BAB II

LANDASAN TEORI

2.1 Analisis Sentimen

Analisis sentimen mengacu pada proses penggalian polaritas eksplisit atau implisit dari opini yang diekspresikan dalam data tekstual (misalnya, media sosial) termasuk ulasan konsumen *online* [13]. Analisis sentimen telah digunakan untuk pencarian informasi dan permintaan yang menangani kebutuhan di sisi konsumen, sedangkan untuk pemilik bisnis dan pemangku kepentingan lainnya untuk pengambilan keputusan operasional (misalnya, *branding*, tindakan pencegahan/pembalikan)[14].Analisis sentimen tradisional berfokus pada penggalian polaritas opini pada tingkat yang kasar, yang tidak dapat sepenuhnya memenuhi tujuan yang disebutkan di atas. Sentimen biasanya tergantung pada *domain* (contohnya: Lezat menunjukkan sentimen positif dalam domain makanan, yang tidak menunjukkan sentimen apa pun dalam *domain* laptop). Selain itu, konsumen cenderung mengungkapkan sentimen mereka terkait / terkait dengan fitur atau aspek tertentu dari barang atau jasa yang berbeda.

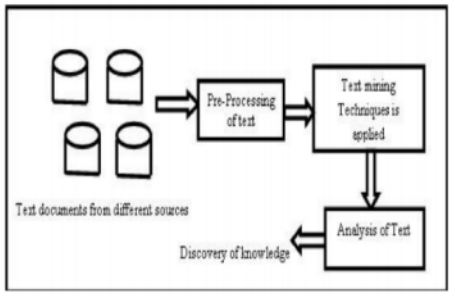
Terdapat beberapa karakteristik sentimen analisis yang dijabarkan pada beberapa poin dibawah ini[8]:

1. Kategorisasi sentimen yaitu membedakan antara kalimat subjektif dan kalimat objektif.
2. Tingkat analisis yang dapat dibagi menjadi 3 tingkatan yaitu *Message level, Sentence Level* dan *Entity and Aspect Level*

3. Pendapat yang membandingkan sesuatu serta pendapat yang hanya sekedar pendapat. Artinya setiap orang dapat memberikan pendapat dengan membandingkan satu hal dengan hal lain atau hanya memberikan pendapatnya saja.
4. Pendapat eksplisit dan pendapat implisit. Pendapat yang diungkapkan secara jujur, tegas dan jelas atau pendapat yang diungkapkan secara tidak jelas.

2.2 *Text Mining*

Text mining adalah proses penambangan informasi berguna dari sebuah dokumen teks yang menggunakan suatu teknik yang mengekstrak informasi yang berasal dari data terstruktur maupun tidak terstruktur dan juga menemukan sebuah pola[15]. Data terstruktur adalah data yang memiliki *field* tetap dalam suatu catatan atau *file* misalnya *relational database*, sedangkan data tidak terstruktur biasanya merujuk pada informasi yang tidak berada dalam *database*[15].



Gambar 2. 1 ***Flowchart Text Mining***

Sumber: [15]

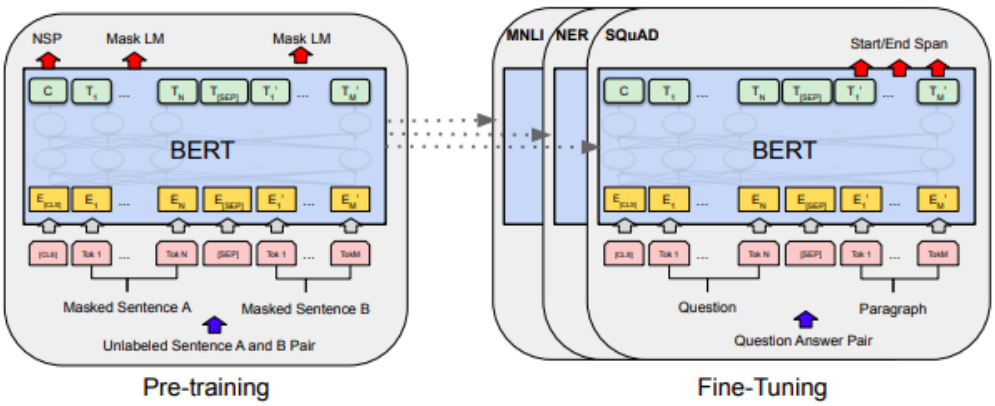
Pada gambar 2.1, proses *text mining* dimulai dengan melakukan *text pre-processing* menggunakan sumber dokumen teks yang tersedia, lalu

