BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Penelitian tentang analisis sentimen dapat dilakukan dengan berbagai metode. Berikut beberapa penelitian terdahulu yang dijadikan referensi untuk penelitian ini.

Manish Munikar, dkk (Munikar et al., 2019) melakukan *fine-grained sentiment classification* pada *dataset* Rotten Tomatoes. Metode yang digunakan yaitu BERT-*base* dan BERT-*large*. Penelitian dengan model BERT-*base* menghasilkan akurasi sebesar 94% pada SST-2 dan 83.9% pada SST-5. Pada model BERT-*large* menghasilkan akurasi 94,7% pada SST-2 dan 84,2% SST-5.

Kuncahyo Setyo N, dkk (Nugroho et al., 2021) melakukan perbandingan antara model BERT-*base-multilingual* dan IndoBERT-*base* pada data ulasan pengguna 10 aplikasi terbaik di Google Play pada tahun 2020 berbahasa Indonesia dengan jumlah data 10.804. Penelitian menghasilkan akurasi yang cukup baik pada model IndoBERT-*base* dengan pelabelan *lexicon-based* yaitu 84% dengan 25 *epoch*.

Penerapan model BERT-*base* juga dilakukan oleh Saad Abdul Rauf, dkk (Abdul et al., 2019) untuk mengecek polaritas ulasan film menggunakan *dataset* yang diperoleh dari IMDB berbahasa Inggris. Jumlah data yang digunakan adalah 10.000 data. Parameter yang digunakan yaitu *batch size* 32, *learning rate* 2e-5, dan *epoch* 4. Data Penelitian ini menghasilkan nilai akurasi 0,89 dan cenderung positif sentimen.

Penelitian yang dilakukan oleh (Atmaja & Yustanti, 2021) memperoleh nilai akurasi sebesar 99% dengan melakukan analisis sentimen terhadap komentar

pengguna aplikasi Ruang Guru yang ada di Google Play Store menggunakan metode BERT dengan jumlah *dataset* 5.437. Model dilatih dengan percobaan 10 *epoch*. Hasil analisis menyatakan bahwa review pengguna mayoritas positif.

Hiras Parasian Doloksaribu, dkk (Doloksaribu & Samuel, 2022) melakukan komparasi algoritma Naïve Bayes dan SVM untuk menganalisis sentimen ulasan aplikasi PeduliLindungi di Play Store. Pengujian dilakukan sebanyak lima kali dengan jumlah data yang berbeda. Hasil penelitian menunjukkan bahwa SVM dibantu dengan TF-IDF mendapatkan akurasi lebih tinggi dibanding menggunakan Naïve Bayes yaitu sebesar 89,05%.

Penelitian yang dilakukan oleh Cindy Alifia P. (Putri, 2020) menggunakan model *pre-trained* BERT-*base* untuk analisis sentimen terhadap data ulasan film dengan menggunakan *dataset* Cornelledu dari Pabo sebanyak 2000 data. Pada penelitian ini mendapatkan akurasi cukup baik sebesar 73,7%. Akurasi ini sudah terbukti cukup bagus dan cukup jauh dibandingkan dengan penggunaan algoritma Naïve Bayes untuk proses klasifikasi yang hanya memiliki akurasi 48%.

Tabel 2. 1 Tinjauan Pustaka

No	Peneliti	Objek	Metode	Hasil
1.	Manish Munikar, dkk (2019)	Ulasan film dari Rotten Tomatoes pada SST dataset	BERT *base* dan BERT *large*	Klasifikasi sentimen *fine-grained* dengan akurasi rata-rata 90%
2	Kuncahyo Setyo N, dkk (2021)	Ulasan bahasa Indonesia pengguna pada 10 aplikasi terbaik di Google Play	BERT-*base*-*multilingual* dan IndoBERT-*base*	Klasifikasi sentimen 3 kelas, positif, negatif, dan netral mendapatkan akurasi tertinggi pada model IndoBERT-*base* +

