

BAB II

LANDASAN TEORI

2.1 Kajian Literatur

Serangkaian perangkat dan prosedur elektronik yang menyiapkan, mengumpulkan, mengolah, menganalisis, menyimpan, menampilkan, mengumumkan, mengirimkan, dan/atau menyebarluaskan informasi elektronik disebut sebagai sistem elektronik, menurut laman Kominfo. penyelenggara Setiap orang, penyelenggara negara, badan usaha, masyarakat, atau badan lain yang menyediakan, mengelola, dan/atau menyelenggarakan sistem elektronik untuk kebutuhan pengguna sendiri atau pihak lain adalah sistem elektronik (PSE). Pendaftaran Penyelenggara Sistem Elektronik memiliki landasan hukum yang jelas. Undang-Undang Nomor 11 Tahun 2008, sebagaimana telah diubah dengan Undang-Undang Nomor 19 Tahun 2016 tentang Perubahan atas Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (ITE), menjadi landasan hukumnya. Penyelenggaraan Sistem Elektronik dan Transaksi (PSTE) dalam PP No. 71/2019; Tata Cara Pendaftaran Penyelenggara Sistem Elektronik, PM Kominfo No. 36/2014; Tata Cara Pendaftaran Sistem Elektronik Badan Penyelenggara Negara, PM Kominfo No. 10/2015; demikian pula PM Kominfo No. 7/2019 tentang Pelayanan Perizinan Berusaha Terpadu Informasi dan Komunikasi. Setiap Penyelenggara Sistem Elektronik wajib melakukan registrasi sesuai dengan PP PSTE. Penyelenggara Sistem Elektronik di ruang publik merupakan salah satu dari dua jenis PSE pendaftaran yang dapat diselesaikan: dan Penyelenggara Sistem Elektronik di swasta. (Inews.id)

2.2 Natural Language Processing (NLP)

Salah satu bidang Artificial Intelligence yang mempelajari dan mengembangkan bagaimana komputer dapat memahami, memahami, dan memproses bahasa alami dalam bentuk teks atau ucapan dikenal sebagai Natural Language Processing. NLP menganalisis bahasa manusia sedemikian rupa cara komputer dapat memahami bahasa sebaik manusia. Linguistik komputasi, ilmu komputasi, ilmu kognitif, dan kecerdasan buatan adalah bagian dari bidang interdisipliner yang dikenal sebagai NLP. Pengenalan ucapan, pemahaman bahasa lisan, sistem dialog, analisis leksikal, terjemahan mesin, grafik pengetahuan, analisis sentimen, sistem cerdas, dan ringkasan bahasa alami hanyalah beberapa dari banyak aplikasi NLP.

Sistem NLP dapat dimulai dengan menentukan struktur dan sifat morfologis kata, seperti makna atau part-of-speech, pada tingkat kata setelah itu, dapat memeriksa urutan kata, tata bahasa, dan makna seluruh kalimat di tingkat kata. tingkat kalimat. Konteks dan seluruh domain mengikuti. Dalam konteks tertentu, satu kata atau kalimat mungkin memiliki arti atau konotasi berbeda yang terhubung ke banyak kata atau kalimat lain dalam konteks.

2.3 Analisis Sentimen

Subbidang penelitian komputasi yang dikenal sebagai "analisis sentimen" menyelidiki opini, perasaan, dan emosi teks. Dalam NLP dan penambangan data, analisis sentimen, juga dikenal sebagai penambangan opini, telah menjadi topik hangat. Tujuan utama analisis sentimen adalah untuk mengolah, mengekstrak, meringkas, dan menganalisis informasi dalam sebuah teks menggunakan

berbagai metode sehingga dapat menyimpulkan emosi dan sudut pandang penulis dari teks dan berbagi informasi subjektif penulis tentang kecenderungan emosional teks termasuk di dalamnya. Sikap positif atau negatif seseorang atau kelompok terhadap sesuatu itulah yang dimaksud dengan istilah "sentimen." Lima kali lipat dari ei, aij, sijkl, hk, dan tl dapat digunakan untuk mewakili pendapat atau sentimen, dengan ei mewakili entitas atau target pendapat, aij mewakili aspek sasaran opini, sijkl mewakili pendapat atau sentimen yang diberikan kepada sasaran, hk mewakili pemegang atau pemberi opini, dan t mewakili waktu opini diberikan.

Karena teks opini merupakan bagian dari data yang tidak terstruktur, maka diperlukan preprocessing untuk menyusun data tersebut. Data juga dapat diproses untuk mengambil aspek-aspek yang ada melalui tokenization, segmentasi kata, Part-of-Speech Tagging, dan stemming, antara lain (Zulfa & Winarko, 2017).

Level dokumen, level kalimat, dan level fine-grained adalah tiga level di mana analisis sentimen biasanya beroperasi. Ada juga level kasar pada level kalimat dan dokumen. Strategi dalam pemeriksaan opini dibagi menjadi dua macam, di tertentu berbasis pembelajaran dan berbasis leksikal. Sementara berbasis leksikal mengandalkan leksikon opini (kamus), pembelajaran berbasis mengandalkan data pelatihan dan pengujian data (Alwasi'a A, 2020).