mengimplementasikan teknik *text mining* dan terakhir menganalisis teks yang menghasilkan pengetahuan baru. *Text mining* dapat diaplikasi untuk *information retrieval, information extraction,* dan c*ategorization*[15]. *Categorization* melibatkan pengidentifikasian tema-tema utama dokumen dengan memasukan dokumen ke dalam serangkaian topik/*intent* yang telah ditetapkan. Saat mengelompokan dokumen, sebuah program komputer akan sering memperlakukan dokumen sebagai "sekumpulan kata-kata" atau "*bag of word*"[15]. Program ini tidak mencoba untuk memproses informasi sebenarnya sebagai informasi yang terekstraksi melainkan mengkategorikan kata dengan menghitung jumlah kata yang muncul dan dari kalkulasi tersebut mengidentifikasi topik utama dari dokumen teks tersebut[15].*Categorization* seringkali bergantung pada *glossary* yang mana topiknya telah ditetapkan sebelumnya dan hubungan diidentifikasikan dengan mencari istilah general, istilah spesifik, sinonim dan istilah terkait[15].

Text preprocessing merupakan proses paling vital dari *text mining*, proses ini juga disebut membersihkan teks yang mana berfungsi untuk mengidentifikasi dan mengeliminasi kesalahan pada teks [16]. Langkah-langkah yang digunakan untuk *text preprocessing* adalah *tokenization*, menghapus *stopword,* dan melakukan proses pengembalian kata menjadi kata akar/dasar[17].

2.3 BERT

BERT, merupakan singkatan dari *Bidirectional Encoder Representations from Transformers*. Dalam bahasa Indonesia berarti representasi dua arah *encoder*. BERT berguna untuk mengolah representasi dua arah yang berada dalam teks tanpa nama dengan menggabungkan sisi kanan dan kiri pada sebuah konteks dalam segala bagian. Alhasil dengan melakukan sedikit modifikasi atau perubahan pada BERT model yang sudah diolah, dapat menghasilkan atau menyelesaikan berbagai permasalahan yang ada. Sebagai contoh yaitu memberikan jawaban atas pertanyaan yang diberikan dan menyimpulkan sebuah perintah/bahasa tanpa ditambahkan pengaturan lainnya. BERT memiliki keunggulan dalam sisi praktis (sederhana) dan kemampuan observasi yang hebat. Keunggulan ini mengakibatkan BERT dapat memahami 11 bahasa perintah pemrograman, sehingga meningkatkan nilai GLUE sebesar 7,7% menjadi 80.5%, MultiNLI naik sebesar 4.6% menjadi 86.7%, nilai SQuAD v1.1 ke 93.2 dan nilai SQuAD v2.0 Test F1 menjadi 83.1[11].



Gambar 2. 2 *Pre-training* dan *Fine-tuning* pada BERT

Sumber: [11]

BERT memiliki 2 *framework* dengan fungsi yang berbeda: *framework pre-training* dan *framework fine tuning*. Pada gambar 2.2, *framework pre-training* berbagai data tanpa nama diolah (*train*) dengan berbagai macam *task* pre-training sedangkan untuk *framework fine tuning*, model BERT diinisialisasi dengan parameter terlatih yang sudah memiliki label data dari *downstream tasks*. Setiap *downstream*, mempunyai model *fine tune* yang berbeda-beda. Sekalipun mereka diinisialisasi dengan inisiasi *pre-trained* yang sama. Agar penjelasan lebih mudah, berikut contoh bagaimana BERT menjawab pertanyaan. Berbagai macam fitur BERT dipadukan dalam berbagai tugas yang berbeda. Adapun, terdapat perbedaan kecil antara arsitektur *pre-trained* dan arsitektur *downstream*.

