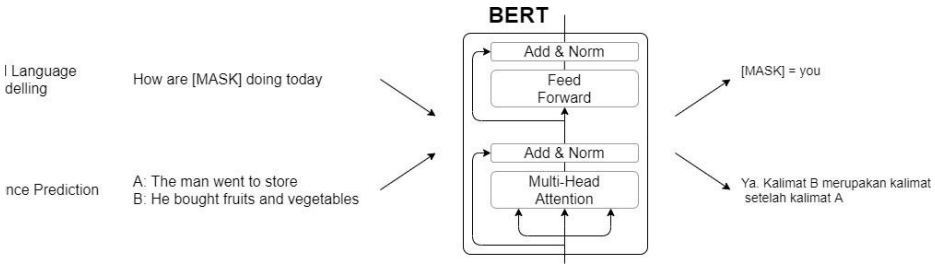

**Gambar 2.7 Proses *Masked Language Modelling***

*2. Next Sentence Prediction*

Dalam proses *training* BERT, model dapat menerima pasangan kalimat sebagai input dan dilatih untuk memprediksi jika kalimat kedua pada pasangan tersebut adalah kalimat berikutnya pada dokumen aslinya atau hanya satu kalimat saja. Selama *training*, 50% dari input adalah pasangan kalimat di mana kalimat kedua adalah kalimat berikutnya pada dokumen asli. Sedangkan 50% lainnya adalah kalimat yang diambil secara acak dari *corpus* sebagai kalimat kedua.

Sebagai representasi input pada BERT, terdapat tiga *embedding layers* yaitu:

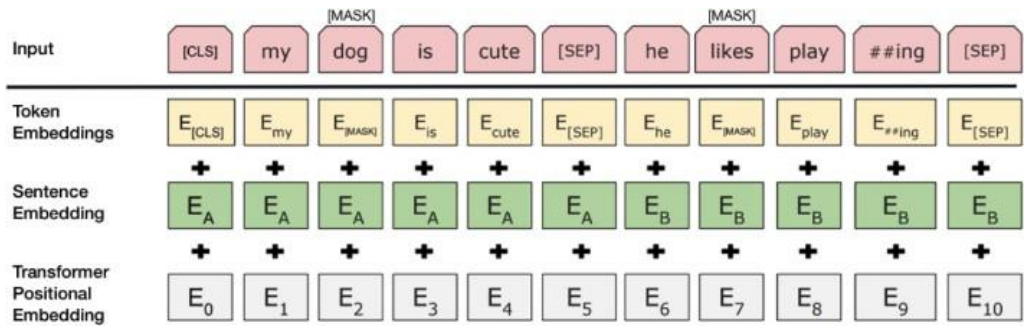
1. *Token embeddings* adalah *layer* pertama yang token masuki, yaitu




**Gambar 2.8 Proses *Pre-training* pada BERT**

representasi vektor dari tiap token. Setiap token dalam input akan dipetakan ke representasi vektor berdimensi tinggi dari token yang diberikan. Tiap token diganti menjadi id yang didapatkan berdasarkan *vocabulary*.*Sentence embeddings* menunjukkan kalimat pertama atau kalimat kedua, ditambahkan ke setiap token dan digunakan untuk membedakan antar kalimat jika terdapat lebih dari dua kalimat. Lapisan ini hanya memiliki dua representasi: A untuk token yang termasuk dalam kalimat pertama, dan B untuk token yang termasuk dalam kalimat kedua.