No	Peneliti	Objek	Metode	Hasil
				lexicon-based yaitu 84%, dengan 25 *epochs*
3.	Saad Abdul Rauf, dkk (2019)	Data ulasan film berbahasa Inggris pada IMDB	BERT	Klasifikasi sentimen dengan nilai akurasi 0.89 dan polaritas dominan positif
4.	Raden Mas Risqi W.P.K.A dan Wiyli Yustanti (2021)	Data ulasan aplikasi Ruang Guru pada Google Play Store	BERT	Klasifikasi sentimen mendapat akurasi 99% dengan mayoritas positif.
5.	Hiras Parasian D. & Yusran Timur S. (2022)	Data ulasan aplikasi PeduliLindungi di Playstore	SVM dan Naïve Bayes	Klasifikasi sentimen dengan akurasi tertinggi pada metode SVM dengan TF-IDF yaitu 89.05%
6.	Cindy Alifia Putri, dkk (2020	2000 ulasan film dengan menggunakan *dataset* Cornelledu dari Pabo	BERT-*base*	Klasifikasi sentimen 2 kelas, positif dan negatif mendapat akurasi 73%

2.2 Dasar Teori

2.2.1 PeduliLindungi

PeduliLindungi merupakan salah satu bentuk kebijakan pemerintah dalam menangani kasus penyebaran COVID-19 di Indonesia dalam bentuk aplikasi seluler yang rilis perdana pada tanggal 27 Maret 2020. Aplikasi PeduliLindungi dikembangkan oleh Kementerian Komunikasi dan Informatika yang bekerja sama dengan Komite Penanganan COVID-19 dan Pemulihan Ekonomi Nasional (KPCPEN), Kementerian Kesehatan (Kemenkes), dan Kementerian BUMN. PeduliLindungi dikembangkan untuk membantu pemerintah dalam pelacakan

penyebaran COVID-19 agar dapat segera dihentikan penyebarannya. Aplikasi PeduliLindungi dapat diakses melalui website dan dapat diunduh melalui Play Store dan App Store. Tujuan dikembangkannya aplikasi PeduliLindungi adalah untuk memudahkan pemerintah dalam mendeteksi alur penyebaran COVID-19. PeduliLindungi memiliki beberapa fitur, seperti pengawasan/pelacakan, statsitik kasus COVID19, pendaftaran vaksin dan akses sertifikat vaksin, akses surat hasil tes COVID-19, dan lainnya (PeduliLindungi, February 2, 2022).

PeduliLindungi akan mengambil data lokasi pengguna selama mereka bepergian. Sehingga pemerintah dapat melakukan penelusuran riwayat kontak dengan penderita COVID-19. Aplikasi PeduliLindungi memiliki beberapa fitur, yaitu pendaftaran vaksinasi, *scan QR Code*, riwayat perjalanan, paspor digital, zona risiko, statistik, info penting, teledokter, dan *Electronic Health Alert Card* (EHAC). Untuk menggunakan semua fiturnya, pengguna harus melakukan pendaftaran salah satunya menggunakan Nomor Induk Kependudukan (NIK) (PeduliLindungi, February 2, 2022).

2.2.2 Google Play Store

Google Play Store merupakan sebuah *market* digital untuk perangkat Android, Android TV, Wear OS, Chrome OS, dan Web yang dikembangkan oleh Google pada tahun 2008. Produk yang tersedia di Google Play meliputi aplikasi, *games*, musik, buku, dan film. Menurut AppBrain, pada tahun 2019 jumlah aplikasi yang tersedia di Google Play sekitar 2,57 juta aplikasi. Serta lebih dari sepuluh ribu komentar tekstual dari aplikasi yang baru diluncurkan di Google Play (Li et al.,

2015). Pengguna dapat mendapatkan produk dari Google Play secara gratis maupun berbayar.