2.4 Machine Learning

Dalam Artificial Intelligence, salah satu bidang keilmuannya adalah machine learning atau pembelajaran mesin. Seperti namanya, tujuan pembelajaran mesin adalah melatih mesin dengan banyak contoh atau kumpulan data yang terkait dengan tugas yang sedang dikerjakan. Berdasarkan kumpulan data tersebut, mesin

mempelajari pola yang diberikan dan membuat aturannya sendiri. Oleh karena itu, ketika salah satu data dimasukkan ke dalam mesin, mesin dapat mengenali data tersebut. Secara umum, machine learning dibagi menjadi empat kategori besar, yaitu supervised learning, unsupervised learning, self-supervised learning, dan reinforcement learning (Chollet, 2018). Pembelajaran terawasi adalah pendekatan yang paling sering digunakan. Pembelajaran yang diawasi membuat pembelajaran mesin dari kumpulan data yang telah diberi label atau diberi keterangan. Sedangkan unsupervised learning sebaliknya, dengan menyediakan dataset yang tidak berlabel. Self Supervised learning adalah pembelajaran terawasi tetapi tanpa dataset yang diberi label oleh annotator. Dataset yang digunakan masih menggunakan label tetapi label diperoleh dari data input menggunakan algoritma heuristik (Chollet, 2018). Algoritma yang sering digunakan dalam machine learning antara lain K-Nearest Neighbor, Naïve-Bayes, Support Vector Machine, K-Means, dan lain-lain.

2.5 Neural Network

Salah satu metode yang paling umum untuk pembelajaran mesin adalah jaringan saraf, juga dikenal sebagai jaringan saraf tiruan. Ia bekerja dengan mensimulasikan mekanisme pembelajaran yang didasarkan pada cara kerja sistem saraf manusia atau makhluk hidup lainnya. Sel-sel yang membentuk sistem saraf disebut neuron. Akson dan dendrit menghubungkan neuron ini satu sama lain. Menurut Aggarwal (2018), sinapsis adalah penghubung antara dendrit dan akson.

Arsitektur jaringan adalah arsitektur di mana berbagai lapisan terhubung satu sama lain. Lapisan tersembunyi dan unit tersembunyi adalah lapisan yang masing-masing terletak di antara lapisan input dan lapisan output (Osinga, 2018). Unit-

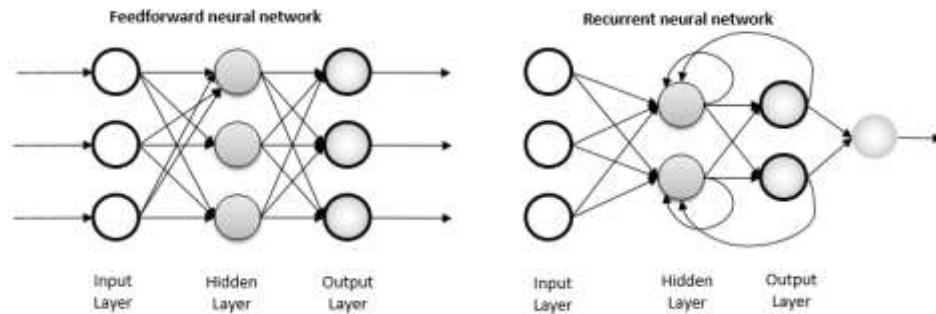
unit ini disebut sebagai "tersembunyi" karena tidak dapat diamati secara langsung sebagai input atau output dari luar. Lapisan tersembunyi dari unit tersembunyi, yang masing-masing merupakan unit saraf, bertindak sebagai inti jaringan saraf dengan menerapkan non-linearitas ke jumlah bobot masukan. Untuk menghubungkan lapisan, setiap unit di setiap lapisan mengambil input dan output dari semua unit di lapisan sebelumnya, serta hubungan antara setiap pasangan unit dari dua lapisan yang berdekatan. Setiap unit rahasia akan mencakup semua unit informasi (Jurafsky dan Martin, 2019).

Jaringan feed-forward dan jaringan berulang / rekursif adalah dua kategori utama arsitektur jaringan saraf.

1) Multi-Layer Perceptrons (MLP) atau Feed-forward Network Jaringan di mana unit terhubung tanpa siklus dan outputnya dikembalikan ke lapisan bawah dikenal sebagai jaringan feed-forward. Jaringan dapat berfungsi dengan input baik tetap ukuran atau panjang variabel yang dapat diabaikan oleh urutan elemen karena hal ini. Jaringan memperoleh kemampuan untuk menggabungkan komponen input saat dimasukkan ke dalam sistem. Dari input ke output, data hanya mengalir dalam satu arah. Pengenalan pola memanfaatkan jenis jaringan yang lebih sederhana ini. Jenis jaringan feedforward tertentu yang dikenal sebagai Convolutional Neural Network (CNN atau ConvNet) sering digunakan dalam pengenalan gambar.

2) Ketika ada input berurutan, jaringan saraf berulang (RNN) sering digunakan. Sebagian besar waktu, input ini ditemui ketika jaringan memproses suara atau teks. RNN mengambil urutan item sebagai inputnya dan menghasilkan fixed-ukuran vektor yang merangkum urutan. Data dapat melakukan perjalanan

melalui jaringan dalam dua arah menggunakan RNN, yang juga memungkinkan untuk loop. Dibandingkan dengan CNN, jaringan ini lebih kuat dan kompleks (Cahyadi et al., 2020).