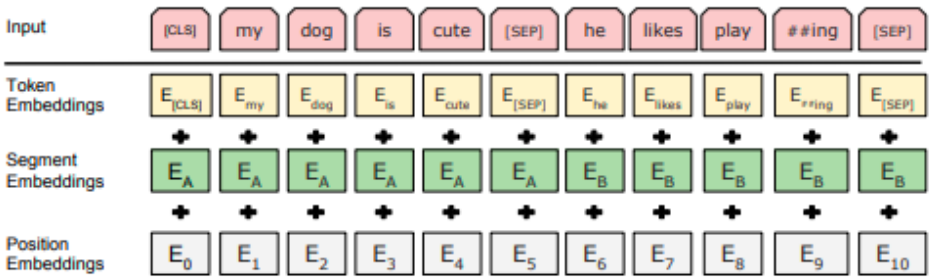
2.3.1 Arsitektur Model

Struktur BERT menggunakan *encoder transformer* dua arah berdasarkan implementasi yang disampaikan [18] dan yang dipublikasikan oleh *library* Tensor2Tensor. Saat ini, penggunaan transformator menjadi sangat umum, sedangkan untuk pengaplikasiannya mirip dengan yang aslinya.

2.3.2 Representasi input/output

Untuk membuat BERT menangani berbagai tugas *down-stream*, representasi input kami dapat secara jelas mewakili satu kalimat dan sepasang kalimat (contoh: Pertanyaan, Jawaban) dalam satu urutan token. Sepanjang pekerjaan ini, sebuah "kalimat" bisa menjadi rentang teks yang bersebelahan secara acak, daripada kalimat linguistik yang sebenarnya. Sebuah "urutan" mengacu pada urutan token input ke BERT, yang

mungkin berupa satu kalimat atau dua kalimat yang dikemas bersama. Kami menggunakan *embeddings* WordPiece [19] dengan 30.000 token kosakata. Token pertama dari setiap urutan selalu merupakan token klasifikasi khusus ([CLS]). Keadaan tersembunyi terakhir yang sesuai dengan token ini digunakan sebagai representasi urutan agregat untuk tugas klasifikasi. Pasangan kalimat digabungkan menjadi satu urutan. Kami membedakan kalimat dengan dua cara. Pertama, kami memisahkannya dengan token khusus ([SEP]). Kedua, kami menambahkan *embedding* yang dipelajari di setiap token yang menunjukkan apakah itu milik kalimat A atau kalimat B. Seperti yang ditunjukkan pada Gambar 1, kami menunjukkan embedding input sebagai E, vektor tersembunyi terakhir dari token khusus [CLS] sebagai C ∈ RH, dan vektor tersembunyi terakhir untuk token input ke-i sebagai Ti R H. Untuk token yang diberikan, representasi inputnya dibangun dengan menjumlahkan token, segmen, dan posisi *embeddings* yang sesuai. Visualisasi dari konstruksi ini dapat dilihat pada Gambar 2.3



Gambar 2. 3 Konstruksi Input atau Output pada BERT

Sumber: [11]

2.3.3 *Pre-trained* BERT

BERT menggunakan dua t*ask unsupervised* pada proses *pre-trained*, yang dideskripsikan pada bagian dibawah ini:

