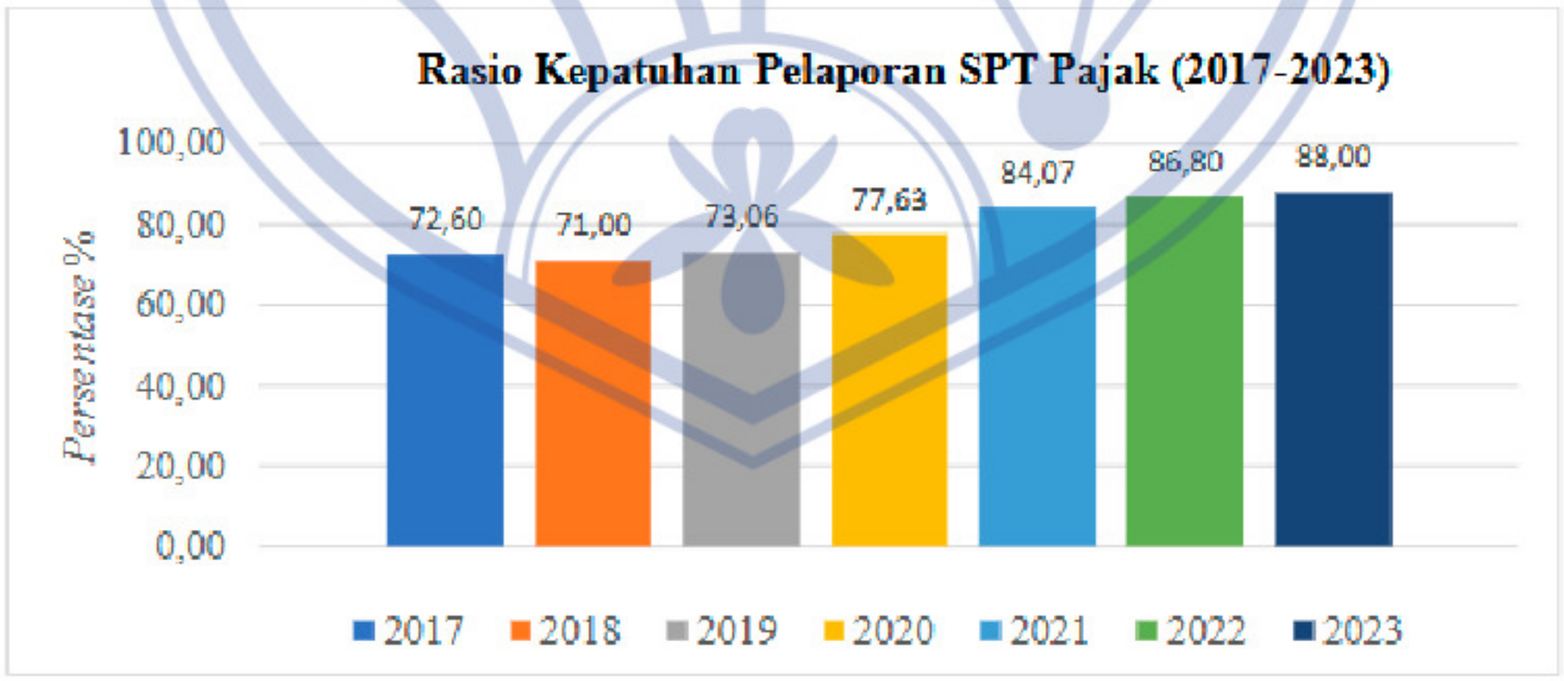
BAB II
KAJIAN LITERATUR

2.1 Sistem Coretax

Coretax merupakan sistem administrasi perpajakan yang dikembangkan oleh Direktorat Jenderal Pajak (DJP) sebagai bagian dari program modernisasi sistem perpajakan di Indonesia. Sistem ini dirancang untuk meningkatkan efisiensi pengelolaan data perpajakan serta memberikan kemudahan bagi wajib pajak dalam melakukan berbagai aktivitas administrasi perpajakan seperti pembuatan faktur pajak, pelaporan Surat Pemberitahuan (SPT), serta pengelolaan transaksi perpajakan secara digital. Implementasi sistem Coretax diharapkan mampu meningkatkan kualitas pelayanan perpajakan serta memperkuat integrasi sistem administrasi perpajakan secara nasional [2]. Modernisasi sistem perpajakan melalui pemanfaatan teknologi informasi merupakan langkah strategis untuk meningkatkan efektivitas pengelolaan administrasi pajak serta mendorong kepatuhan wajib pajak. Salah satu inovasi yang dikembangkan oleh Direktorat Jenderal Pajak adalah *Core Tax Administration System* (Coretax), yaitu sistem terpadu yang mengintegrasikan berbagai layanan perpajakan seperti pembuatan faktur pajak, pelaporan Surat Pemberitahuan (SPT), serta pengelolaan data perpajakan secara digital sehingga proses administrasi perpajakan dapat dilakukan secara lebih efisien, terintegrasi, dan transparan [10]. Untuk melihat perkembangan tingkat kepatuhan wajib pajak dapat dilihat dari data pelaporan SPT dalam beberapa tahun terakhir, dapat dilihat pada Gambar 2.1



Gambar 2. 1 Persentase Kepatuhan Wajib Pajak 2017 – 2023 [3]

Berdasarkan Gambar 2.1 tingkat kepatuhan wajib pajak dalam pelaporan SPT menunjukkan kecenderungan meningkat dari tahun ke tahun. Meskipun sempat mengalami

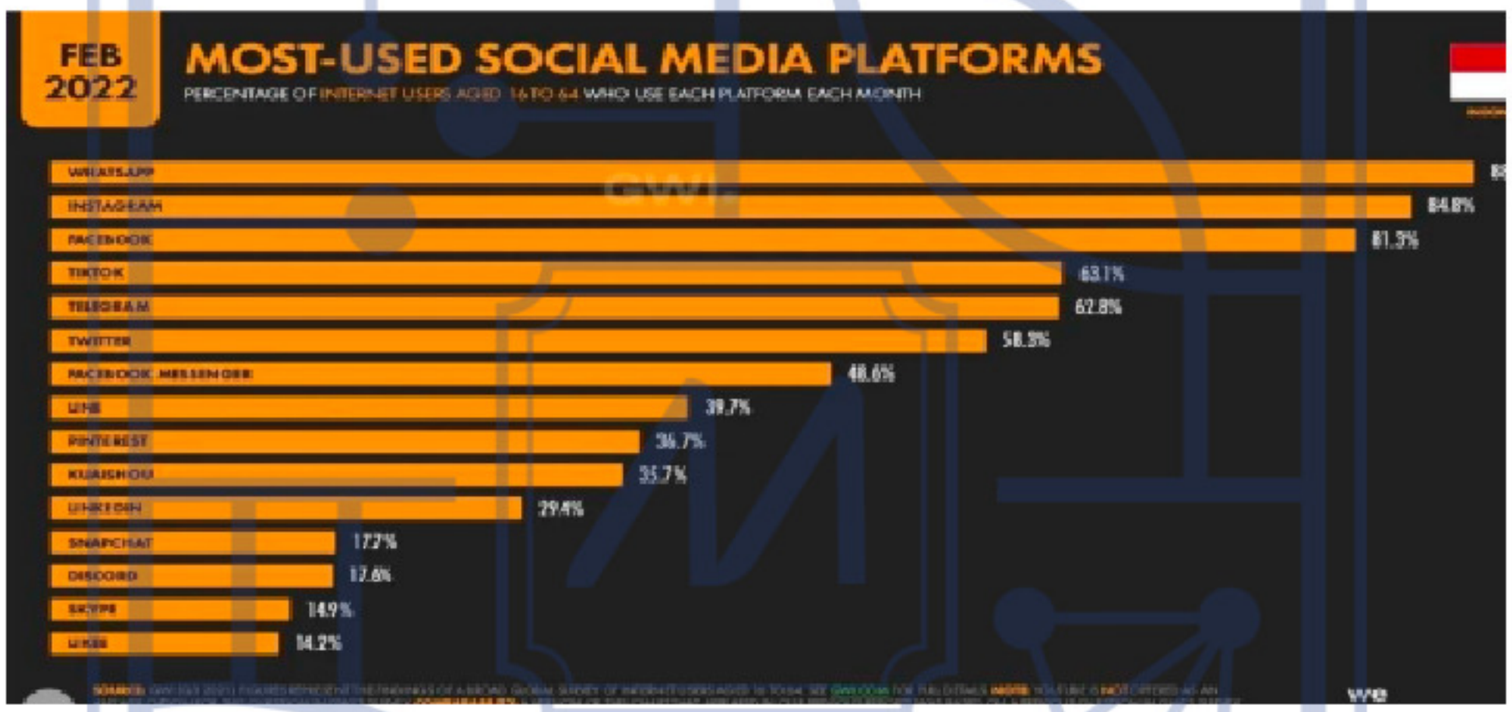
penurunan pada tahun 2018, tingkat kepatuhan kembali meningkat hingga mencapai sekitar 88% pada tahun 2023 . Hal ini menunjukkan adanya perbaikan dalam kesadaran wajib pajak, namun capaian tersebut masih belum sepenuhnya memenuhi target yang diharapkan. Oleh karena itu, diperlukan sistem yang mampu meningkatkan efisiensi, transparansi, serta pengawasan perpajakan, salah satunya melalui implementasi Coretax [3]. Melalui sistem Coretax, pemerintah dapat mengintegrasikan seluruh proses administrasi perpajakan, mulai dari pendaftaran, pelaporan, pembayaran, hingga pengawasan dalam satu sistem terpadu. Hal ini memungkinkan pengawasan dilakukan secara lebih efektif serta mendukung peningkatan kepatuhan wajib pajak. Selain itu, penggunaan teknologi seperti integrasi data dan otomatisasi proses juga membantu dalam mendeteksi ketidaksesuaian data serta potensi ketidakpatuhan secara lebih cepat. Namun demikian, implementasi sistem Coretax juga menghadapi berbagai tantangan teknis seperti gangguan sistem, keterlambatan akses layanan, serta kesulitan dalam pengelolaan data yang dapat mempengaruhi pengalaman pengguna. Kondisi ini memunculkan berbagai tanggapan serta opini dari masyarakat yang banyak disampaikan melalui media sosial [2]. Oleh karena itu, analisis terhadap opini masyarakat yang disampaikan melalui media sosial menjadi penting untuk dilakukan guna memahami persepsi publik terhadap implementasi sistem Coretax.