Pengguna dapat memberikan ulasan dan rating berupa bintang 1 sampai 5 untuk aplikasi dan produk lain yang didistribusikan dalam Google Play. Selain itu, pengembang aplikasi dapat menanggapi ulasan yang diberikan menggunakan Google Play Developer Console. Dengan adanya fitur ini, pengembang dapat terus mengevaluasi produk yang didistribusikan di Google Play.

2.2.3 Analisis Sentimen

Analisis sentimen atau *opinion mining* merupakan salah satu bidang *Natural Language Processing* (NLP) yang menganalisis pendapat, perilaku, penilaian, dan emosi seseorang terkait dengan produk, topik, layanan, organisasi, individu, atau kegiatan lainnya yang diekspresikan ke dalam bentuk teks (Liu, 2020). Analisis sentimen dapat membedakan teks subjektif ke dalam beberapa kategori, yaitu positif, negatif, netral, atau dalam bentuk emosi seperti kegembiraan, kemarahan, atau kesedihan (Ge-Stadnyk et al., 2017). Analisis sentimen dapat diterapkan dalam berbagai tugas, seperti *subjectivity classification*, *sentiment classification*, *opinion spam detection*, *implicit language detection*, dan *aspect extraction*. Pendekatan dalam analisis sentimen ada tiga, yaitu *machine learning* (*unsupervised dan semi-supervised), lexicon based (corpus-based dan dictionary-based)*, dan *hybrid* (gabungan keduanya) (Ligthart et al., 2021).

Analisis sentimen dapat diimplementasikan ke dalam 3 level (Liu, 2012), yaitu level dokumen yang akan mengklasifikasi seluruh dokumen pendapat ke dalam sentimen positif atau negatif. Level kalimat akan menentukan setiap kalimat opini

ke dalam opini positif, negatif, atau netral. Level entitas dan aspek, jika analisis level dokumen dan kalimat tidak bisa menemukan apa yang sebenarnya disukai dan tidak disukai. Level aspek akan melakukan analisis yang lebih halus (*fine-grained*).

2.2.4 *Text Mining*

Text m*ining* merupakan proses untuk mengubah teks tidak terstruktur dalam dokumen dan *database* menjadi data terstruktur (Tan et al., 2000). *Text mining* menganalisis sejumlah besar teks bahasa alami dan mendeteksi pola *lexical* untuk mengekstrak informasi yang berguna. Dalam *text mining* terdapat beberapa tahapan, yaitu do*cument gathering, pre-processing, text transformation, attribute selection, pattern selection*, dan e*valuation* (Mohan, 2016).

1. Pada tahapan *document gathering*, data dapat diperoleh dari *email*, survei, informasi dari media sosial, ulasan, berita, dan sumber lainnya.
2. Karena data yang diperoleh masih mentah, maka harus dilakukan *pre-processing* untuk menghilangkan data yang tidak diperlukan sehingga data siap digunakan untuk langkah selanjutnya. Secara umum, tahapan *pre-processing* yaitu:
    a. *Data cleansing* : menghapus karakter-karakter, tanda baca, dll.
    b. *Case folding* : mengubah semua kata menjadi kapital (*uppercase*) atau tidak kapital (*lowercase*).
    c. *Tokenizing* : dokumen dikenali sebagai *string*. Pada tahap ini string input akan dipotong berdasarkan tiap kata.

d. *Filtering* atau *stopword* : tahap ini akan mengambil kata-kata penting hasil dari token. Dapat dilakukan dengan *stop list* (membuang kata yang kurang penting) atau *word list* (menyimpan kata yang penting).
e. *Stemming* : mengubah kata menjadi kata dasar.
f. *Lemmatization* : mengubah bentuk awal tiap kata lampau dari hasil proses *stemming*.