2.3.3.1 Masked LM

Secara intuitif, masuk akal untuk percaya bahwa model dua arah yang dalam benar-benar lebih kuat daripada model kiri-ke-kanan atau penggabungan dangkal dari kiri-kanan dan model kanan-ke-kiri. Sayangnya, model bahasa bersyarat standar hanya dapat dilatih dari kiri-ke-kanan atau kanan-ke-kiri, karena pengkondisian dua arah akan memungkinkan setiap kata untuk "melihat dirinya sendiri" secara tidak langsung, dan model tersebut dapat dengan mudah memprediksi kata target dalam berbagai lapisan. konteks Untuk melatih representasi dua arah yang dalam, kita cukup menutupi beberapa persentase token input secara acak, lalu memprediksi token yang disamarkan tersebut. Kami menyebut prosedur ini sebagai “Masked LM" (MLM), meskipun sering disebut sebagai tugas Cloze dalam literatur. Dalam hal ini, vektor tersembunyi terakhir yang sesuai dengan token mask dimasukkan ke dalam output softmax melalui kosakata, seperti dalam LM standar. Dalam semua eksperimen yang dilakukan, menutupi 15% dari semua token WordPiece di setiap urutan secara acak[20]. Berbeda dengan *denoising auto-encoder*, kami hanya memprediksi kata-kata bertopeng daripada merekonstruksi seluruh input[20].

Meskipun hal ini memungkinkan kita untuk mendapatkan model dua arah *pre-training*, sisi negatifnya adalah kita membuat ketidakcocokan antara *pre-training* dan *fine-tuning*, karena token [MASK] tidak muncul selama *fine-tuning*. Untuk mengurangi ini, kami tidak selalu mengganti kata-kata "mask" dengan token [MASK] yang sebenarnya. Generator data pelatihan memilih 15% dari posisi token secara acak untuk prediksi. Jika token ke-i dipilih, kami mengganti token ke-i dengan (1) token [MASK] 80% dari waktu (2) token acak 10% dari waktu (3) token ke-i yang tidak berubah 10% dari waktu. Kemudian, Ti akan digunakan untuk memprediksi token asli dengan kehilangan entropi silang.

2.3.3.2 Next Sentence Prediction (NSP)

Banyak *downstream tasks* yang penting seperti *Question Answering* (QA) dan *Natural Language Inference* (NLI) yang didasarkan pada pemahaman hubungan antara dua kalimat, yang dimana tidak secara langsung ditangkap oleh pemodelan bahasa. Untuk melatih sebuah model yang memahami hubungan kalimat, perlu melakukan *pre-training* untuk memprediksi kalimat binerisasi berikutnya yang dapat dihasilkan dari korpus monolingual apapun. Khususnya, ketika memilih kalimat A dan B di setiap contoh *pre-training*, 50% dari waktu B adalah kalimat aktual selanjutnya yang mengikuti A (diberi label sebagai IsNext), dan 50% dari waktu itu adalah kalimat acak yang berasal dari

korpus (diberi label sebagai NotNext). Seperti yang kami tunjukkan pada Gambar 1, C digunakan untuk *next sentence prediction* (NSP). Meskipun sederhana, *pre-training* untuk tugas ini sangat bermanfaat bagi QA dan NLI. *task NSP* sangat berkaitan dengan tujuan pembelajaran representasi yang digunakan dalam [21] dan [22]. Namun, dalam pekerjaan sebelumnya, hanya kalimat *embeddings* yang ditransfer ke *downstream task*, dimana BERT mentransfer semua parameter untuk menginisialisasi parameter model *end-task*.

2.3.4 *Fine-tuning* BERT

Fine-tuning pada BERT sangatlah mudah karena menggunakan mekanisme *selfattention* dimana transformer mengizinkan BERT untuk banyaknya model *downstream task*, Aplikasi yang melibatkan pasangan teks, pola yang umum adalah menyandikan pasangan teks secara independen sebelum menerapkan perhatian silang dua arah, seperti A Decomposable Attention Model for Natural Language Inference [23]. BERT menggunakan mekanisme *self-attention* untuk menyatukan dua tahap tersebut, sebagai pengkodean yang menggabungkan pasangan teks dengan *self-attention* secara efektif, termasuk *biderectional cross attention* antara dua kalimat. Setiap *task*, kita hanya perlu memasukan *taskspesific* input dan output ke BERT dan *finetune* semua parameter *end-to-end*. Pada input, kalimat A dan kalimat B dari *pre-training* dianalogikan dengan (1) pasangan kalimat dalam parafrase, (2) pasangan hipotesis-premis dalam

entailment, (3) pasangan bagian-pertanyaan dalam menjawab pertanyaan, dan (4) menurunkan pasangan teks kedua dalam klasifikasi teks atau penandaan urutan. Pada output, representasi token dimasukkan ke dalam lapisan keluaran untuk *token level task*, seperti penandaan urutan atau menjawab pertanyaan, dan representasi [CLS] dimasukkan ke dalam lapisan keluaran untuk klasifikasi, seperti analisis entailment atau sentimen. Dibandingkan dengan *pre-training*, *fine-tuning* relatif tidak mahal.