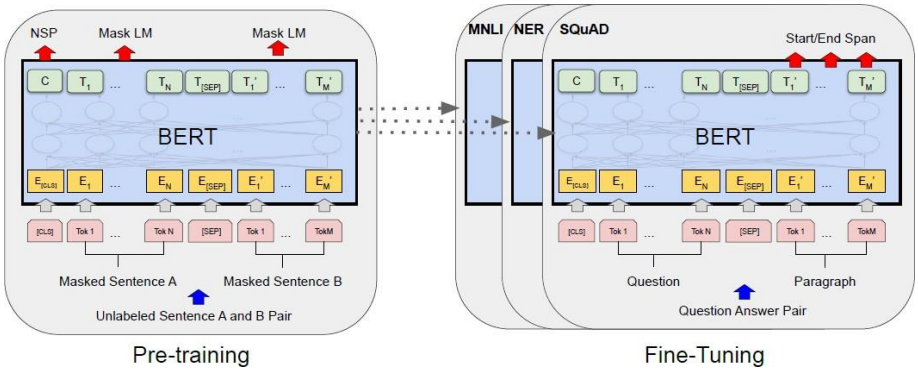
2. *Positional embedding* ditambahkan ke setiap token untuk menyimpan informasi tentang posisi kata dalam urutan. Konsep dan implementasi dari *positional embedding* ditunjukkan dalam Transformer. BERT telah mempelajari posisi *embedding layer* selama *pre-training*.



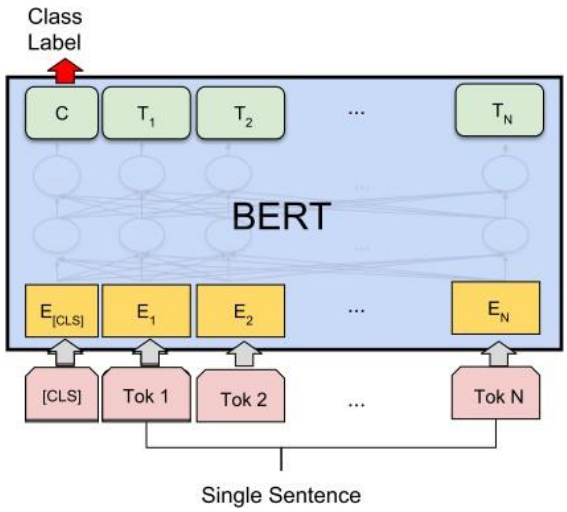

**Gambar 2.9 Representasi Input pada BERT** [11]

Untuk melatih sebuah model bahasa, *classifier* perlu dilatih dengan sedikit perubahan pada model BERT selama fase pelatihan (*training*) yang disebut *fine-tuning*. Seperti yang dipaparkan oleh Devlin dan rekan-rekannya, terdapat rekomendasi *hyperparameters* yang dapat di-*fine-tuning* untuk mencapai hasil yang maksimal. *Fine-tuning* sangat mudah dilakukan karena mekanisme *self-attention* di Transformer membuat BERT bisa membuat model untuk berbagai

tugas, baik pada kalimat tunggal (*single sentence*) atau kalimat berpasangan, dengan menukar masukan dan keluaran yang sesuai.




**Gambar 2.10 Prosedur Pre-*training* dan *Fine-tuning*** [11]




**Gambar 2. 11 Ilustrasi Fine-tuning pada Tugas dengan Single Sentence (Devlin et al., 2019)**

### 2.2.5 Deep Learning

Deep Learning, sebuah cabang dari Machine Learning yang telah mendorong terobosan besar dalam pemrosesan data dan kecerdasan buatan. Deep

Learning menggunakan arsitektur jaringan saraf tiruan (Artificial Neural Network - ANN) yang memiliki banyak lapisan (layer) untuk mempelajari representasi data secara hierarkis.

Perkembangan pesat dalam teknologi komputasi dan ketersediaan besar data telah menjadi pendorong utama dalam popularitas Deep Learning. Jaringan saraf tiruan dengan banyak lapisan (deep neural networks) memungkinkan model untuk secara otomatis memperoleh fitur-fitur yang relevan dari data tanpa perlu diprogram secara eksplisit. Hal ini memungkinkan Deep Learning untuk menyelesaikan tugas-tugas yang sangat kompleks seperti pengenalan gambar, analisis bahasa alami, dan permainan mesin.