Gambar 2. 1 Feed-forward neural network da Recurrent neural network (Pekel & Kara, 2017)

Setiap lapisan diikuti oleh fungsi aktivasi, fungsi matematika yang digunakan untuk memetakan output dari satu lapisan ke input lapisan berikutnya. Salah satu fungsi aktivasi adalah fungsi softmax. Fungsi aktivasi softmax memastikan bahwa jumlah vektor keluaran tepat 1. Simpul keluaran dengan probabilitas tertinggi kemudian dipilih sebagai label prediktif untuk kalimat masukan (Munikar et al., 2019). Fungsi aktivasi ini cocok untuk jaringan yang membutuhkan output dengan label yang telah ditentukan sebelumnya. Misalnya dalam jaringan yang dilatih untuk mengidentifikasi kucing dan tikus. Jika input untuk identifikasi kucing memiliki vektor keluaran 0,65, jaringan melihat kucing dengan kepastian 65%. Softmax hanya berfungsi jika hanya ada satu jawaban (Osinga, 2018). Simpul keluaran dengan probabilitas tertinggi kemudian dipilih sebagai label prediktif untuk kalimat yang menjadi masukannya (Munikar et al., 2019). Rumus fungsi softmax dirumuskan dalam persamaan 2.1

$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \dots\dots\dots(2.1)$$

Dengan $i = 1, \dots, K$ di mana:

1. $z = (z_1, \dots, z_k) \in \mathbb{R}^K$ adalah vektor input yang dimasukkan dalam fungsi softmax atau output dari lapisan terakhir yang disebut juga dengan logit. Masukan yang dapat diterima adalah bilangan real.
2. e^{z_i} adalah eksponensial dari tiap elemen dari input vektor.
3. $\sum_{j=1}^K e^{z_j}$ adalah proses normalisasi untuk memastikan semua nilai output dari softmax akan berjumlah tepat 1 dan masing-masing nilai berada di antara kisaran (0,1)

Jaringan akan menampilkan kerugian selama proses pelatihan. Setelah beberapa iterasi, jika nilai kerugian ini tidak berubah, jaringan tidak belajar apa pun dari proses sebelumnya. Akibatnya, untuk mengoptimalkan output jaringan saraf, kami memerlukan loss function. Kemampuan misfortune mengambil prakiraan yang diberikan oleh organisasi otak dan tujuan (apa yang perlu diciptakan organisasi) dan menghitung skor jarak, dengan tujuan dapat mengetahui seberapa baik organisasi. Keluaran fungsi kerugian kemudian digunakan sebagai sinyal umpan balik untuk sedikit menyesuaikan bobot untuk menurunkan skor kerugian. Salah satu fungsi kerugian yang biasanya digunakan dalam jaringan saraf adalah cross-entropy loss, yang sering dipasangkan dengan softmax. Untuk variabel acak tertentu, perhitungan antara dua distribusi probabilitas menghasilkan fungsi kerugian ini. Kemampuan model klasifikasi untuk menghasilkan output probabilitas dengan nilai antara 0 dan 1 dihitung menggunakan cross-entropy loss. fungsi loss ini dapat digunakan ketika output yang diberikan berupa probabilitas.

$$L = \sum_{i=1}^k y_i \log(o_i) \dots \dots \dots (2.2)$$

di mana:

1. y_i adalah label dari klasifikasi
2. o_i adalah probabilitas yang diprediksi oleh model terhadap label
3. $\log(o_i)$ adalah nilai logaritma dari tiap probabilitas yang diprediksi oleh model Jaringan syaraf tiruan memiliki beberapa parameter, seperti weights W , bias b yang dipelajari oleh gradient descent.

Hyperparameters, di sisi lain, adalah parameter yang dipilih oleh pembuat algoritme, seperti nilai optimal yang ditetapkan pada set pengembangan, dan tidak melibatkan penurunan gradien dalam set data pelatihan. Menurut Jurafsky & Martin (2019), hiperparameter termasuk kecepatan pembelajaran, ukuran mini-batch, arsitektur model (jumlah lapisan, jumlah node tersembunyi per lapisan, dan fungsi aktivasi yang dipilih), dan lain-lain.

Slope plunge adalah perhitungan peningkatan yang digunakan saat menyiapkan model AI. Hanya nilai parameter fungsi (koefisien) dengan biaya terendah yang dapat ditentukan menggunakan penurunan gradien. Pengurangan fungsi kerugian seminimal mungkin adalah tujuan utama penurunan gradien Laju pembelajaran, yang menunjukkan seberapa cepat atau lambat fungsi bergerak menuju bobot optimal, menentukan jumlah langkah yang dilakukan penurunan gradien. Turunan Gradien Batch, Turunan Gradien Stokastik, dan Turunan Gradien Mini-Batch adalah tiga yang paling sering digunakan metode penurunan gradien.

1. Kesalahan setiap instance dalam set data pelatihan dihitung menggunakan vanilla gradient descent atau batch gradient descent (BGD). Proses ini mirip dengan siklus epoch pelatihan. Semua biaya pelatihan set data dihitung. Jaringan saraf tiruan dapat beroperasi lebih cepat dan menghasilkan gradien kesalahan yang stabil berkat algoritme ini. Namun, model tersebut

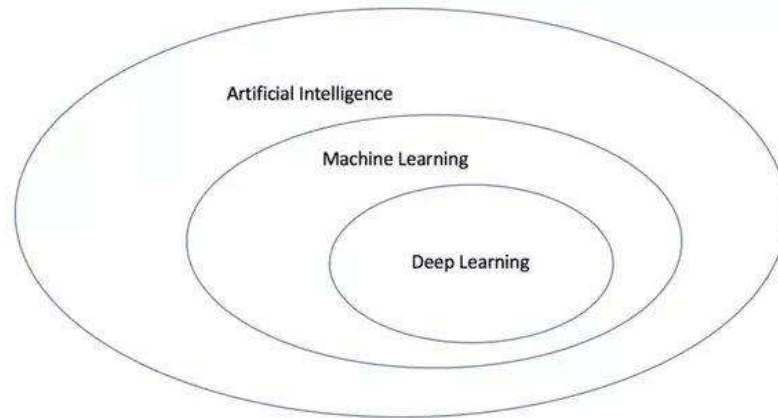
mungkin juga tidak dapat menghasilkan kondisi konvergensi yang optimal jika gradien kesalahannya stabil. Selain itu, memori harus menampung seluruh dataset pelatihan.