2.4 IndoNLU

IndoNLU merupakan salah satu produk dari komunitas yang bernama IndoBenchmark. IndoNLU *Benchmark* adalah sekumpulan sumber daya untuk melatih, mengevaluasi, dan menganalisis system pemahaman bahasa alami untuk bahasa Indonesia[12]. IndoNLU mencakup dua belas *tasks*, mulai dari klasifikasi kalimat tunggal hingga pelabelan urutan pasangan kalimat dengan tingkat kompleksitas yang berbeda. Kumpulan data untuk *tasks* terletak pada domain dan *styles* yang berbeda untuk memastikan keragaman *task*.

2.4.1 SmSA

IndoNLU memiliki sebuah s*ingle-sentence classification task* yang berguna untuk melakukan sentimen analisis yang disebut SmSA. SmSA merupakan salah satu *downstream task* khusus IndoBERT yang menggunakan *dataset* dari koleksi komentar dan ulasan di Indonesia yang didapatkan dari berbagai *platform online* [24]. Pada *dataset* tersebut, terdapat tiga sentimen yang dapat diklasifikasikan yaitu positif, negatif, dan netral.

2.4.2 Indobert-BASE

IndoNLU mengikuti pengaturan hiperparameter *pre-training* BERT [12] untuk melatih model IndoBERT-base terlebih dahulu. IndoBERT-base memiliki parameter sebesar 124,5M, 12 *layer*, 12 *head*, 768 *embedding size*, 728 *hidden size*, dan tipe *pre-train* dengan metode kontekstual.

2.5 Tools yang digunakan

2.5.1 *Python*

Python semakin banyak digunakan dalam aplikasi ilmiah yang secara tradisional didominasi oleh R, MATLAB, Stata, SAS, lingkungan komersial atau sumber terbuka lainnya [25]. Kematangan dan stabilitas pustaka numerik fundamental, kualitas dokumentasi, dan ketersediaan distribusi telah berkembang pesat membuat *Python* dapat diakses dan nyaman untuk khalayak luas. Selain itu, matplotlib yang terintegrasi dengan *Python* menyediakan lingkungan penelitian dan pengembangan interaktif dengan visualisasi data yang sesuai untuk sebagian besar pengguna.

2.5.1.1 Pandas DataFrame

Library panda, yang dikembangkan sejak 2008, dimaksudkan untuk menutup celah dalam kekayaan alat analisis data yang tersedia antara *Python*, sistem tujuan umum dan bahasa komputasi ilmiah, dan berbagai *platform* komputasi statistik khusus domain dan bahasa basis data. *Pandas* tidak hanya

bertujuan untuk menyediakan fungsionalitas yang setara tetapi juga mengimplementasikan banyak fitur, seperti penyelarasan data otomatis dan pengindeksan hierarki, yang tidak tersedia dengan cara yang terintegrasi erat di perpustakaan lain atau lingkungan komputasi menurut pengetahuan kita [25].

2.5.1.2 Numpy Array

Array NumPy adalah kumpulan elemen multidimensional yang memiliki tipe data yang sama untuk setiap elemen. Sebuah *array* dimana memiliki elemen dan bentuk. *Array NumPy* benar-benar hanya cara mudah untuk mendeskripsikan satu atau lebih blok memori komputer, sehingga angka diwakili dapat dengan mudah dimanipulasi[25].