2.2 Media Sosial TikTok

Media sosial merupakan platform berbasis internet yang memungkinkan pengguna untuk berinteraksi, berbagi informasi, serta membangun jaringan komunikasi secara daring. Perkembangan media sosial pada era digital telah mengubah cara masyarakat dalam menyampaikan pendapat dan berpartisipasi dalam pertukaran informasi di ruang publik. Melalui berbagai fitur yang tersedia, pengguna dapat dengan mudah menyampaikan pandangan, kritik, maupun tanggapan terhadap berbagai isu yang sedang berkembang secara terbuka dan interaktif [11]. Media sosial juga menjadi salah satu sarana bagi masyarakat untuk menyampaikan opini terhadap berbagai isu publik, termasuk kebijakan pemerintah, layanan digital, maupun fenomena sosial lainnya. Platform seperti Twitter (X) sering dimanfaatkan oleh pengguna untuk menyampaikan pendapat secara langsung mengenai suatu topik tertentu. Analisis terhadap opini yang disampaikan melalui media sosial dapat memberikan gambaran mengenai persepsi masyarakat terhadap suatu fenomena atau kebijakan tertentu [12].

Selain Twitter (X), terdapat berbagai platform media sosial lain yang juga banyak digunakan masyarakat untuk mengekspresikan opini, salah satunya adalah TikTok. TikTok

merupakan salah satu platform media sosial berbasis video pendek yang memungkinkan pengguna untuk membuat, mengedit, serta membagikan konten video secara kreatif melalui perangkat digital. Platform ini mengalami perkembangan yang pesat dan memiliki jumlah pengguna yang sangat besar di berbagai negara. Selain itu, TikTok juga menyediakan berbagai fitur interaksi seperti *like*, komentar, dan *share* yang memungkinkan pengguna untuk berpartisipasi secara aktif dalam menanggapi konten yang diunggah oleh pengguna lainnya [13]. Perkembangan penggunaan TikTok sebagai media sosial yang populer dapat dilihat dari tingkat penggunaannya di Indonesia, sebagaimana ditunjukkan pada gambar berikut. Untuk mengetahui posisi TikTok dibandingkan dengan platform media sosial lainnya di Indonesia, data tingkat penggunaan media sosial disajikan pada Gambar 2.2.



Gambar 2. 2 Data Media Sosial yang Paling Banyak Diakses di Indonesia Tahun 2022 [14]

Gambar 2.2 menunjukkan bahwa TikTok merupakan salah satu media sosial yang paling banyak digunakan di Indonesia dengan persentase sebesar 63,1% dari total pengguna internet. Data ini menunjukkan bahwa TikTok telah menjadi platform yang dominan dalam ekosistem media sosial serta memiliki tingkat adopsi yang tinggi di kalangan masyarakat. Tingginya jumlah pengguna ini menjadikan TikTok sebagai sumber data yang relevan dalam penelitian yang berkaitan dengan opini publik dan analisis sentimen [14]. Karakteristik interaksi pada TikTok memiliki perbedaan dibandingkan dengan beberapa platform media sosial lainnya. Pada TikTok, pengguna dapat memberikan tanggapan secara langsung melalui kolom komentar pada setiap konten video yang dipublikasikan. Interaksi tersebut memungkinkan terjadinya diskusi terbuka antar pengguna mengenai berbagai topik yang sedang berkembang. Oleh karena itu, komentar pada platform TikTok dapat menjadi sumber data yang potensial dalam penelitian yang berkaitan dengan opini publik maupun analisis sentimen [15]. Komentar yang disampaikan oleh pengguna pada TikTok sering kali memuat tanggapan terhadap berbagai isu sosial, pengalaman pengguna terhadap suatu layanan, serta pendapat terhadap kebijakan atau sistem tertentu. Informasi yang terdapat dalam komentar

tersebut dapat dimanfaatkan sebagai sumber data untuk menganalisis kecenderungan opini masyarakat terhadap suatu topik. Dalam penelitian ini, komentar pengguna pada platform TikTok digunakan sebagai sumber data untuk melakukan analisis sentimen terhadap implementasi sistem Coretax.

2.3 Karakteristik Data Teks Media Sosial

Bahasa yang digunakan oleh pengguna media sosial cenderung bersifat tidak baku, informal, serta mengandung berbagai variasi penulisan yang tidak mengikuti kaidah bahasa Indonesia yang benar [16], [17]. Pengguna sering menggunakan singkatan, kata tidak lengkap, serta bentuk bahasa gaul (*slang*) yang berkembang sesuai dengan kebiasaan komunikasi di dunia digital [16], [18]. Selain itu, teks pada media sosial juga sering mengandung elemen tambahan seperti emotikon, emoji, tanda baca berulang, serta penggunaan huruf yang tidak konsisten, seperti kombinasi huruf besar dan kecil dalam satu kalimat [17], [19]. Variasi tersebut dapat menimbulkan *noise* pada data, sehingga menyulitkan model dalam memahami makna yang sebenarnya dari teks yang dianalisis [18].

Karakteristik lain yang umum ditemukan adalah adanya kesalahan penulisan (*typo*), penggunaan kata yang berulang untuk menekankan emosi, serta struktur kalimat yang tidak lengkap [19]. Kondisi ini membuat data teks media sosial menjadi tidak terstruktur dan membutuhkan proses pengolahan awal sebelum digunakan dalam analisis lebih lanjut [16]. Oleh karena itu, tahap preprocessing menjadi sangat penting dalam penelitian berbasis data media sosial. Proses ini bertujuan untuk membersihkan dan menormalkan teks sehingga dapat meningkatkan kualitas data serta membantu model dalam memahami konteks kalimat secara lebih akurat [17], [20]. Dalam penelitian ini, karakteristik teks dari komentar pengguna TikTok yang bersifat tidak baku menjadi pertimbangan utama dalam penerapan tahapan *preprocessing* [18].

2.4 Crawling Data

Crawling data merupakan teknik pengumpulan data secara otomatis dari berbagai sumber di internet dengan menggunakan program komputer yang disebut *web crawler* atau *web spider*. *Web crawler* merupakan perangkat lunak yang secara sistematis menelusuri halaman web dan mengunduh dokumen atau informasi yang tersedia secara otomatis untuk keperluan pengindeksan maupun analisis data. Teknik ini memungkinkan proses pengumpulan data dilakukan secara efisien dan terstruktur sehingga sering digunakan dalam penelitian berbasis data digital [21]. Dalam penelitian yang memanfaatkan data dari internet,

teknik *crawling* sering digunakan untuk memperoleh data dalam jumlah besar dari berbagai situs web maupun platform digital. Proses ini memungkinkan peneliti mengakses dan mengumpulkan informasi yang tersedia secara publik sehingga dapat dimanfaatkan dalam berbagai bidang penelitian seperti data *mining*, *text mining*, maupun *Natural Language Processing* (NLP). Dengan memanfaatkan teknik *crawling*, proses pengumpulan data yang sebelumnya dilakukan secara manual dapat dilakukan secara otomatis dan lebih efisien[22]. Salah satu metode yang umum digunakan dalam proses *crawling* data adalah *web scraping*. *Web scraping* merupakan teknik ekstraksi data dari halaman web dengan mengambil elemen-elemen tertentu dari struktur HTML menggunakan perangkat lunak atau bahasa pemrograman tertentu. Teknik ini memungkinkan pengambilan berbagai jenis data seperti teks, komentar pengguna, maupun informasi lainnya yang tersedia pada halaman web sehingga data tersebut dapat diolah lebih lanjut dalam bentuk dataset terstruktur [23].