2. Menurut Jurafsky & Martin (2019), algoritma stochastic gradient descent (SGD) meminimalkan fungsi kerugian dengan menghitung gradiennya setiap kali pelatihan dilakukan. Karena hanya menggunakan satu sampel acak pada satu waktu dan menggeser bobotnya untuk meningkatkan kinerja sampel tersebut, algoritme ini disebut sebagai stokastik. Akibatnya, SGD akan menjalankan prosedur satu per satu pada setiap kumpulan data. Bobot dapat sering diperbarui, memungkinkan peningkatan yang sangat spesifik. Namun, karena tingkat kesalahan dapat berfluktuasi, sering terjadi pembaruan lebih mahal dan dapat menghasilkan gradien yang tidak rata.
3. Ide SGD dan FGD digabungkan dalam penurunan gradien batch mini. Proses pelatihan satu set sampel dataset, biasanya 512 atau 1024, yang lebih kecil dari dataset asli disebut mini batching. Dataset pelatihan akan dipecah menjadi batch yang lebih kecil oleh algoritma ini, dan bobot setiap batch akan diperbarui.

2.6 Deep Learning

Subbidang kecerdasan buatan yang dikenal sebagai pembelajaran mendalam adalah subbidang pembelajaran mesin. Menurut Hollet (2018), pembelajaran mendalam adalah jenis jaringan saraf yang lebih maju dan dalam daripada jaringan saraf standar karena jumlah lapisannya meningkat secara signifikan. Jumlah lapisan tersembunyi dalam jaringan disebut sebagai "kedalamannya", dan semakin banyak pemain, semakin dalam pembelajaran jaringan. Menurut Goldberg (2017), deep learning belajar memprediksi dan merepresentasikan data dengan benar,

menjadikannya cocok untuk prediksi. Ada tiga pendekatan untuk pembelajaran mendalam: pembelajaran terawasi, semi-terawat, dan tanpa pengawasan. Banyak kerangka kerja, termasuk Torch, Theano, TensorFlow, dan lainnya, memberikan dukungan untuk pembelajaran mendalam.



Gambar 2. 2 Hubungan antara Kecerdasan Buatan, Machine Learning, dan Deep Learning (Chollet, 2018).

2.7 Bidirectional Encoder Representations from Transformers (BERT)

Representasi Encoder Bidirectional Transformers (Devlin et al., 2019), atau disingkat BERT, adalah model representasi bahasa terlatih yang dibuat pada tahun 2018 oleh peneliti Google AI Language. BERT dibuat menggunakan pembelajaran semi-diawasi, ELMo, ULMFiT, OpenAI Transformers, dan Transformers serta teknik deep learning. Seperti namanya, BERT menggunakan Transformers serta teknik deep learning. Seperti namanya, BERT menggunakan Transformers. Transformer adalah komponen yang berkonsentrasi pada hubungan yang relevan antara kata-kata dalam teks (Vaswani et al., 2017). Pemahaman mekanisme self-attention dapat dipahami dan ditransformasikan oleh transformer. Transformer menggunakan mekanisme self-attention untuk mengubah

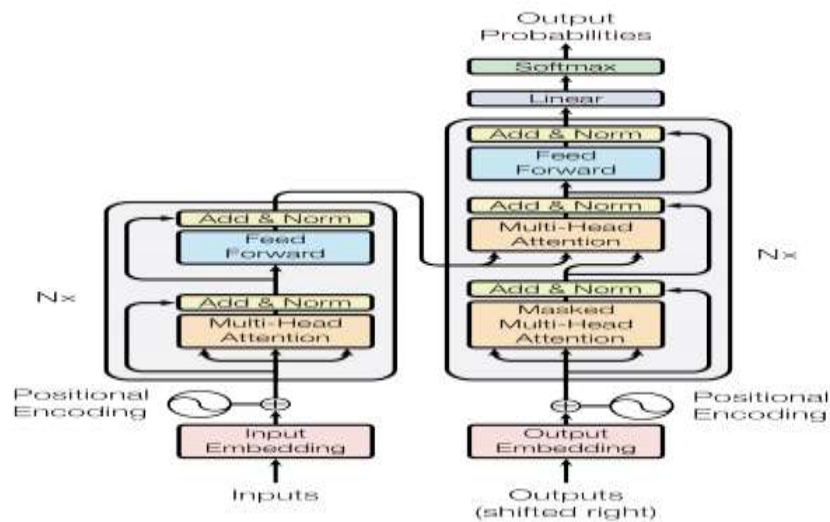
"pemahaman" dari kata-kata terkait lainnya menjadi kata-kata yang akan diproses oleh mekanisme tersebut. Ada dua mekanisme dalam Transformer:

a. Encoder

Encoder digunakan untuk membaca dengan teliti semua input teks tanpa penundaan sesaat. Tumpukan N lapisan yang identik membentuk encoder. Jaringan saraf feed-forward dan lapisan self-attention adalah dua sub-lapisan dari setiap lapisan. Encoder dapat membantu node menggunakan lapisan self-attention untuk tidak hanya fokus pada kata yang dilihat tetapi juga mempelajari konteks semantiknya. Semua posisi di lapisan encoder sebelumnya dapat ditangani oleh posisi mana pun di encoder.

b. Decoder

Urutan keluaran prediktif dihasilkan oleh dekoder. Selain itu, ada tumpukan N, atau enam, lapisan pengenalan di dekoder. Mirip dengan encoder, setiap lapisan memiliki dua sub-lapisan dan lapisan perhatian tambahan di antara mereka untuk membantu node saat ini dalam memperoleh konten penting yang memerlukan perhatian (Vaswani et al., 2017) dengan memperhatikan output dengan banyak kepala. melalui encoder. Lapisan self-attention di decoder, seperti encoder, memungkinkan setiap posisi untuk menangani semua posisi sebelumnya dan saat ini.