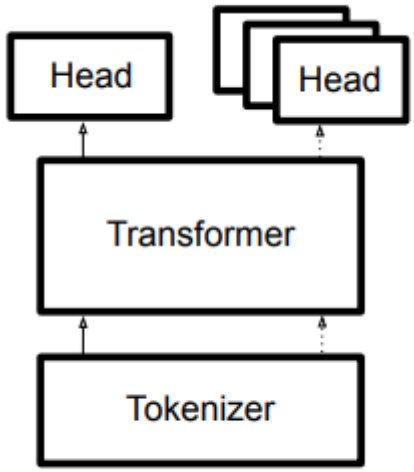
2.5.2 *Scikit-Learn*

Scikit-learn memanfaatkan lingkungan yang kaya ini untuk menyediakan implementasi mutakhir dari banyak algoritma *machine learning* yang terkenal, sambil mempertahankan antarmuka yang mudah digunakan yang terintegrasi erat dengan bahasa *Python*. Ini menjawab kebutuhan yang meningkat akan analisis data *statistic* oleh non-spesialis dalam industri perangkat lunak dan web[26].

2.5.3 Hugging Face Transformer

Hugging face transformer dirancang untuk mencerminkan *pipeline* model *machine learning* NLP standar: memproses data, menerapkan model, dan membuat prediksi. Meskipun *library* menyertakan alat yang

memfasilitasi pelatihan dan pengembangan, dalam laporan teknis ini *hugging face transformer* fokus pada spesifikasi pemodelan inti. Setiap model di *library* sepenuhnya ditentukan oleh tiga *building blocks* yang ditunjukkan pada gambar 2.4. (a) *tokenizer*, yang mengonversi mentah teks ke pengkodean *index encodings*, (b) *transformer*, yang mengubah *index encoding* menjadi *contextual embeddings*, dan (c) *head*, yang menggunakan *contextual embeddings* untuk membuat *task-specific prediction*. Sebagian besar kebutuhan pengguna dapat diatasi dengan ketiga komponen ini[27].



Gambar 2. 4 *Hugging Face Transformer Flow*

Sumber: [27]

2.5.4 *Jupyter Notebook*

Notebook Jupyter adalah *opensource*, alat berbasis browser yang berfungsi sebagai *notebook* lab virtual untuk mendukung alur kerja, kode, data, dan visualisasi yang merinci proses penelitian. Ini adalah mesin dan

dapat dibaca manusia, yang memfasilitasi interoperabilitas dan komunikasi ilmiah. Buku catatan ini dapat tinggal di *repository online* dan menyediakan koneksi ke objek penelitian seperti kumpulan data, kode, dokumen metode, alur kerja, dan publikasi yang ada di tempat lain. *Notebook Jupyter* merupakan salah satu sarana untuk membuat sains semakin terbuka. Relevansi mereka dengan komunitas JCDL terletak pada interaksinya dengan berbagai komponen infrastruktur perpustakaan digital seperti pengindentifikasi digital, mekanisme persistensi, control versi, kumpulan data, dokumentasi, perangkat lunak, dan publikasi[28].

2.5.5 Google Collaboration

Google Collaboratory adalah layanan cloud berbasis *Jupyter Notebook* untuk menyebarkan pendidikan dan penelitian *machine learning*. Ini menyediakan *runtime* yang sepenuhnya dikonfigurasi untuk pembelajaran mendalam dan akses gratis ke GPU yang kuat (Pessoa et al., 2018).

2.6 Populasi dan Sampel

Menurut Sugiyono[29], dalam bukunya menyatakan bahwa "Populasi adalah wilayah generalisasiyang terdiri atas objek atau subjek yang mempunyai kuantitas dan karakteristik tertentu yang ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya.

Menurut Sugiyono[29], dalam bukunya menyatakan bahwa "Sampel adalah sebagian dari jumlah dan karakteristik yang dimiliki oleh populasi". Setiap

subjek diupayakan mempunyai peluang untuk menjadi sampel agar dalam pengambilan sampel didapatkan sampel yang dapat mewakili atau representatif.