Selain menggunakan teknik *web scraping* secara umum, penelitian ini juga memanfaatkan platform berbasis cloud seperti Apify dalam proses pengumpulan data. Apify merupakan platform otomatisasi web yang menyediakan berbagai *tools* untuk melakukan *crawling* dan *scraping* data dari berbagai sumber internet secara terstruktur dan efisien. Platform ini memungkinkan pengguna untuk menjalankan proses ekstraksi data tanpa perlu membangun *crawler* dari awal, karena telah tersedia berbagai *actor* yang dapat digunakan sesuai kebutuhan, termasuk untuk platform media sosial seperti TikTok [24]. Apify bekerja dengan menjalankan skrip otomatis yang disebut *actor*, yaitu program yang dirancang untuk mengekstraksi data dari halaman web tertentu secara sistematis. Salah satu keunggulan Apify adalah kemampuannya dalam menangani proses *scraping* secara skalabel, termasuk pengelolaan *request*, rotasi *proxy*, serta penyimpanan data dalam berbagai format seperti JSON atau CSV [25]. Selain itu, teknik web *scraping* memungkinkan proses ekstraksi data dari halaman web dilakukan secara otomatis menggunakan perangkat lunak, sehingga dapat mengumpulkan data dalam jumlah besar secara lebih efisien dibandingkan metode manual [25]. Dalam konteks penelitian berbasis media sosial, teknik *crawling* data banyak digunakan untuk mengumpulkan berbagai bentuk interaksi pengguna seperti komentar, ulasan, maupun tanggapan terhadap suatu isu tertentu. Data yang diperoleh dari proses *crawling* kemudian dapat digunakan untuk menganalisis opini publik serta kecenderungan sentimen masyarakat terhadap suatu topik yang sedang berkembang di media sosial [26].

2.5 Natural Language Processing

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan yang berfokus pada kemampuan komputer untuk memahami, memproses, dan menganalisis bahasa manusia dalam bentuk teks maupun suara. NLP menggabungkan konsep dari linguistik, ilmu komputer, dan pembelajaran mesin untuk memungkinkan sistem komputer menafsirkan makna dari bahasa alami yang digunakan manusia dalam komunikasi sehari-hari [27]. Perkembangan NLP dalam beberapa tahun terakhir mengalami kemajuan yang signifikan terutama dengan munculnya model berbasis *deep learning* dan arsitektur *transformer*. Model-model tersebut mampu memahami konteks bahasa secara lebih baik dibandingkan metode tradisional berbasis statistik atau *rule-based*. Dengan memanfaatkan teknik pembelajaran mendalam, sistem NLP dapat melakukan berbagai tugas pemrosesan teks seperti klasifikasi teks, analisis sentimen, ekstraksi informasi, penerjemahan bahasa, serta sistem tanya jawab secara otomatis [28]. Dalam konteks penelitian berbasis media sosial, NLP banyak digunakan untuk mengolah data teks yang berasal dari berbagai platform digital seperti komentar pengguna, ulasan produk, maupun unggahan pada media sosial. Data teks pada media sosial umumnya bersifat tidak terstruktur dan memiliki variasi bahasa yang cukup kompleks, misalnya penggunaan Bahasa informal, singkatan, maupun kata tida k baku. Oleh karena itu, teknik NLP digunakan untuk mengubah teks tersebut menjadi repr esentasi yang dapat dipahami oleh sistem komputer sehingga memungkinkan proses analisis lebih lanjut dilakukan secara otomatis [29].

Salah satu penerapan penting NLP adalah dalam analisis sentimen, yaitu proses mengidentifikasi kecenderungan opini atau sikap yang terkandung dalam suatu teks. Melalui pendekatan ini, teks dapat diklasifikasikan ke dalam kategori sentimen tertentu seperti positif atau negatif. Analisis sentimen banyak digunakan untuk memahami opini publik terhadap suatu produk, layanan, maupun kebijakan berdasarkan data yang diperoleh dari media sosial [30]. Dalam penelitian ini, teknik NLP digunakan untuk mengolah data komentar pengguna pada platform TikTok yang berkaitan dengan sistem Coretax. Data komentar yang telah diperoleh melalui proses *crawling* kemudian diproses melalui tahap *preprocessing*, pelabelan data, dan pembagian dataset sebelum digunakan dalam proses klasifikasi sentimen menggunakan model IndoBERT. Dengan memanfaatkan pendekatan NLP, penelitian ini diharapkan mampu mengidentifikasi kecenderungan sentimen pengguna terhadap implementasi sistem Coretax berdasarkan komentar yang terdapat pada media sosial TikTok.

2.6 Analisis Sentimen

Analisis sentimen merupakan salah satu teknik dalam *text mining* dan *Natural Lang uage Processing (*NLP) yang digunakan untuk mengidentifikasi serta mengklasifikasikan opini, emosi, atau sikap yang terkandung dalam suatu teks. Teknik ini bertujuan untuk memahami persepsi atau pandangan seseorang terhadap suatu objek, layanan, maupun topik tertentu berdasarkan informasi yang terdapat pada teks digital [31]. Dalam penerapannya, analisis sentimen umumnya mengelompokkan teks ke dalam beberapa kategori sentimen, seperti sentimen positif dan negatif. Proses klasifikasi tersebut dilakukan dengan menganalisis pola bahasa, kata kunci, serta konteks kalimat yang digunakan oleh pengguna. Melalui pendekatan ini, analisis sentimen dapat memberikan gambaran mengenai kecenderungan opini publik terhadap suatu isu atau layanan tertentu [32].

Analisis sentimen telah banyak dimanfaatkan dalam berbagai bidang, seperti pemasaran digital, analisis opini masyarakat di media sosial, serta evaluasi kualitas layanan berdasarkan umpan balik pengguna. Dengan memanfaatkan data yang berasal dari komentar atau ulasan pengguna, organisasi atau instansi dapat memperoleh informasi yang berguna untuk memahami persepsi masyarakat dan meningkatkan kualitas layanan yang diberikan [33]. Dalam penelitian analisis sentimen, proses pelabelan data dilakukan dengan mengelompokkan teks ke dalam kategori sentimen tertentu berdasarkan makna opini yang terkandung di dalamnya. Sentimen positif umumnya ditandai dengan adanya ungkapan kepuasan, dukungan, apresiasi, atau pengalaman baik terhadap suatu layanan maupun sistem. Sebaliknya, sentimen negatif ditandai dengan adanya keluhan, kritik, ketidakpuasan, atau pengalaman buruk yang disampaikan pengguna terhadap suatu objek tertentu. Penentuan label sentimen dilakukan dengan mempertimbangkan konteks keseluruhan kalimat agar makna opini yang terkandung dalam teks dapat dipahami secara tepat. Pendekatan ini banyak digunakan dalam penelitian analisis sentimen pada data media sosial karena mampu membantu model klasifikasi dalam mengenali pola opini pengguna secara lebih akurat [34].

Selain itu, penelitian menjelaskan bahwa komentar yang mengandung ungkapan keluhan, keberatan, atau kritik terhadap suatu kebijakan dikategorikan sebagai sentimen negatif, sedangkan komentar yang menunjukkan dukungan maupun respon positif terhadap suatu layanan dikategorikan sebagai sentimen positif. Penelitian tersebut menggunakan proses pelabelan manual dengan mempertimbangkan konteks kalimat secara utuh agar hasil label lebih konsisten dan sesuai dengan opini pengguna di media sosial [34]. Seiring dengan perkembangan teknologi NLP, metode berbasis *deep learning* khususnya model *transformer*

seperti BERT dan IndoBERT menunjukkan kinerja yang lebih baik dalam tugas analisis sentimen dibandingkan metode *machine learning* tradisional. Model tersebut mampu memahami hubungan kontekstual antar kata dalam suatu kalimat sehingga menghasilkan akurasi klasifikasi sentimen yang lebih tinggi pada data teks, termasuk data yang berasal dari media sosial [35]. Dalam penelitian ini, analisis sentimen digunakan untuk mengidentifikasi kecenderungan opini pengguna terhadap sistem Coretax berdasarkan komentar yang diperoleh dari platform media sosial TikTok.

2.7 Preprocessing

Tahap *preprocessing* menjadi langkah awal dalam pengolahan data teks yang bertujuan membersihkan sekaligus menyiapkan data mentah supaya bisa diproses lebih lanjut oleh model analisis teks secara efektif. Data yang dikumpulkan dari media sosial pada umumnya masih mengandung banyak elemen tidak relevan seperti simbol, kata-kata tidak baku, maupun ketidakkonsistenan dalam penulisan. Karena itu, *preprocessing* diperlukan untuk meningkatkan kualitas data agar analisis yang dilakukan dapat berjalan lebih akurat [36]. Pada penelitian yang menggunakan model *transformer* seperti BERT atau IndoBERT, *preprocessing* yang diterapkan cenderung lebih sederhana dibandingkan pendekatan *machine learning konvensional*. Hal ini disebabkan karena model *transformer* mampu memahami konteks kalimat secara menyeluruh. Dengan demikian, menghilangkan kata-kata umum *(stopword removal*) atau menyederhanakan kata *(stemming)* justru berisiko merusak makna kontekstual yang sangat penting bagi mekanisme *self-attention* pada model tersebut [37]. Dalam penelitian ini, *preprocessing* dilakukan terhadap data komentar pengguna TikTok yang membahas sistem Coretax, agar data tersebut siap diolah oleh model IndoBERT. Adapun tahapan *preprocessing* yang diterapkan adalah sebagai berikut:

1. Data *Cleaning*

   Proses *cleaning* bertujuan untuk membuang elemen-elemen yang tidak diperlukan dari teks komentar. Elemen yang dihapus meliputi tanda baca (seperti titik, koma, tanda tanya, tanda seru), simbol dan karakter khusus, emotikon atau emoji, serta angka-angka yang tidak memberi kontribusi terhadap makna sentimen. Implementasi tahap ini menggunakan *library Regular Expression* (re) yang tersedia dalam bahasa Python [38], [39].