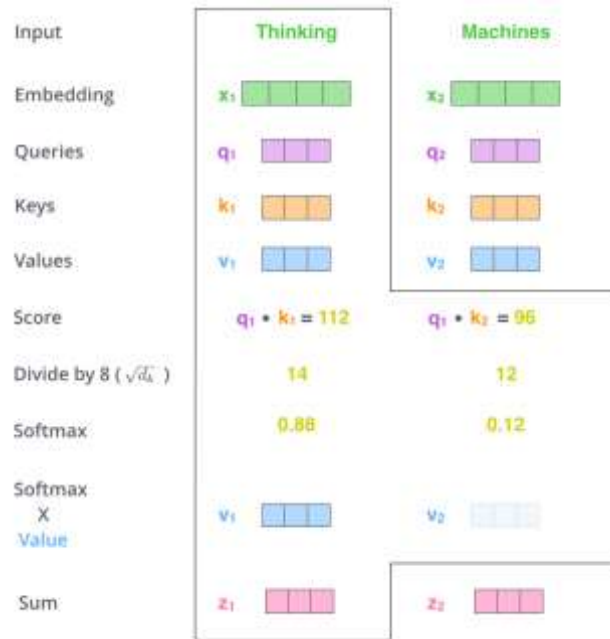


Gambar 2. 3 Encoder (kiri) dan Decoder (kanan) (Vaswani et al., 2017)

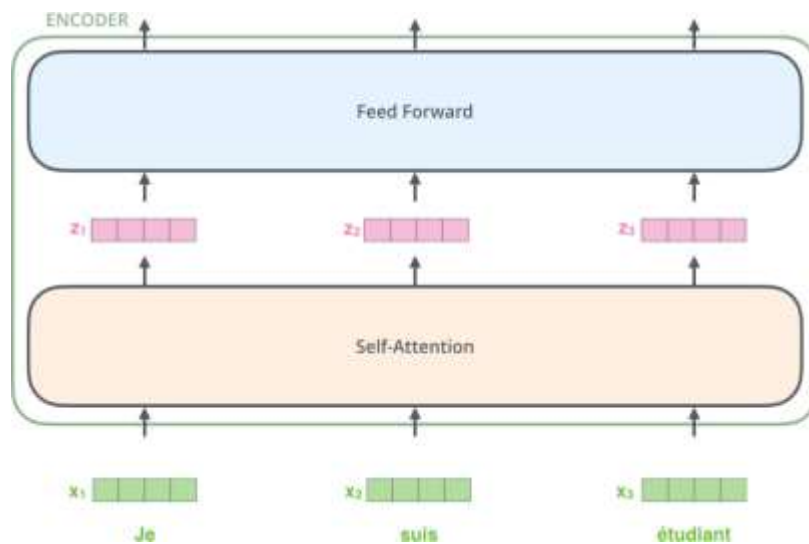
Langkah-langkah berikut menunjukkan proses yang terjadi pada encoder dan decoder (Alammar, 2018):

1. Menggunakan embeddings, encoder mengubah setiap kata input menjadi daftar vektor. Enkoding posisi ditambahkan ke lapisan self-attention untuk menunjukkan posisi setiap kata karena lapisan self-attention tidak membedakan urutan kata dalam kalimat. Ukuran dari setiap vektor kata input adalah 512. Karena proses ini hanya terjadi pada encoder terendah, output dari encoder pertama akan dikirim ke encoder kedua.
2. Lapisan self-attention dan jaringan saraf feed-forward adalah dua lapisan yang digunakan setiap encoder untuk memproses vektor input. Dari setiap vektor input, tiga vektor—vektor Query, Key, dan Value—dihasilkan dalam self-attention layer. Embedding dikalikan untuk membuat ketiga vektor ini. Setiap vektor memiliki dimensi 64. Setelah itu, seperti yang digambarkan pada Gambar 2.4, vektor kueri dan vektor kunci dikalikan untuk menentukan nilai self-attention setiap kata. nilai self-attention kemudian dibagi 8, karena 64 adalah akar kuadrat dari setiap dimensi vektor. Setiap nilai vektor akan

dikalikan dengan nilai softmax ketika nilai self-attention juga dihitung menggunakan softmax. Lapisan self-attention terakhir output adalah jumlah dari vektor nilai. Output lapisan self-attention kemudian diumpungkan ke depan untuk setiap posisi, seperti yang digambarkan pada Gambar 2.5.



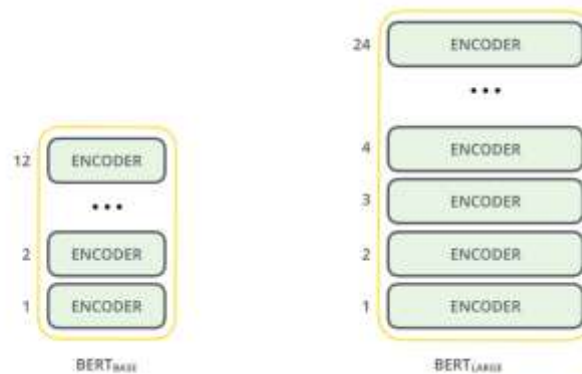
Gambar 2. 4 Proses pada Self-attention Layer (Alammar, 2018)



Gambar 2. 5 Proses Encoder (Allamar,2018)