Menurut Sugiyono[29], "Teknik *sampling* merupakan teknik pengambilan sampel". Untuk menentukan sampel yang akan digunakan pada suatu penelitian, terdapat berbagai macam teknik. Salah satunya adalah *probability sampling*, yaitu teknik *sampling* yang memberikan peluang yang sama bagi setiap unsur pada populasi untuk dipilih menjadi anggota sampel. Pada *probability sampling* terdapat teknik, *proportionate stratified random sampling*. Teknik ini digunakan bila populasi mempunyai anggota atau unsur yang tidak homogen. Berikut merupakan rumus *proportionate stratified random sampling*:

Proportionate Stratified Random Sampling Formula: *nh = (Nh / N) * n*

$$\boldsymbol{Propotionate\ Stratified\ Random\ Sampling\ Formula}$$

$$\boldsymbol{nh = (Nh / N) * n}$$

Rumus 1 Proportionate Stratified Random Sampling

nh*= Sample size for hth stratum*

Nh*= Population size for hth stratum*

N *= Size of entire population*

n *= Size of entire sample*

2.7 Penelitian Pendahulu

Tabel 2. 1 Penelitian Pendahulu

Nama Peneliti	Judul Penelitian	Nama Jurnal	Hasil Penelitian	Hasil yang Diambil
Handani	*Sentiment*	*Journal of*	Hasil klasifikasi	Klasifikasi

et al., 2019	*Analysis for Go-Jek on Google Play Store*	*Physics: Conference Series*	analisis sentimen dari ulasan pengguna Gojek pada Play Store.	analisis sentimen pada ulasan Gojek pada Play Store.
Siahaan et al., 2020	Apakah *Youtuber* Indonesia Kena *Bully* Netizen?	Jurnal Sistem Informasi	Hasil klasifikasi analisis sentimen dari banyaknya komentar *cyberbullying* yang diterima oleh *Youtuber* Indonesia di Instagram	Klasifikasi analisis sentimen yang ada pada komentar *cyberbullying* yang diterima *Youtuber* Indonesia pada akun Instagram.
Destitus et al., 2020	*Support Vector Machine* VS *Information Gain :*Analisis Sentimen *Cyberbullying* di Twitter Indonesia	Jurnal Sistem Informasi	Hasil identifikasi *tweet cyberbullying* menggunakan metode *Support Vector Machine* dan *Information Gain*	Cara perhitungan *accuracy, recall, f1-measure*,dan *precision* pada penggunaan *Support Vector Machine* dan *Information Gain*.
Willie et al., 2020	*IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding*	*International Joint Conference on Natural Language Processing* (*IJCNLP*). *Journal of Physics: Conference Series*	Model IndoBERT memiliki performa yang lebih baik dibandingkan dengan model scratch, XLM, fasttext maupun m-bert untuk melakukan *task sentiment analisis* (SmSA) dan IndoBERT base merupakan	Penggunaan *library* SmSA dengan model Indobert- base sebagai model pre-trained.

			model yang membutuhkan durasi training tersingkat diantara ketiga model tersebut.	
Putri et al., 2020	Analisis Sentimen *Review* film Berbahasa Inggris Dengan Pendekatan *Bidirectional Encoder Representations from Transformers*	Jurnal Teknik Informatika dan Sistem Informasi	Penggunaan model *pre-trained* BERT base menunjukkan hasil yang lebih baik dibandingkan dengan penggunaan algoritma *naïve bayes* pada analisis sentimen.	Implementasi BERT dan penggunaan model *pre-trained* BERT base

Berikut merupakan penjelasan pada table 2.1 penelitian terdahulu, yaitu :

1. Pada penelitian terdahulu yang berjudul *Sentiment analysis for Gojek on Google Play Store* menjadi salah satu referensi dikarenakan penelitian ini mengambil review di playstore yang dimana pengguna terbesar Gojek itu berasal dari Play Store dan banyak pengguna Gojek memberikan *review* di *platform* tersebut. Dibandingkan dengan kompetitor lainnya, Gojek tetap diposisi pertama *platform* digital berbasis transportasi *online*. Hasil penelitian terdahulu ini menunjukkan bahwa analisis sentimen dibutuhkan supaya Gojek bisa belajar dan mengetahui aspek apa saja yang dibutuhkan oleh pelanggannya(*customer*) sehingga dengan adanya analisis sentimen bisa menjadi bahan evaluasi dalam peningkatan layanannya. Pada penelitian ini terdapat perbedaan yaitu penggunaan algoritma BERT dan

polaritas berbeda pada algoritma yang dimana penelitian ini menggunakan BERT yang kategori utamanya yaitu polaritasnya positif, negatif dan netral[10].