2. *Case Folding*

   Case folding adalah proses mengubah semua huruf dalam teks menjadi huruf kecil (*lowercase*). Langkah ini bertujuan untuk menyamakan format tulisan sehingga model

tidak membedakan antara huruf besar dan huruf kecil yang sebenarnya memiliki makna yang sama. Penerapan *case folding* dilakukan dengan memanfaatkan metode bawaan *Python*, yaitu *.lower*() [37], [38].

3. Normalisasi *Slang*

   Normalisasi merupakan proses mengubah kata-kata tidak baku atau bahasa gaul *(slang)* yang sering muncul di media sosial menjadi bentuk baku sesuai kaidah bahasa Indonesia. Tahap ini penting dilakukan agar model dapat memahami makna komentar secara lebih tepat. Normalisasi dilaksanakan dengan menyusun kamus pemetaan *(mapping dictionary)* yang berisi pasangan kata tidak baku beserta bentuk bakunya, berdasarkan hasil observasi terhadap data komentar yang terkumpul [39], [40].

2.8 Pelabelan Data

Pelabelan data merupakan proses pemberian kategori atau label pada setiap data yang akan digunakan dalam proses pelatihan model *machine learning* atau *deep learning*. Dalam konteks pengolahan bahasa alami atau *Natural Language Processing* (NLP), pelabelan data sering disebut sebagai data *annotation*, yaitu proses memberikan informasi tambahan pada data mentah sehingga data tersebut dapat digunakan untuk melatih model klasifikasi teks secara efektif. Proses ini sangat penting karena kualitas label yang diberikan akan sangat mempengaruhi performa model yang dihasilkan dalam proses analisis data teks [41]. Dalam penelitian analisis sentimen, pelabelan data bertujuan untuk mengidentifikasi kecenderungan opini yang terdapat dalam suatu teks. Setiap teks biasanya dikategorikan ke dalam beberapa kelas sentimen seperti positif dan negatif. Dataset yang telah diberi label kemudian digunakan sebagai data latih untuk membangun model klasifikasi yang mampu mengenali pola sentimen dalam teks secara otomatis [42]. Selain aspek teknis dalam proses pelabelan, kualitas pelabelan data juga sangat dipengaruhi oleh konsistensi dan subjektivitas penilai (*annotator*).

Dalam analisis sentimen, perbedaan persepsi antar penilai dapat menyebabkan ketidakkonsistenan label, terutama pada teks yang mengandung ambiguitas, ironi, atau sarkasme. Oleh karena itu, diperlukan pedoman pelabelan (*annotation guidelines*) yang jelas untuk memastikan setiap data diberi label secara konsisten dan sesuai dengan konteks yang dimaksud [43]. Pedoman pelabelan biasanya mencakup definisi setiap kategori sentimen, contoh kasus untuk masing-masing label, serta aturan dalam menangani teks yang ambigu. Misalnya, komentar yang mengandung keluhan atau kritik terhadap sistem cenderung dikategorikan sebagai sentimen negatif, sedangkan komentar yang menunjukkan kepuasan

atau dukungan terhadap sistem dikategorikan sebagai sentimen positif. Dengan adanya pedoman ini, proses pelabelan dapat dilakukan secara lebih sistematis dan mengurangi potensi bias dalam penentuan label [44]. Selain itu, untuk meningkatkan kualitas pelabelan, beberapa penelitian menyarankan penggunaan lebih dari satu *annotator* dalam proses pelabelan data. Hasil pelabelan dari beberapa annotator kemudian dibandingkan untuk mengukur tingkat kesepakatan menggunakan metrik seperti *inter-annotator agreement*. Nilai kesepakatan yang tinggi menunjukkan bahwa proses pelabelan dilakukan secara konsisten, sehingga dataset yang dihasilkan lebih reliabel untuk digunakan dalam pelatihan model [45]. Selain penggunaan lebih dari satu annotator, tingkat konsistensi pelabelan juga dapat dievaluasi menggunakan *Cohen's Kappa* (κ). *Cohen's Kappa* merupakan metode statistik yang digunakan untuk mengukur tingkat kesepakatan antar penilai (*inter-annotator agreement*) pada data kategorikal dengan mempertimbangkan kemungkinan kesepakatan yang terjadi secara acak. Metode ini banyak digunakan dalam penelitian analisis sentimen untuk mengevaluasi reliabilitas hasil pelabelan manual sebelum dataset digunakan pada proses pelatihan model. Nilai *Cohen's Kappa* berada pada rentang −1 hingga 1, di mana nilai yang semakin mendekati 1 menunjukkan tingkat kesepakatan yang semakin tinggi antar annotator, sedangkan nilai mendekati 0 menunjukkan bahwa kesepakatan yang terjadi cenderung bersifat acak [46].

Perhitungan *Cohen's Kappa* dapat dirumuskan sebagai berikut:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \quad \text{.....(2.1)}$$

Keterangan:

1. P_o = proporsi kesepakatan aktual (*observed agreement*).
2. P_e = proporsi kesepakatan yang terjadi secara acak (*expected agreement*).
3. *K* = nilai koefisien *Cohen's Kappa.*

Interpretasi nilai *Cohen's Kappa* umumnya dibagi menjadi beberapa kategori, yaitu nilai di bawah 0,20 menunjukkan kesepakatan rendah, 0,21–0,40 cukup, 0,41–0,60 sedang, 0,61–0,80 tinggi, dan nilai di atas 0,80 menunjukkan tingkat kesepakatan yang sangat tinggi. Oleh karena itu, penggunaan *Cohen's Kappa* dapat membantu memastikan bahwa proses pelabelan sentimen dilakukan secara konsisten sehingga menghasilkan dataset yang lebih reliabel untuk analisis sentimen [47]. Proses pelabelan data dapat dilakukan melalui beberapa pendekatan, yaitu pelabelan manual oleh manusia, pelabelan otomatis menggunakan algoritma, maupun kombinasi keduanya *(semi-supervised labeling)*.

Pelabelan manual dilakukan dengan membaca setiap teks secara langsung dan menentukan label yang sesuai berdasarkan konteks kalimat. Metode ini sering digunakan dalam penelitian karena mampu menghasilkan label yang lebih akurat meskipun memerlukan waktu dan tenaga yang cukup besar [48]. Di sisi lain, pelabelan otomatis dapat dilakukan dengan menggunakan pendekatan berbasis leksikon, algoritma *machine learning*, maupun bantuan model kecerdasan buatan untuk mempercepat proses anotasi data. Beberapa penelitian terbaru menunjukkan bahwa proses pelabelan data juga dapat dilakukan dengan bantuan model kecerdasan buatan untuk mengurangi biaya dan waktu anotasi dataset. Namun demikian, pelabelan manual masih sering digunakan sebagai standar utama karena mampu mempertimbangkan konteks bahasa yang kompleks seperti ironi atau sarkasme dalam teks [49]. Dalam penelitian ini, proses pelabelan dilakukan pada dataset komentar pengguna TikTok yang membahas mengenai sistem Coretax. Setiap komentar akan diklasifikasikan ke dalam dua kategori sentimen yaitu sentimen positif dan sentimen negatif sesuai dengan tujuan penelitian. Dataset yang telah diberi label kemudian digunakan sebagai data latih dalam proses pelatihan model IndoBERT untuk melakukan klasifikasi sentimen terhadap opini pengguna mengenai implementasi sistem Coretax.

2.9 Pembagian Data

Pembagian data merupakan proses membagi dataset menjadi beberapa bagian yang digunakan dalam proses pelatihan dan evaluasi model machine learning atau *deep learning*. Dalam penelitian berbasis klasifikasi teks, dataset umumnya dibagi menjadi dua bagian utama yaitu data *training* dan data *testing*. Data *training* digunakan untuk melatih model agar mampu mempelajari pola dari dataset, sedangkan data *testing* digunakan untuk mengevaluasi kemampuan model dalam melakukan prediksi terhadap data yang belum pernah digunakan sebelumnya [50]. Pembagian dataset menjadi data *training* dan data *testing* sangat penting dalam proses pembangunan model karena dapat membantu menilai kemampuan model dalam melakukan generalisasi terhadap data baru. Jika seluruh data digunakan untuk proses pelatihan tanpa dilakukan pemisahan, maka evaluasi model menjadi tidak objektif dan berpotensi menghasilkan performa yang terlalu tinggi akibat model hanya menghafal data yang digunakan pada saat pelatihan [51]. Dalam penelitian *machine learning* dan analisis sentimen, rasio pembagian dataset yang umum digunakan adalah 80% data *training* dan 20% data *testing* atau 70% data *training* dan 30% data *testing*. Apabila distribusi sentimen dalam dataset cenderung tidak seimbang (misalnya jumlah komentar negatif jauh lebih banyak daripada positif), maka proses pembagian data perlu menerapkan teknik

stratified sampling. *Stratified sampling* memastikan bahwa proporsi setiap kelas (positif dan negatif) pada data *training*, data *validation*, dan data *testing* tetap sama dengan proporsi pada dataset asli, sehingga model tidak menjadi bias terhadap kelas mayoritas dan evaluasi kinerja menjadi lebih akurat [52]. Selain *stratified sampling*, penerapan nilai acak tetap (*random seed*) juga merupakan bagian penting dalam pembagian data. *Random seed* digunakan untuk memastikan proses pembagian data menghasilkan subset yang identik setiap kali kode dijalankan ulang, sehingga hasil eksperimen bersifat *reproducible* dan dapat diverifikasi oleh pihak lain [53].