3. *Text transformation* merupakan proses untuk mendapatkan representasi dokumen yang diharapkan. Terdapat dua pendekatan yang sering digunakan yaitu model *bag of word* dan *vector space model*.
4. *Feature selection* merupakan proses memilih subset dari fitur penting yang digunakan dalam pembuatan model. Karena fitur yang berlebihan dan tidak relevan tidak memberikan informasi tambahan atau dengan kata lain tidak berguna.
5. *Pattern Selection* merupakan proses penggabungan *data mining* dan *text mining.* Teknik *data mining* klasik digunakan dalam database terstruktur yang juga dihasilkan dari tahap sebelumnya.
6. *Evaluation* akan mengukur hasil dari proses yang telah dilakukan. Hasil dapat disimpan untuk keperluan selanjutnya.

2.2.5 *Machine Learning*

Machine learning merupakan cabang algoritma komputasi yang secara luas didefinisikan sebagai kemampuan mesin untuk meniru perilaku manusia dengan belajar dari lingkungan sekitarnya (El Naqa & Murphy, 2015). Algoritma *machine learning* dibangun dengan model yang didasarkan pada data sampel atau data

pelatihan untuk membuat prediksi atau keputusan secara otomatis dengan sedikit atau tanpa input dari manusia dan mengambil tindakan berdasarkan pengamatan masa lalu. *Machine learning* dapat diterapkan dalam berbagai bidang, seperti pada bidang kesehatan, perbankan, teknologi, pendidikan, dan lainnya.

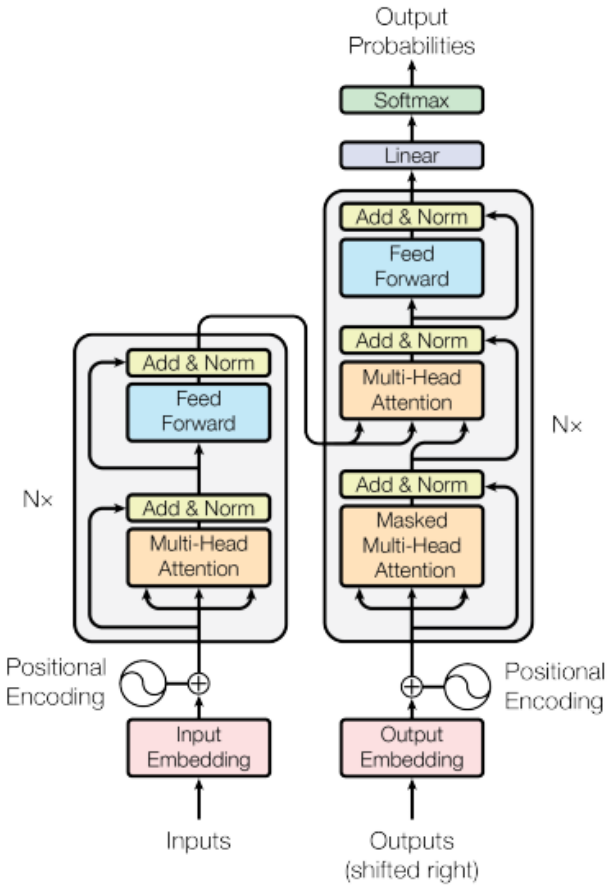
2.2.6 VADER

VADER (*Valence Aware Dictionary for sEntiment Reasoning*) merupakan salah satu *sentiment analyzer* berbasis *lexicon* dan *rule-based*. VADER adalah *open-source* di bawah lisensi MIT yang dikembangkan oleh George Berry, Ewan Klein, dan Pier Paolo. Dengan menggunakan VADER, pelabelan data dapat dilakukan secara otomatis ke dalam tiga kelas, yaitu positif, netral, dan negatif berdasarkan perhitungan teks sentimen sesuai dengan orientasi semantiknya (Illia et al., 2021). Keuntungan menggunakan VADER adalah tersedianya kamus yang berisi nilai pada setiap kata. Kriteria pengelompokan positif, netral, dan negatif adalah jika hasil *compound* lebih dari 0,05 maka dimasukkan kategori positif yang diwakilkan dengan angka 1. Jika hasil *compound* diantara -0,05 dan 0,05 maka termasuk kategori netral yang diwakilkan dengan angka 0. Yang terakhir, jika hasil *compound* di bawah -0,05 maka termasuk kategori negatif yang diwakilkan dengan angka -1 (Tanzilal Mustaqim, 2020).