Dalam literatur, Deep Learning pertama kali diusulkan oleh Hinton dan Salakhutdinov pada tahun 2006. Mereka mengusulkan metode pre-training deep neural networks secara berurutan untuk mempercepat konvergensi dan mengatasi masalah hilangnya gradien pada deep neural networks [14]. Metode ini membuka pintu bagi kemajuan pesat dalam bidang Deep Learning.

Salah satu jenis jaringan saraf tiruan yang sering digunakan dalam Deep Learning adalah Convolutional Neural Network (CNN). CNN telah menjadi metode yang sangat sukses dalam tugas pengenalan gambar dan penglihatan komputer. Konvolusi memungkinkan CNN untuk mendeteksi fitur-fitur visual seperti tepi, bentuk, dan pola yang kompleks dari data gambar.

Selain itu, Recurrent Neural Network (RNN) seperti yang telah dibahas sebelumnya dalam subbab 2.3, termasuk dalam keluarga jaringan saraf tiruan yang sering digunakan dalam tugas yang melibatkan urutan data, seperti teks, suara, dan data deret waktu.

Keberhasilan Deep Learning dalam berbagai tugas telah mendorong banyak penelitian di Indonesia. Beberapa penelitian yang relevan dengan aplikasi Deep Learning dalam konteks Indonesia antara lain adalah penelitian oleh Fitroh [15] metode yang digunakan mendapat akurasi yang baik dalam mendeteksi citra dermoskopi dan dapat membantu dermatologi dalam deteksi dini kanker kulit.

### 2.2.6 Bi-LSTM

Bidirectional LSTM (Bi-LSTM) dalam analisis sentimen, sebuah pendekatan yang memanfaatkan fitur Bi-LSTM untuk memahami konteks dari kedua arah data berurutan. Bi-LSTM merupakan perkembangan dari arsitektur LSTM yang memungkinkan model untuk memiliki pandangan konteks dari teks atau urutan data dalam kedua arah maju (forward) dan mundur (backward).

Analisis sentimen adalah tugas yang kompleks dan membutuhkan pemahaman yang baik tentang konteks kata atau kalimat dalam data berurutan. Dengan menggunakan Bi-LSTM, model dapat mengeksplorasi konteks sebelum dan sesudah suatu kata atau kalimat, yang memungkinkan model untuk lebih akurat dalam mengenali sentimen dari teks.

Bi-LSTM memiliki dua lapisan LSTM yang berjalan dalam arah berlawanan. Lapisan pertama berjalan maju (forward) dari awal hingga akhir urutan data, sementara lapisan kedua berjalan mundur (backward) dari akhir hingga awal urutan data. Informasi dari kedua lapisan ini digabungkan untuk memberikan pemahaman yang lebih lengkap tentang konteks kata atau kalimat dalam analisis sentimen.

Salah satu keunggulan Bi-LSTM dalam analisis sentimen adalah kemampuannya untuk menangani masalah long-term dependencies, yaitu

hubungan antara kata atau kalimat yang berjauhan secara temporal dalam data berurutan. Dengan melibatkan konteks dari kedua arah, Bi-LSTM mampu mengatasi kendala tersebut dan mengenali sentimen yang berkaitan dengan konteks keseluruhan dari teks.

### 2.2.7 Semi-Supervised Learning

Semi-Supervised Learning dengan menggunakan metode Bidirectional Long Short-Term Memory (Bi-LSTM) dalam labeling data sentimen, sebuah pendekatan yang memanfaatkan data yang tidak sepenuhnya berlabel (unlabeled data) bersama dengan data yang berlabel (labeled data) untuk meningkatkan efisiensi dalam tugas analisis sentimen.

Analisis sentimen adalah proses untuk menentukan sentimen atau perasaan dari teks, seperti positif, negatif, atau netral. Dalam konteks Semi-Supervised Learning untuk analisis sentimen, kita memiliki data yang tidak berlabel dan data yang berlabel. Bi-LSTM adalah varian dari arsitektur LSTM yang memiliki dua lapisan LSTM, satu berjalan maju (forward) dan satu berjalan mundur (backward), sehingga memungkinkan model untuk memahami konteks dari kedua arah dalam urutan data.