Beberapa penelitian menunjukkan bahwa penggunaan 80% data sebagai data pelatihan dan 20% sebagai data pengujian dapat memberikan hasil evaluasi model yang baik karena model memiliki data yang cukup untuk mempelajari pola, sekaligus menyediakan data yang memadai untuk proses evaluasi performa model [30]. Selain itu, pembagian dataset juga dapat membantu mengurangi risiko *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan diri dengan data *training* sehingga tidak mampu melakukan prediksi dengan baik terhadap data baru [30]. Dalam penelitian ini, dataset komentar pengguna TikTok yang telah melalui tahap *preprocessing* dan pelabelan data akan dibagi menjadi tiga bagian yaitu data *training*, data *validation*, dan data *testing*. Karena data komentar TikTok cenderung didominasi sentimen negatif, pembagian data akan dilakukan dengan teknik *stratified sampling* menggunakan *rasio* 80% data *training*, 10% data *validation*, dan 10% data *testing*. Data *training* digunakan untuk melatih model IndoBERT dalam mempelajari pola sentimen pada komentar pengguna, data *validation* digunakan untuk mengevaluasi model selama proses pelatihan, sedangkan data *testing* digunakan untuk mengevaluasi performa akhir model dalam mengklasifikasikan sentimen positif dan negatif terhadap implementasi sistem Coretax.

Meskipun *stratified sampling* mampu mempertahankan proporsi distribusi kelas pada setiap pembagian dataset, teknik ini tidak secara langsung mengatasi permasalahan ketidakseimbangan jumlah data antar kelas (*class imbalance*). Ketidakseimbangan kelas terjadi ketika jumlah data pada suatu kelas jauh lebih besar dibandingkan kelas lainnya sehingga model cenderung lebih banyak mempelajari pola dari kelas mayoritas dan kurang optimal dalam mengenali kelas minoritas. Kondisi tersebut dapat menyebabkan penurunan performa klasifikasi terutama pada kelas dengan jumlah data yang lebih sedikit. Oleh karena itu, diperlukan metode tambahan untuk membantu model mempelajari distribusi data secara lebih seimbang. Salah satu metode yang dapat digunakan adalah *class weight*, yaitu teknik pemberian bobot berbeda pada setiap kelas berdasarkan jumlah sampelnya. Untuk mengatasi

ketidakseimbangan kelas (*class imbalance*) antara sentimen positif dan sentimen negatif pada dataset, penelitian ini menerapkan metode *class weight*. *Class weight* bekerja dengan cara memberikan bobot yang lebih tinggi pada kelas minoritas dan bobot yang lebih rendah pada kelas mayoritas di dalam fungsi loss selama proses pelatihan model. Dengan demikian, model dipaksa untuk memperhatikan kelas minoritas secara lebih proporsional meskipun jumlah datanya jauh lebih sedikit.

Nilai *class weight* dihitung menggunakan rumus berikut:

$$w_i = N/(K \times n_i) \quad \text{.....(2.2)}$$

Keterangan:

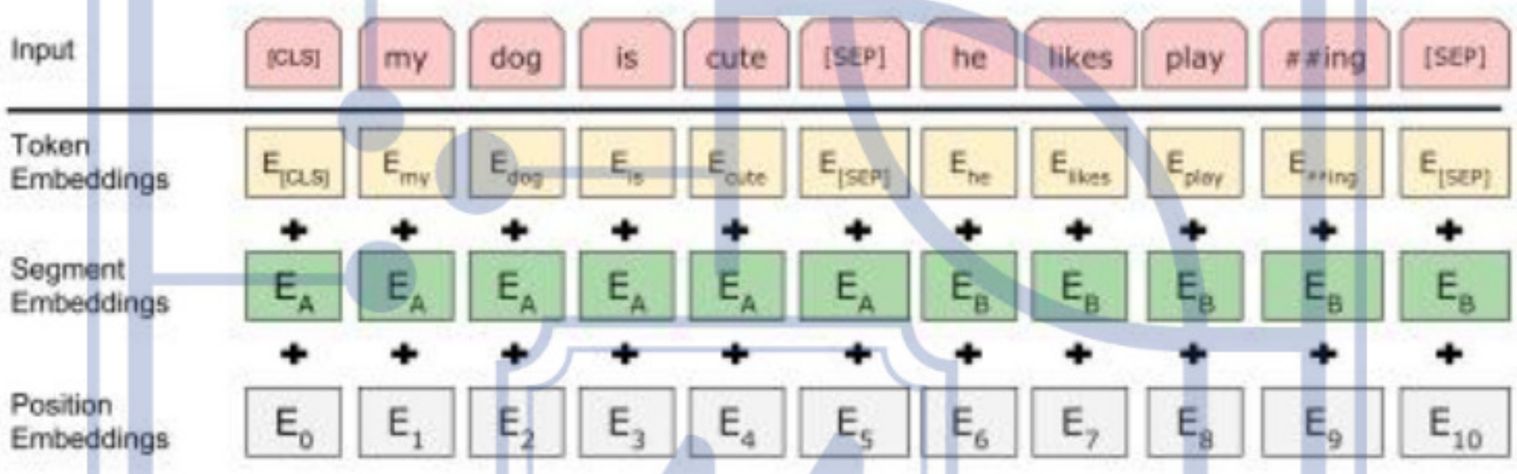
- w_i = bobot untuk kelas ke-i
- N = jumlah total sampel dataset
- K = jumlah kelas (dalam penelitian ini K = 2)
- n_i = jumlah sampel pada kelas ke-i

Kelas dengan jumlah data lebih sedikit diberikan bobot yang lebih besar, sedangkan kelas mayoritas diberikan bobot yang lebih kecil agar model dapat memberikan perhatian yang lebih baik terhadap kelas minoritas selama proses pelatihan. Dalam penelitian ini, penerapan *class weight* digunakan karena distribusi sentimen komentar pengguna TikTok terhadap sistem Coretax menunjukkan kecenderungan dominasi sentimen negatif. Oleh karena itu, pemberian bobot kelas diterapkan pada tahap pelatihan model IndoBERT untuk membantu meningkatkan kemampuan model dalam mengenali kedua kelas sentimen secara lebih seimbang [54].

2.10 BERT (Bidirectional Encoder Representations From Transformers)

BERT *(Bidirectional Encoder Representations from Transformers)* merupakan mod el bahasa berbasis arsitektur *transformer* yang dikembangkan oleh Google untuk memahami konteks kata dalam suatu teks secara dua arah *(bidirectional)*. Berbeda dengan pendekatan model bahasa sebelumnya yang memproses teks secara sekuensial dalam satu arah, BERT mampu menganalisis hubungan antar kata dengan mempertimbangkan konteks dari kedua sisi kalimat secara bersamaan. Pendekatan ini memungkinkan model menghasilkan representasi teks yang lebih kaya secara kontekstual sehingga meningkatkan kinerja pada berbagai tugas *Natural Language Processing* (NLP) seperti klasifikasi teks, analisis sentimen, dan *question answering*. Secara arsitektur, BERT dibangun menggunakan model

Transformer Encoder yang terdiri dari beberapa lapisan *encoder* yang tersusun secara bertingkat. Setiap lapisan *encoder* memiliki dua komponen utama yaitu mekanisme *multi-head self attention* dan *feed forward neural networ*k. Mekanisme *self attention* memungkinkan model untuk memahami hubungan antar kata dalam suatu kalimat dengan memperhatikan konteks dari setiap kata terhadap kata lainnya. Dengan arsitektur ini, setiap token dalam kalimat dapat diproses dengan mempertimbangkan keseluruhan konteks kalimat secara bersamaan sehingga menghasilkan representasi bahasa yang lebih kontekstual [35]. Untuk memahami struktur dasar model BERT, arsitektur model tersebut dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Arsitektur Model BERT [35]

Gambar 2.3 menunjukkan arsitektur model BERT yang terdiri dari beberapa komponen utama sebagai berikut [55]:

1. Input Token

   Proses dimulai dari teks yang telah diubah menjadi token. Setiap kata dalam kalimat direpresentasikan sebagai token yang kemudian menjadi masukan bagi model.

2. *Embedding Layer*

   Token yang telah dihasilkan selanjutnya dikonversi menjadi representasi numerik melalui *embedding layer* yang dilambangkan sebagai E_1, E_2, hingga E_n. *Embedding* ini berfungsi untuk merepresentasikan setiap token dalam bentuk vektor sehingga dapat diproses oleh model.

3. *Transformer Encoder*

   Representasi *embedding* kemudian diproses melalui beberapa lapisan *Transformer Encoder* (Trm). Setiap lapisan bekerja secara bertingkat untuk mengolah informasi dari token yang diberikan.