2.2.7 BERT (*Bidirectional Encoder Representation from Transformers*)

BERT (*Bidirectional Encoder Representation from Transformers*) adalah algoritma *Natural Language Processing* (NLP) terbaru yang dikembangkan oleh Google. Pertama kali diperkenalkan oleh para peneliti Google AI pada tahun 2018.

BERT memberikan hasil yang optimal dalam menyelesaikan berbagai tugas NLP seperti *question answering, natural language inference, classification,* dan *general language understanding evaluation.* BERT memiliki dua varian, yaitu BERT-*base* dan BERT-*large*. BERT-*base* memiliki jumlah L=12, H=768, A=12, total parameter=110M dan BERT-*large* memiliki jumlah L=24, H=1024, A=16, total parameter =340M (Devlin et al., 2019).



Gambar 2. 1 Arsitektur Transformer (Vaswani et al., 2017)

BERT menggunakan arsitektur *deep neural network* yang disebut Transformer yang terlihat pada Gambar 2.1. BERT hanya menerima *input* berupa vektor angka dengan teknik *word embedding*. Proses *embedding* setiap token dalam urutan input direpresentasikan melalui proses ini. Karena arsitektur Transformer

tidak memiliki koneksi berulang, maka posisi dari setiap token dalam urutan input harus secara eksplisit direpresentasikan dengan menambahkan vektor *positional encoding* ke dalam *input embedding.* Urutan *input* beserta *positional encoding*-nya kemudian dimasukkan ke dalam mekanisme *multi-head self-attention*. Mekanisme ini memungkinkan model untuk fokus pada bagian-bagian yang berbeda dalam urutan input pada setiap lapisan dan menangkap ketergantungan jarak jauh antar token. Setelah mekanisme *self-attention, output* dimasukkan ke dalam jaringan saraf *feed-forward*. Jaringan ini menerapkan transformasi *non-linear* untuk setiap posisi secara independen. Untuk meningkatkan pelatihan model dan membantunya konvergen lebih cepat, *residual connections* dan lapisan *normalization* digunakan. *Residual connections* memungkinkan gradien untuk mengalir lebih mudah melalui jaringan, sedangkan lapisan *normalization* membantu untuk menstabilkan distribusi nilai *output*.

Arsitektur Transformer memiliki *stack encoder* dan *decoder*. *Stack encoder* bertanggung jawab untuk meng-*encode* urutan *input*, sementara *stack decoder* bertanggung jawab untuk menghasilkan urutan *output*. Pada *stack decoder*, mekanisme *self-attention* dijadikan *mask* sehingga setiap posisi hanya dapat fokus pada posisi selanjutnya termasuk posisi saat ini. Hal ini diperlukan untuk mencegah model curang dan menghasilkan *output* yang bergantung pada token masa mendatang. Selama proses *decoding, stack decoder* juga memperhatikan *output* dari *stack encoder*. Hal ini memungkinkan model untuk menggunakan informasi dari urutan *input* untuk menghasilkan urutan *output*. Akhirnya, *output* dari *stack decoder* diproyeksikan menjadi vektor probabilitas atas kosakata. Distribusi

probabilitas ini digunakan untuk menghasilkan urutan *output* token per token. (Vaswani et al., 2017).