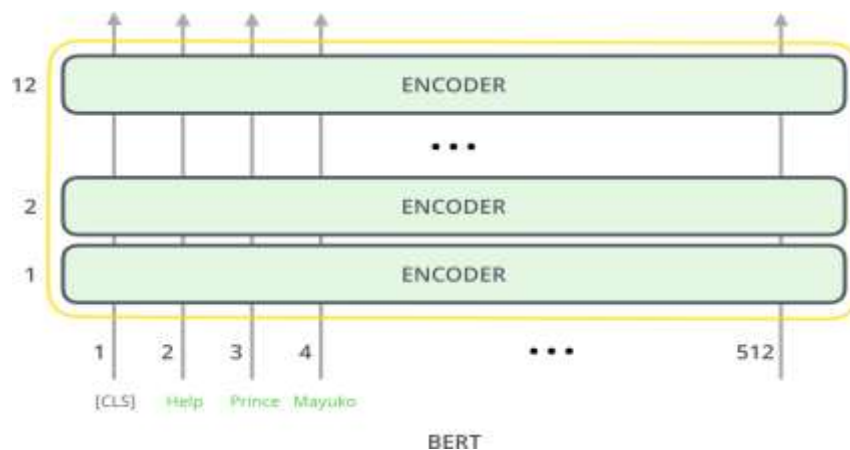
3. Keluaran dari encoder, yaitu kunci vektor dan nilai vektor, masuk ke decoder setelah selesainya setiap proses encoder. Lapisan add & norm encoder dan decoder, yang terdiri dari struktur residual dan lapisan normalisasi, memproses setiap input dan output dari jaringan saraf feed-forward dan lapisan self-attention. Siklus yang terjadi di decoder setara dengan encoder, namun antara lapisan self-attention dan jaringan otak feed-forward ada lapisan pertimbangan yang membantu decoder dengan memusatkan perhatian pada bagian kata yang bersangkutan. Lapisan perhatian-diri decoder hanya dapat fokus pada posisi keluaran sebelumnya. Dekoder menerima keluaran yang sama sebagai pembuat encode dengan terus mengumpukan keluaran dari setiap langkah. Akhirnya, nilai float vektor dihasilkan sebagai keluaran tumpukan decoder. Lapisan softmax dan lapisan tambahan yang terhubung sepenuhnya diperlukan untuk menerjemahkannya ke dalam kata-kata.

Model BERT hanya menggunakan proses sampai dengan encoder dalam arsitekturnya, yang merupakan trafo multi-layer bidirectional seperti implementasi Transformer asli. BERT memiliki dua model ukuran yang dapat digunakan dalam implementasinya: BERTBASE dan BERTLARGE. Encoder atau Blok Transformer datang dalam beberapa lapisan di kedua ukuran model BERT. Encoder di BERTBASE memiliki parameter 110M, 12 kepala self-attention, ukuran tersembunyi 768, dan 12 lapisan. Sebaliknya, BERT LARGE memiliki parameter 340M, 24 lapisan, 16 self- kepala sadar, ukuran tersembunyi 1024, dan 24 lapisan. BERT LARGE membutuhkan empat hari pelatihan dengan 16 TYPUs, sedangkan BERTBASE dilatih dengan empat TYPUs cloud.



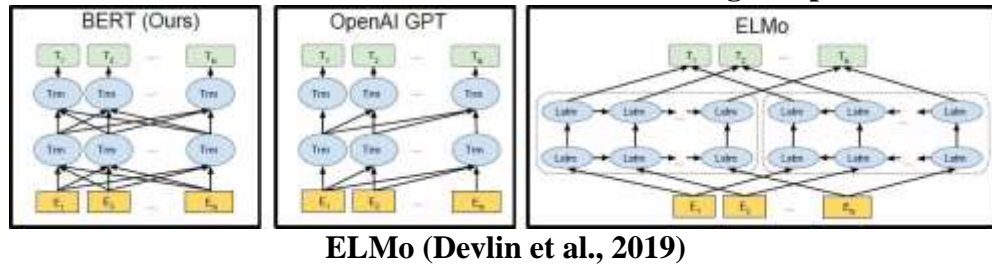
Gambar 2. 6 Perbedaan Ukuran *BERTBASE* dan *BERTLARGE*

Seperti namanya, BERT hanya menggunakan encoder. Dengan demikian, arsitektur BERT sebanding dengan Gambar 2.7. Sebaliknya, model directional memeriksa urutan teks dari kiri ke kanan, kanan ke kiri, atau kombinasi dari kiri ke kanan. dan kanan ke kiri. Hal ini tidak dilakukan oleh BERT. Model satu arah mungkin tidak memiliki kedalaman pemahaman konteks yang sama dengan model bahasa yang dilatih di kedua arah. Gambar 2.8 menunjukkan bagaimana arsitektur BERT menumpuk terhadap OpenAI GPT dan ELMo. Dari tiga model arsitektur, hanya BERT yang memeriksa konteks kiri dan kanan setiap layer secara bersamaan.