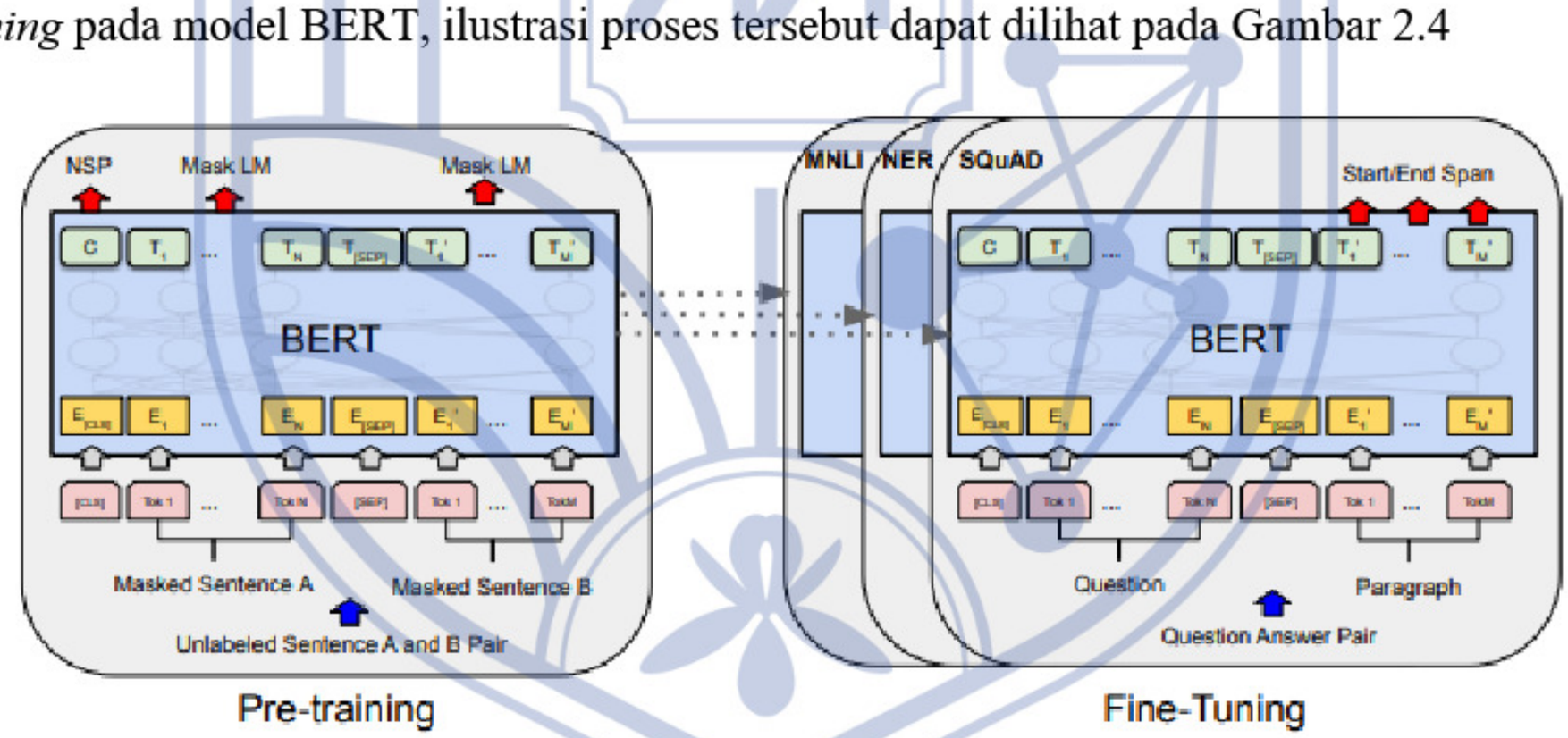
4. Mekanisme *Self-Attention*

   Pada setiap lapisan *transformer*, digunakan mekanisme *self-attention* yang memungkinkan model memahami hubungan antar kata dalam satu kalimat, sehingga setiap kata dapat saling mempengaruhi dalam proses representasi.

5. Representasi Kontekstual

   Melalui proses berlapis tersebut, model menghasilkan representasi kontekstual untuk setiap token. Representasi ini mempertimbangkan seluruh informasi dalam kalimat secara dua arah (*bidirectional*), sehingga mampu memahami makna teks secara lebih akurat [55].

Model BERT dilatih menggunakan kumpulan data teks dalam jumlah besar melalui tahap *pre-training* untuk mempelajari pola bahasa serta hubungan antar kata dalam suatu kalimat. Setelah proses tersebut, model dapat disesuaikan kembali melalui tahap *fine-tuning* agar dapat digunakan pada berbagai tugas spesifik dalam NLP. Dalam analisis sentimen, kemampuan BERT dalam memahami konteks kalimat memungkinkan model mengidentifi kasi kecenderungan opini atau emosi yang terdapat dalam teks secara lebih akurat [56]. Berdasarkan kemampuan BERT dalam memahami konteks bahasa secara mendalam, model ini kemudian dikembangkan lebih lanjut untuk berbagai bahasa, termasuk bahasa Indonesia. Salah satu pengembangannya adalah IndoBERT yang dirancang khusus untuk memahami karakteristik Bahasa Indonesia, yang akan dibahas pada subbab berikutnya. Untuk member ikan gambaran yang lebih jelas mengenai perbedaan antara tahap *pre-training* dan *fine-tuning* pada model BERT, ilustrasi proses tersebut dapat dilihat pada Gambar 2.4



Gambar 2. 4 Ilustrasi tahapan *Pre-training* dan *Fine-Tuning* [35]

Berdasarkan Gambar 2.4, terdapat dua fase utama yang diilustrasikan, yaitu fase *pre-training* (bagian atas gambar) dan fase *fine-tuning* (bagian bawah gambar). Kedua fase tersebut dijelaskan sebagai berikut [35]:

1. Fase *Pre-training* (Pelatihan Awal)
   a. Model BERT dilatih menggunakan data dalam jumlah yang sangat besar dan tidak berlabel (*unlabeled* data).
   b. Pelatihan dilakukan melalui dua tugas utama secara simultan, yaitu:

i. *Masked Language* Model (Mask LM) : Mengajarkan model untuk menebak kata-kata yang sengaja dihilangkan (ditandai dengan token [MASK]).
ii. *Next Sentence Prediction* (NSP) : Mengajarkan model untuk memahami hubungan antar kalimat dengan memprediksi apakah suatu kalimat merupakan kelanjutan logis dari kalimat sebelumnya.

c. Melalui Mask LM, model belajar memahami hubungan antar kata secara dua arah *(bidirectional)*, yang menjadi keunggulan utama arsitektur BERT.

2. Fase *Fine-tuning* (Penyesuaian Halus)
   a. Arsitektur dasar model yang telah melalui fase *pre-training* tetap dipertahankan.
   b. Lapisan keluaran (*output layer*) model diganti dengan lapisan baru yang disesuaikan dengan tugas spesifik yang ingin diselesaikan, misalnya:
      i. Klasifikasi sentimen
      ii. Ekstraksi jawaban (*Start/End Span*) pada tugas *question answering*
   c. Proses *fine-tuning* dilakukan dengan memanfaatkan data berlabel (labeled data) dalam jumlah yang relatif lebih kecil dibandingkan data *pre-training*.
   d. Seluruh parameter dalam model ikut disesuaikan (*fine-tuned*) agar model mampu menyelesaikan tugas spesifik yang diinginkan secara optimal [35].

Model IndoBERT yang sudah melewati tahap *pre-training* kemudian diadaptasi lebih lanjut melalui proses *fine-tuning* agar dapat digunakan untuk tugas klasifikasi sentimen dalam penelitian ini. Langkah pertama yang dilakukan adalah menentukan arsitektur model yang akan dipakai. Pada umumnya, varian IndoBERT yang digunakan adalah tipe *base* yang memiliki 12 lapisan tersembunyi serta 768 unit tersembunyi, karena arsitektur ini dinilai mampu memberikan keseimbangan yang baik antara performa yang dihasilkan dengan kebutuhan sumber daya komputasi. Setelah itu, pada bagian atas model dasar ditambahkan sebuah lapisan klasifikasi yang terdiri dari lapisan *dropout* untuk mencegah terjadinya *overfitting*, kemudian dilanjutkan dengan lapisan *fully-connected* yang jumlah *neuron*-nya disesuaikan dengan banyaknya kelas sentimen yang ingin diprediksi. Fungsi aktivasi *softmax* juga digunakan pada lapisan akhir agar model dapat menghasilkan nilai probabilitas untuk setiap kelas. Selanjutnya, model dikompilasi dengan menentukan *optimizer* seperti AdamW, fungsi *loss* yang sesuai dengan format label yang digunakan (misalnya *Sparse Categorical Crossentropy* atau *Categorical Crossentropy*), serta metrik evaluasi seperti akurasi. Tahap berikutnya adalah menentukan berbagai *hyperparameter* penting, di antaranya *learning rate* yang umumnya berada pada rentang 2e-5 hingga 5e-5, *batch size* sebesar 16 atau 32 tergantung pada kapasitas perangkat yang tersedia, serta jumlah *epoch* sekitar 3 hingga 5 kali