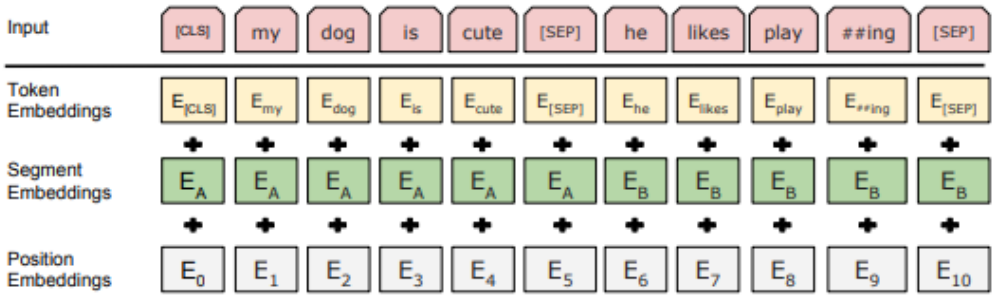
BERT dapat mempelajari hubungan kontekstual antara kata-kata dalam sebuah kalimat yang telah dilatih pada data teks yang besar. Secara khusus, BERT dilatih pada dua tugas yaitu *Masked Language Modeling* (MLM) dan *Next Sentence Prediction* (NSP). Dalam MLM, beberapa kata dalam sebuah kalimat secara acak diganti dengan token [MASK], dan model harus memprediksi kata asli. Tugas ini membantu model memahami makna kata dalam konteks. Sedangkan, dalam NSP, model diberikan dua kalimat dan harus memprediksi apakah kalimat kedua mungkin mengikuti kalimat pertama. Tugas ini membantu model memahami hubungan antar kalimat (Devlin et al., 2018).

Representasi *input* BERT ditampilkan pada Gambar 2.3. Berikut merupakan langkah-langkah tokenisasi dalam BERT (Khalid, 2019) :

1. Tokenisasi : Membagi teks menjadi token-token yang terdiri dari kata-kata. BERT menggunakan tokenisasi WordPiece, yang berarti beberapa token dapat dibagi lagi menjadi sub-token.
2. *Token Embeddings* : BERT menambahkan dua token khusus ke awal dan akhir setiap kalimat, yaitu [CLS] dan [SEP]. Token [CLS] digunakan untuk merepresentasikan kalimat secara keseluruhan yang berada di awal kalimat, sedangkan token [SEP] di akhir kalimat digunakan untuk memisahkan kalimat dalam *input* yang berbeda dari urutan *input*.
3. Konversi Token menjadi ID : Setiap token dalam *input* kemudian dikonversi menjadi ID token yang sesuai menggunakan kamus token yang telah

ditetapkan. Selanjutnya, setiap ID token dikonversi menjadi vektor dengan mengambil nilai *embedding* dari matriks *embedding* kata yang telah dilatih sebelumnya. Matriks *embedding* menggambarkan setiap kata dalam ruang vektor yang terdiri dari banyak dimensi.

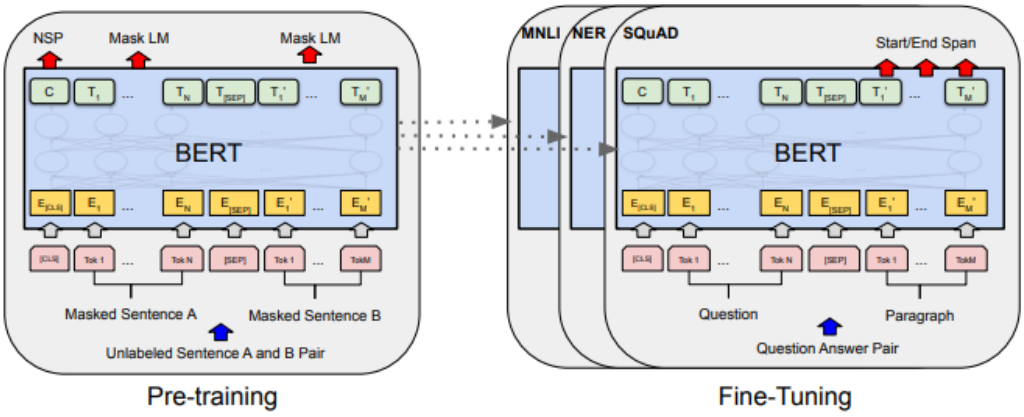
4. *Segment Embeddings* : Jika *input* terdiri dari dua kalimat, setiap token dalam *input* harus ditandai sebagai milik kalimat pertama atau kedua. Ini dilakukan dengan memberikan segmen ID 0 atau 1 ke setiap token, tergantung pada kalimat mana yang mengandung token tersebut.
5. *Position Embedding* : BERT menggunakan *Position embedding* untuk menambahkan informasi posisi absolut ke dalam representasi token. Ini dilakukan dengan menambahkan vektor posisional yang telah ditentukan sebelumnya ke setiap vektor token.