2. Pada penelitian terdahulu yang berjudul "Apakah Youtuber Indonesia Kena *Bully* Netizen?" menjadi salah satu referensi dikarenakan penelitian tersebut melakukan analisis sentimen untuk mengetahui banyaknya unsur komentar *cyberbullying* padaYoutuber Indonesia di Instagram. Hasil penelitian terdahulu penggunaan *Support Vector Machine* mendapatkan akurasi sebesar 81,2%, bisa disimpulkan bahwa dalam sebuah komentar Instagram *Youtuber* Indonesia terdapat 49,524% komentar yang mengandung unsur *cyberbullying*, yang dikatakan bahwa *Youtuber* Indonesia tidak melulu dirundung oleh masyarakat Indonesia. Hal positif yang mereka lakukan menjadi contoh bagi masyarakat Indonesia. Pada penelitian ini terdapat perbedaan pada penggunaan algoritma yang dipakai yaitu BERT dan juga penggunaan 3 polaritas seperti positif, negatif, dan netral[30].
3. Pada penelitian terdahulu yang berjudul "*Support Vector Machine* VS *Information Gain* : Analisis Sentimen *Cyberbullying* di Twitter Indonesia" menjadi salah satu referensi dikarena penelitian ini melakukan perhitungan *accuracy*, *precision*, *recall*, dan *f-measure* untuk mengetahui performa masing-masing model *Support Vector Machine* dan *Information Gain*. Hasil pada penelitian terdahulu penggunaan *Support Vector Machine* mendapatkan *accuracy* yang cukup tinggi yaitu 80%, *precision* 75,1%,

recall 96% dan *f-measure* 85% sedangkan pada *Information Gain* didapatkan *accuracy* 86%, *precision* 81%, *recall* 95%, dan *f-measure* 87% sehingga hasil indentifikasi *tweet cyberbullying* dengan kedua metode memiliki hasil yang cukup maksimal. Hasil yang digunakan pada penelitian ini penggunaan perhitungan *accuracy*, *precision*, *recall* dan *f-measure* untuk melihat performa model pada BERT yang selanjutkan akan di implementasikan ke *dataset* baru[31].

4. Pada penelitian terdahulu yang berjudul *IndoNLU: Benchmark and Resources for Evaluating Indonesian Natural Language Understanding* menjadi salah satu referensi dikarenakan hasil dari performa model untuk *sentiment analysis* lebih baik menggunakan model IndoBERT karena dibandingkan dengan model lainnya performa IndoBERT paling baik dilihat dari penelitian ini m-bert didapatkan performa sebesar 84,14, XLM sebesar 86,33, scratch sebesar 67,35, fasttext sebesar 76,92 dan sedangkan IndoBERT mendapatkan 87,73. Pada penelitian ini, model yang digunakan adalah IndoBERT base yang walaupun performa tidak sebaik model lainnya tetapi dapat dilihat dari sisi *resource* yang dimiliki dan juga memiliki *duration* yang tergolong singkat dibanding model lainnya tetapi sudah memiliki *performance* yang cukup memuaskan[12].
5. Pada penelitian terdahulu yang berjudul Analisis Sentimen *Review* film Berbahasa Inggris Dengan Pendekatan *Bidirectional Encoder Representations from Transformers* menjadi salah satu referensi dikarenakan penelitian tersebut menunjukkan model *pre-trained* dari

BERT-base terbukti dapat digunakan. Bert-base terbukti dapat digunakan tidak hanya ditunjukkan pada kalimat-kalimat pendek untuk diklasifikasikan. Walaupun hanya dilakukan pengambilan 128 karakter akhir dari total karakter yang ada pada dokumen, BERT-base masih dapat melakukan training dengan baik sehingga didapatkan 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 nilai performa 48%[32].