Gambar 2. 7 Arsitektur BERT

Gambar 2. 8 Perbedaan Antara Arsitektur BERT Dengan OpenAI GPT dan



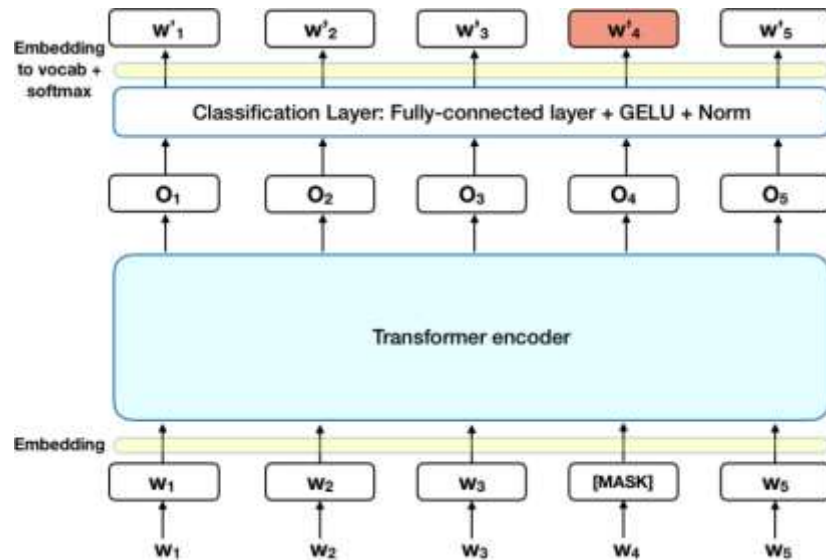
30.000 token kosakata Penyematan WordPiece digunakan oleh BERT. Token klasifikasi khusus [CLS] selalu menjadi token pertama di setiap urutan. BERT dapat dilatih untuk memahami bahasa dan diasah untuk mempelajari tugas-tugas tertentu. Pra-pelatihan dan penyetelan adalah dua fase pelatihan BERT. BERT dilatih untuk memahami dan mempelajari bahasa dan konteksnya selama tahap pertama, yang dikenal sebagai pra-pelatihan. Model Bahasa Bertopeng dan Prediksi Kalimat Berikutnya adalah dua tugas tanpa pengawasan yang dilatih BERT secara bersamaan untuk dipahami.

1. Masked Language Modeling (Masked LM)

Pemodelan Bahasa Bertopeng digunakan untuk menyembunyikan atau menutupi kata-kata acak yang tidak mungkin dalam kalimat. [MASK] token digunakan untuk menggantikan 15% kata dalam setiap urutan kata sebelum dimasukkan ke BERT. Model kemudian akan mencoba memprediksi kata tersebut nilai asli menggunakan konteks yang disediakan oleh kata lain yang tidak ditutup dengan [MASK] dalam urutan kata. Memprediksi kata-kata keluaran secara teknis berarti:

- i) Membutuhkan lapisan klasifikasi di atas output encoder.
- ii) Mengalikan vektor output dengan matriks embedding kemudian mengubahnya menjadi vocabulary dimension.

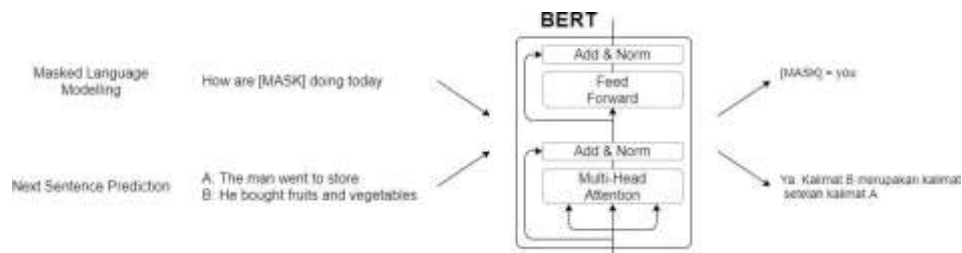
iii) Menghitung probabilitas dari setiap kata di vocabulary dengan softmax.



Gambar 2. 9 Proses Masked Language Modeling

2. Next Sentence Prediction

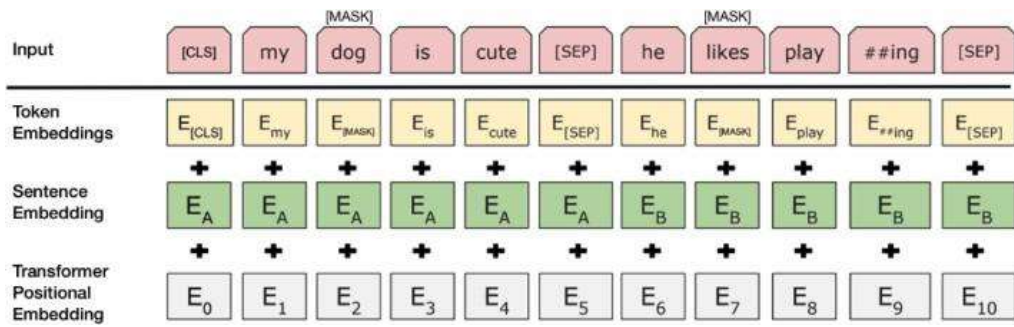
Model tersebut dapat dilatih untuk memprediksi apakah kalimat kedua dalam pasangan adalah kalimat berikutnya dalam dokumen asli atau hanya satu kalimat selama proses pelatihan BERT. Sepasang kalimat, dengan kalimat kedua menjadi kalimat berikutnya dalam dokumen asli, memperhitungkan lima puluh persen input selama pelatihan. Lima puluh persen sisanya adalah kalimat kedua yang diturunkan secara acak dari korpus.