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

BERT menggunakan dua paradigma pelatihan yaitu *pre-training* dan *fine tuning* yang ditunjukkan pada Gambar 2.3. *Pre-training* termasuk *unsupervised learning* karena model dilatih pada *unlabelled dataset* untuk mengekstrak pola. Model ini dilatih pada BooksCorpus (800M kata) dan English Wikipedia (2.5B kata) oleh google. Proses *pre-training* terdiri dari dua tugas, yaitu *Masked*

Language Modeling (MLM) dan *Next Sentence Prediction* (NSP). Sedangkan, selama proses *fine tuning*, model dilatih kembali pada tugas *downstream* dengan data berlabel. *Fine-tuning* melibatkan penyesuaian parameter dari model BERT yang sudah dilatih pada tugas tertentu dengan menggunakan data berlabel untuk mengoptimalkan kinerja model pada tugas tersebut. *Fine-tuning* dilakukan dengan menambahkan lapisan khusus untuk tugas di atas model BERT yang sudah dilatih sebelumnya dan kemudian melatih seluruh model dari awal hingga akhir pada data khusus tugas tersebut. Jumlah parameter di lapisan khusus tugas jauh lebih kecil dari model BERT yang sudah dilatih sebelumnya. Selama *fine-tuning*, model dilatih dengan tingkat pembelajaran yang lebih kecil dibandingkan saat *pre-training*. Hal ini karena model yang sudah dilatih sebelumnya telah belajar fitur umum bahasa dan lapisan khusus tugas perlu mempelajari hanya fitur khusus dari tugas *downstream* (Sun et al., 2020).



Gambar 2. 3 Paradigma Pelatihan BERT (Devlin et al., 2019)

2.2.8 BERT-*Base Multilingual*

BERT *base multilingual* merupakan salah satu variasi model BERT *base* yang telah dilatih sebelumnya menggunakan korpus besar dengan bahasa yang

berbeda termasuk bahasa Indonesia dan bahasa Inggris. Hal ini berarti bahwa model telah belajar untuk merepresentasikan arti kata dan kalimat dalam berbagai bahasa, dan dapat disesuaikan pada tugas yang melibatkan teks dalam salah satu bahasa tersebut. BERT *base multilingual* memiliki 12-*layer*, 768-*hidden*, 12-*heads*, dan 110M parameter. BERT *base multilingual* memiliki dua model yaitu *cased* dan *uncased*. Model *bert-base-multilingual-cased* dilatih pada korpus dengan 104 bahasa yang berbeda dan membedakan antara huruf besar dan kecil dalam teks sehingga dapat memberikan lebih banyak informasi tentang arti kata-kata dalam beberapa bahasa. Sedangkan, *bert-base-multilingual-uncase* dilatih pada korpus dengan 102 bahasa yang berbeda dan sebelum diproses oleh model, teks harus diubah menjadi huruf kecil (Pires et al., 2019).

BERT *base multilingual* dapat digunakan untuk berbagai tugas pemrosesan bahasa alami, seperti klasifikasi teks, *named entity recognition*, *question-answering*, dan masih banyak lagi. Dapat di *fine-tuned* pada tugas tertentu menggunakan data berlabel dan mencapai kinerja canggih pada banyak *benchmarks*. Salah satu manfaat menggunakan BERT *base multilingual* adalah dapat menangani banyak bahasa dalam satu model. Hal ini dapat menghemat banyak waktu dan sumber daya, karena model yang sama dapat digunakan dalam berbagai bahasa (Devlin et al., 2019).