Penerapan Semi-Supervised Learning dengan Bi-LSTM dalam analisis sentimen melibatkan penggunaan data yang tidak berlabel untuk membantu dalam proses pembelajaran. Bi-LSTM yang telah dilatih dengan menggunakan data yang berlabel akan menghasilkan representasi yang lebih baik dari teks, dan pengetahuan ini dapat diterapkan pada data yang tidak berlabel.

Kelebihan dari penggunaan Bi-LSTM dalam analisis sentimen adalah kemampuannya untuk memperoleh konteks dari kedua arah dalam urutan data,

sehingga dapat meningkatkan pemahaman dan akurasi analisis sentimen. Selain itu, Bi-LSTM juga dapat menangani informasi kontekstual yang lebih kaya daripada arsitektur LSTM tradisional.

Penelitian tentang penerapan Semi-Supervised Learning dengan menggunakan Bi-LSTM dalam labeling data sentimen juga telah dilakukan di Indonesia. Penelitian relevan diantaranya adalah penelitian anotasi semi otomatis dengan membandingkan 3 metode yaitu CNN, LSTM dan Bi-LSTM, hasil yang di dapat adalah LSTM menunjukkan kinerja yang lebih baik daripada CNN, namun Bi-LSTM menunjukkan kinerja lebih baik daripada LSTM tradisional [16].

Kelebihan dari penerapan Semi-Supervised Learning dengan menggunakan Bi-LSTM dalam analisis sentimen adalah kemampuannya untuk memanfaatkan data yang tidak berlabel untuk meningkatkan performa model dan mengurangi ketergantungan pada data yang berlabel, sehingga dapat meningkatkan efisiensi dan akurasi hasil.

### 2.2.8 INDOBERT

INDOBERT adalah model bahasa alami berbasis deep learning yang dikembangkan khusus untuk bahasa Indonesia. Model ini menggunakan arsitektur BERT (Bidirectional Encoder Representations from Transformers) yang dikembangkan oleh Google pada tahun 2018. INDOBERT merupakan hasil kerja sama antara Institut Teknologi Bandung (ITB), Institut Teknologi Sepuluh Nopember (ITS), dan Universitas Kristen Petra (UK Petra) dengan dukungan dari

INDOBERT memiliki 12 lapisan (layer) encoder dan lebih dari 100 juta parameter. Model ini dilatih menggunakan korpus bahasa Indonesia yang besar

dan beragam, termasuk data dari Wikipedia, Twitter, dan berbagai sumber online lainnya. Proses pelatihan dilakukan dengan memanfaatkan teknik transfer learning, yaitu dengan menggunakan model BERT yang telah dilatih pada korpus bahasa Inggris (BERT-base) sebagai dasar awal untuk melatih model INDOBERT pada korpus bahasa Indonesia.

Dalam melakukan transfer learning, model INDOBERT dilatih dengan dua tahap. Tahap pertama dilakukan pada model BERT-base dalam bahasa Inggris, di mana parameter awal model diatur sedemikian rupa sehingga dapat membaca dan memahami teks dalam bahasa Indonesia. Tahap kedua melibatkan proses fine-tuning pada data bahasa Indonesia, di mana model disesuaikan untuk tugas tertentu seperti analisis sentimen atau klasifikasi teks.

Model INDOBERT telah terbukti efektif dalam berbagai tugas NLP, termasuk dalam analisis sentimen, klasifikasi topik, dan pemrosesan bahasa alami lainnya. Performa model ini dapat ditingkatkan melalui proses fine-tuning pada data khusus untuk tugas tertentu, sehingga hasilnya lebih akurat dan dapat diandalkan.

Dalam keseluruhan, INDOBERT adalah model bahasa alami berbasis deep learning yang dikembangkan khusus untuk bahasa Indonesia, dan mampu memberikan representasi kata-kata yang lebih baik serta memiliki performa yang efektif dan dapat diandalkan dalam berbagai tugas NLP.