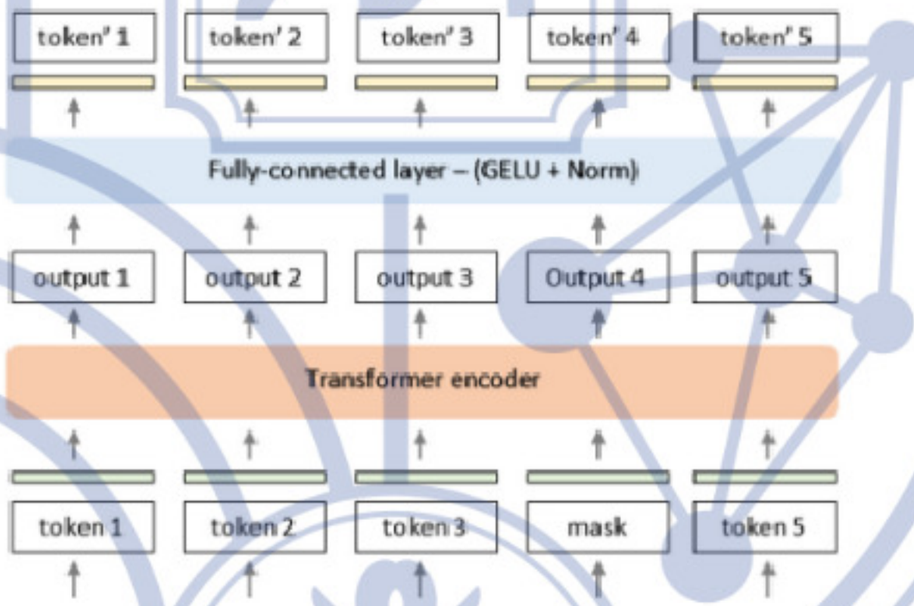
untuk menghindari risiko *overfitting*. Proses pelatihan kemudian dijalankan menggunakan data latih, dengan evaluasi secara berkala terhadap data validasi pada setiap akhir *epoch* guna memantau sejauh mana performa model berkembang[57], [58].

2.11 IndoBERT (Indonesian *Bidirectional Encoder Representation From Transfomers*)

IndoBERT merupakan model bahasa berbasis arsitektur *transformer* yang dirancang secara khusus untuk pemrosesan bahasa Indonesia. Model ini merupakan pengembangan dari BERT yang telah dilatih menggunakan berbagai korpus teks berbahasa Indonesia dalam jumlah besar, seperti artikel berita, halaman *web,* dan data teks lainnya. Melalui proses pelatihan tersebut, IndoBERT mampu mempelajari pola bahasa, hubungan antar kata, serta makna kontekstual dalam bahasa Indonesia secara lebih efektif dibandingkan model bahasa umum yang tidak secara khusus dilatih menggunakan data bahasa Indonesia [40]. Berbagai penelitian menunjukkan bahwa IndoBERT memiliki kinerja yang sangat baik dalam berbagai tugas *Natural Language Processing (NLP)*, termasuk klasifikasi teks, analisis sentimen, serta pengenalan entitas dalam teks berbahasa Indonesia. Kemampuan model ini dalam memahami konteks kalimat secara mendalam membuat IndoBERT mampu menghasilkan performa yang lebih tinggi dibandingkan beberapa metode berbasis *machine learning konvensional* [7]. Selain itu, beberapa penelitian juga melaporkan bahwa IndoBERT mampu memberikan hasil yang lebih optimal dibandingkan model *deep learning* lainnya seperti *Long Short-Term Memory* (LSTM) dalam tugas klasifikasi teks berbahasa Indonesia. Keunggulan tersebut disebabkan oleh mekanisme *self-attention* pada arsitektur transformer yang memungkinkan model menangkap hubungan kontekstual antar kata secara lebih efektif dalam suatu kalimat [59].

Proses pelatihan IndoBERT dilakukan dengan pendekatan *pre-training* menggunakan data dalam jumlah besar, kemudian dilanjutkan dengan *fine-tuning* pada tugas tertentu. Dalam tahap *pre-training*, model dilatih untuk memahami struktur bahasa melalui berbagai teknik pembelajaran berbasis konteks. Selain itu, IndoBERT juga dilatih menggunakan dataset besar seperti Indo4B yang mencakup berbagai jenis teks formal maupun informal dalam bahasa Indonesia [60]. Model yang digunakan dalam penelitian ini adalah IndoBERT sebagai model dasar. IndoBERT merupakan *pretrained language model* berbasis BERT yang telah dilatih menggunakan korpus berbahasa Indonesia sehingga mampu memahami konteks linguistik, struktur kalimat, serta variasi penggunaan bahasa dalam Bahasa Indonesia. Model ini dibangun menggunakan arsitektur *transformer* yang

memungkinkan pemahaman konteks secara mendalam terhadap teks. IndoBERT memiliki beberapa varian, seperti IndoBERT-base dan IndoBERT-*large*, yang dibedakan berdasarkan jumlah parameter dan kedalaman lapisan transformernya. Varian IndoBERT-base lebih ringan dan umum digunakan untuk tugas klasifikasi teks, termasuk analisis sentimen pada dataset berukuran menengah. Secara arsitektur, IndoBERT-base terdiri dari 12 lapisan *transformer* dengan dimensi vektor sebesar 768 [61]. Selain itu, indoBERT juga memiliki beberapa varian lainnya yang dibedakan berdasarkan proses *pre-training* dan dataset yang digunakan, yaitu IndoBERT-base-p1 dan IndoBERT-base-p2. Perbedaan keduanya terletak pada jumlah serta keberagaman data pelatihan, di mana IndoBERT-base-p2 dilatih menggunakan dataset yang lebih besar dibandingkan IndoBERT-base-p1. Dengan dataset yang lebih luas, IndoBERT-base-p2 mampu memahami konteks bahasa Indonesia secara lebih baik, khususnya pada teks tidak baku seperti data media sosial. Oleh karena itu, penelitian ini menggunakan IndoBERT-base-p2 karena dinilai lebih efektif dalam meningkatkan kinerja klasifikasi sentimen pada komentar pengguna TikTok. Proses tokenisasi pada model IndoBERT yang mengubah teks menjadi representasi token dapat dilihat pada Gambar 2.5.



Gambar 2. 5 Tahap Tokenisasi IndoBERT [61]

Proses klasifikasi sentimen menggunakan IndoBERT diawali dengan tahap tokenisasi, di mana setiap kata atau token dalam teks direpresentasikan dalam bentuk vektor. Pada tahap ini, model menambahkan token khusus, yaitu [CLS] di awal kalimat sebagai representasi keseluruhan input, [SEP] sebagai penanda batas kalimat, serta [PAD] untuk menyeragamkan panjang *sekuens* dalam satu *batch*. Seluruh token kemudian diubah menjadi *indeks* numerik berdasarkan kosakata yang dimiliki oleh *tokenizer* IndoBERT. Selanjutnya, indeks token tersebut diproyeksikan ke dalam ruang vektor melalui proses *embedding* yang mencakup token *embedding*, *position embedding*, dan *segment embedding*. Hasil *embedding* ini menjadi input bagi arsitektur IndoBERT yang terdiri dari 12 lapisan *encoder*

Transformer. Setiap lapisan memanfaatkan mekanisme *multi-head self-attention* untuk memahami hubungan antar-token secara dua arah, sehingga konteks setiap kata dapat ditangkap secara menyeluruh.Representasi keluaran dari token [CLS] kemudian digunakan sebagai ringkasan informasi seluruh teks dan diteruskan ke lapisan klasifikasi berupa *fully connected layer* untuk menghasilkan nilai *logits* sesuai jumlah kelas. Pada tahap akhir, fungsi aktivasi *softmax* digunakan untuk mengonversi *logits* menjadi probabilitas, sehingga model dapat menentukan label sentimen dari setiap input teks [61].

Word embedding merupakan teknik yang digunakan untuk merepresentasikan kata dalam bentuk vektor berdimensi tinggi, sehingga komputer dapat memahami aspek semantik dan sintaksis dari bahasa. Metode tradisional seperti Word2Vec dan GloVe menghasilkan representasi vektor yang bersifat statis dan belum mampu mempertimbangkan konteks dalam suatu kalimat.Keterbatasan tersebut kemudian diatasi melalui pendekatan *contextual embedding* berbasis arsitektur Transformer, seperti BERT, yang memanfaatkan mekanisme *self-attention* untuk memahami makna kata secara dua arah. Pendekatan ini memungkinkan model seperti IndoBERT menangkap makna bahasa yang lebih kompleks dan kontekstual dibandingkan dengan metode *embedding konvensional* [62]. Secara formal, *embedding* dapat dirumuskan sebagaimana pada Persamaan (2.3).

$$E = [e_1, e_2, \dots, e_n], \quad e_i \in R^d \qquad (2.3)$$

dengan *ei* merepresentasikan vektor embedding untuk token *ke-i*, dan *d* menyatakan dimensi *embedding*. Pada model IndoBERT, representasi *embedding* setiap token diperoleh melalui penjumlahan tiga komponen utama, yaitu *token embedding* EtokenE_{token}Etoken, *segment embedding* EsegmentE_{segment}Esegment, dan *position embedding* EpositionE _{position}Eposition, sebagaimana dirumuskan pada Persamaan (2.4).

$$E_i = E_{token,i} + E_{segment,i} + E_{position,i} \qquad (2.4)$$

dengan E_i menyatakan representasi *embedding* akhir untuk token ke-i, serta dimensi *embedding* sebesar d=768 pada model IndoBERT Base. Kombinasi ketiga jenis *embedding* tersebut memungkinkan model tidak hanya menangkap makna kata, tetapi juga memahami urutan serta posisi kata dalam suatu kalimat secara lebih komprehensif [62].