Gambar 2. 10 Proses Pre-training pada BERT

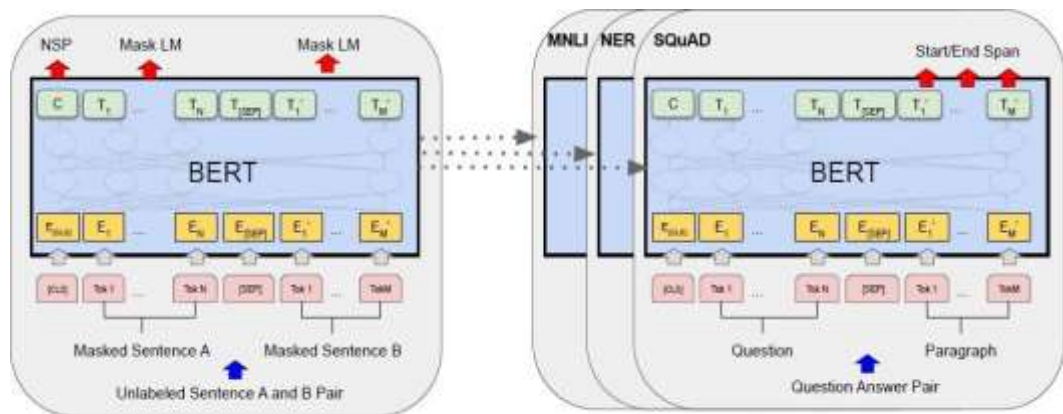
Sebagai representasi input pada BERT, terdapat tiga *embedding layers* yaitu:

1. Lapisan pertama yang dimasukkan token adalah token embeddings, yang merupakan representasi vektor dari setiap token. Representasi vektor berdimensi tinggi dari setiap token dalam input akan dipetakan ke dalamnya. Id yang diturunkan dari kosakata menggantikan setiap token.
2. Ketika ada lebih dari dua kalimat, penyisipan kalimat digunakan untuk membedakannya dengan menambahkan kalimat pertama atau kedua ke setiap token. Hanya ada dua representasi dari lapisan ini: A diberikan untuk token dalam kalimat pertama, dan B adalah diberikan untuk token dalam kalimat kedua.
3. Untuk menyimpan informasi tentang urutan kemunculan kata dalam urutan, penyematan posisi ditambahkan ke setiap token. Transformator mendemonstrasikan konsep dan penerapan penyematan posisi. Selama pra-pelatihan, BERT memeriksa posisi lapisan penyematan.

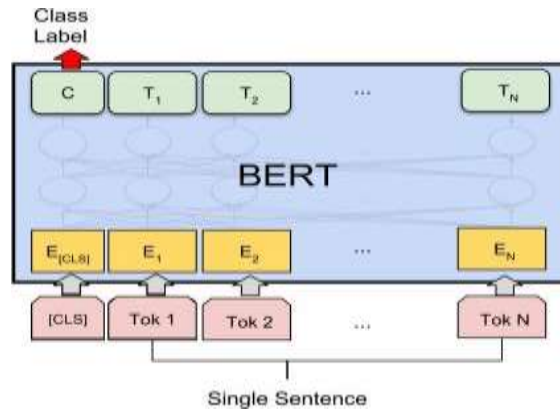


Gambar 2. 11 Representasi Input pada BERT (Devlin et al., 2019)

Selama fase pelatihan, yang dikenal sebagai fine-tuning, classifier harus dilatih dengan sedikit modifikasi pada model BERT untuk melatih model bahasa. Ada saran untuk hyperparameter yang dapat disesuaikan untuk mendapatkan hasil terbaik, seperti Devlin dan timnya. kata rekan-rekannya. Mekanisme self-attention BERT di Transformer memungkinkannya menukar input dan output untuk membuat model untuk berbagai tugas, baik berpasangan atau sebagai kalimat tunggal, membuat fine-tuning menjadi mudah.



Gambar 2. 12 Prosedur Pre-training dan Fine-tuning (Devlin et al., 2019)



Gambar 2. 13 Ilustrasi Fine-tuning pada Tugas dengan Single Sentence (Devlin et al., 2019)

2.8 Penelitian yang Relevan

No	Peneliti	Judul	Metode	Hasil
1	Putri, 2020	Analisis Sentimen Review Film Berbahasa Inggris Dengan Pendekatan Bidirectional Encoder Representations from Transformers	Bidirectional Encoder Representations from Transformers	Hasil akurasi menggunakan BERT adalah 73%, dengan 2000 review dimana 1000 review dengan sentimen positif, dan 1000 review dengan sentimen negatif
2	Abdul et al, 2019	Using BERT for Checking the Polarity of Movie Reviews.	Bidirectional Encoder Representations from Transformers	Hasil evaluation accuracy yang di peroleh mencapai 0.89 dengan loss 0.4856, precission 0.9174 dan recall 0.8812

3	Munakir et al, 2019	Fine-grained Sentiment Classification using BERT.	Bidirectional Encoder Representations from Transformers	Akurasi yang di peroleh oleh penelitian ini adalah 94.0 pada model SST-2 dan 83.9 pada model SST-5 untuk BERTABSE
4	Maharani, 2020	Sentiment Analysis during Jakarta Flood for Emergency Responses and Situational Awareness in Disaster Management using BERT.	Bidirectional Encoder Representations from Transformers	Akurasi yang di peroleh saat tarining dataset adalah 90% dan test dataset adalah 79%
5	Kurniawan et al, 2019	Analisis sentimen opini film Menggunakan Metode Naïve Bayes dan Lexicon Based Features	Naïve Bayes dan Lexicon Based Features	Hasil pengujian menggunakan Naïve-Bayes dengan pembobotan <i>lexicon-based features</i> menghasilkan nilai akurasi 0.8, <i>precision</i> , <i>recall</i> 0.8, dan <i>f-measure</i> 0.8. Sedangkan hasil pengujian menggunakan Naïve-Bayes tanpa pembobotan <i>lexicon-based features</i> menghasilkan nilai akurasi 0.95, <i>precision</i> 1, <i>recall</i> 0.9, dan <i>f-measure</i> 0.9474.