Mekanisme *self-attention* merupakan komponen utama yang membedakan BERT dari model bahasa sebelumnya, karena memungkinkan setiap token dalam suatu teks untuk berinteraksi dan memperhatikan seluruh token lainnya secara simultan, baik yang berada sebelum maupun sesudahnya. Mekanisme ini memanfaatkan tiga representasi vektor utama, yaitu *Query* (Q), *Key* (K), dan *Value* (V), yang digunakan untuk menghitung bobot perhatian (*attention weight*) dalam proses pembentukan representasi konteks [62].

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad \text{.....} \quad (2.5)$$

Rumus tersebut menentukan seberapa besar kontribusi setiap token terhadap token lain dalam satu kalimat. Fungsi *Softmax* mengonversi sekumpulan skor logit menjadi distribusi probabilitas antarkelas, memastikan total probabilitas sama dengan 1 [62].

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{.....} \quad (2.6)$$

Softmax digunakan dalam tahap *fine-tuning* IndoBERT untuk menghasilkan probabilitas sentimen positif dan negatif sebelum dioptimasi dengan fungsi *Cross-Entropy Loss* [59]. Dengan karakteristik tersebut, IndoBERT memiliki keunggulan dalam memahami makna serta struktur teks berbahasa Indonesia secara lebih efektif dibandingkan metode *konvensional*. Keunggulan ini menjadikan IndoBERT banyak dimanfaatkan dalam berbagai tugas *Natural Language Processing* (NLP), khususnya analisis sentimen, serta mampu menunjukkan performa yang lebih unggul dibandingkan beberapa model pralatih lainnya maupun metode klasifikasi tradisional seperti *Support Vector Machine*, *Naïve Bayes*, dan *K-Nearest Neighbor*. Oleh karena itu, penggunaan IndoBERT dinilai sesuai untuk analisis sentimen pada data komentar di media sosial yang bersifat tidak terstruktur, beragam, dan memiliki volume besar, sehingga memungkinkan identifikasi persepsi publik secara lebih akurat [63].

2.12 Confusion Matrix

Confusion matrix merupakan salah satu metode evaluasi yang digunakan untuk menilai kinerja model klasifikasi dengan membandingkan hasil prediksi yang dihasilkan oleh model dengan label sebenarnya dari data. Metode ini memberikan gambaran mengenai kemampuan model dalam mengklasifikasikan data secara benar serta membantu mengidentifikasi kesalahan prediksi yang terjadi selama proses klasifikasi [63]. *Confusion matrix* terdiri dari empat komponen utama, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Keempat komponen tersebut digunakan untuk menggambarkan hasil prediksi model terhadap data yang diuji sehingga dapat diketahui tingkat keberhasilan maupun kesalahan klasifikasi yang dihasilkan oleh model [64].

Dalam penelitian analisis sentimen, *confusion matrix* digunakan untuk mengevaluasi kemampuan model dalam mengklasifikasikan teks ke dalam kategori sentimen tertentu, seperti sentimen positif dan sentimen negatif [65]. Untuk mengevaluasi kinerja model klasifikasi sentimen, digunakan *confusion matrix* yang disajikan pada Tabel 2.1.

Tabel 2. 1 *Confusion Matrix* untuk Klasifikasi Sentimen [66]

Data Aktual	**Prediksi Positif**	**Prediksi Negatif**
Positif	*True Positive* (TP)	*False Negative* (FN)
Negatif	*False Positive* (FP)	*True Negative* (TN)

Keterangan:

1. *True Positive* (TP) merupakan jumlah data yang sebenarnya memiliki sentimen positif dan berhasil diprediksi sebagai positif oleh model.
2. *True Negative* (TN) merupakan jumlah data yang sebenarnya memiliki sentimen negatif dan berhasil diprediksi sebagai negatif oleh model.
3. *False Positive* (FP) merupakan jumlah data yang sebenarnya memiliki sentimen negatif tetapi diprediksi sebagai positif oleh model.
4. *False Negative* (FN) merupakan jumlah data yang sebenarnya memiliki sentimen positif tetapi diprediksi sebagai negatif oleh model

Berdasarkan nilai yang terdapat pada *confusion matrix*, beberapa metrik evaluasi dapat dihitung untuk mengukur performa model klasifikasi. Metrik evaluasi yang umum digunakan meliputi *accuracy,* precision, *recall*, dan *F1-score*. *Accuracy* digunakan untuk mengetahui tingkat ketepatan model dalam melakukan prediksi secara keseluruhan, sedangkan *precision* dan *recall* digunakan untuk mengukur kemampuan model dalam mengidentifikasi kelas tertentu. Selain itu, *F1-score* merupakan kombinasi antara *precision* dan *recall* yang digunakan untuk memberikan gambaran performa model secara lebih seimbang. Dalam penelitian ini, *confusion matrix* digunakan untuk mengevaluasi kinerja model IndoBERT dalam melakukan klasifikasi sentimen terhadap komentar pengguna mengenai sistem Coretax yang diperoleh dari platform media sosial TikTok.

2.12.1 Precision

Precision merupakan salah satu metrik evaluasi yang digunakan untuk mengukur tingkat ketepatan model dalam memprediksi kelas positif. Nilai *precision* menunjukkan perbandingan antara jumlah data yang diprediksi positif dengan benar (*True Positive*) terhadap seluruh data yang diprediksi sebagai positif oleh model. Dengan kata lain, *precision* menggambarkan seberapa akurat model dalam mengidentifikasi data yang termasuk ke dalam kelas positif.

Nilai *precision* dapat dihitung menggunakan persamaan berikut:

$$Precision = \frac{TP}{(TP + FP)} \quad \text{.....(2.7)}$$

Keterangan:

1. TP (*True Positive*) : jumlah data positif yang diprediksi dengan benar oleh model.
2. FP (*False Positive*) : jumlah data yang sebenarnya negatif tetapi diprediksi sebagai positif oleh model.

Semakin tinggi nilai *precision*, maka semakin baik kemampuan model dalam memprediksi kelas positif secara tepat tanpa menghasilkan banyak kesalahan prediksi positif.

2.12.2 Recall

Recall merupakan metrik evaluasi yang digunakan untuk mengukur kemampuan model dalam mengidentifikasi seluruh data yang termasuk ke dalam kelas positif. Nilai *recall* menunjukkan perbandingan antara jumlah data positif yang berhasil diprediksi dengan benar (*True Positive*) terhadap seluruh data yang sebenarnya memiliki label positif. Dengan demikian, *recall* menggambarkan seberapa baik model mampu menemukan semua data yang termasuk dalam kelas positif.

Nilai *recall* dapat dihitung menggunakan persamaan berikut:

$$Recall = \frac{\text{TP}}{\text{TP+FN}} \quad \text{.....(2.8)}$$

Keterangan:

1. TP (*True Positive*) : jumlah data positif yang diprediksi dengan benar oleh model.
2. FN (*False Negative*) : jumlah data yang sebenarnya positif tetapi diprediksi sebagai negatif oleh model.

Semakin tinggi nilai *recall*, maka semakin baik kemampuan model dalam mendeteksi seluruh data yang termasuk dalam kelas positif .

2.12.3 F1-Score

F1- Score merupakan metrik evaluasi yang digunakan untuk mengukur keseimbang an antara nilai *precision* dan *recall* dalam suatu model klasifikasi. Nilai *F1-Score* dihitung sebagai rata-rata harmonis dari *precision* dan *recall* sehingga mampu memberikan gambaran performa model secara lebih menyeluruh, terutama ketika terdapat ketidakseimbangan jumlah data pada masing-masing kelas.

Nilai dapat dihitung menggunakan persamaan berikut:

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad \text{.....(2.9)}$$

Nilai *F1-Score* akan berada pada rentang 0 hingga 1. Semakin mendekati nilai 1, maka semakin baik performa model dalam melakukan klasifikasi data karena menunjukkan keseimbangan yang baik antara *precision* dan *recall.*

2.12.4 Accuracy

Accuracy merupakan metrik evaluasi yang digunakan untuk mengukur tingkat ketepatan model dalam melakukan klasifikasi secara keseluruhan. Nilai *accuracy* menunju kkan perbandingan antara jumlah prediksi yang benar terhadap seluruh data yang digunakan dalam proses pengujian model.

Nilai *accuracy* dapat dihitung menggunakan persamaan berikut:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (2.10)$$

Keterangan:

1. TP (*True Positive*) : jumlah data positif yang diprediksi dengan benar oleh model
2. TN (*True Negative*) : jumlah data negatif yang diprediksi dengan benar oleh model
3. FP (*False Positive*) : jumlah data negatif yang diprediksi sebagai positif oleh model
4. FN (*False Negative*) : jumlah data positif yang diprediksi sebagai negatif oleh model

Semakin tinggi nilai *accuracy* yang diperoleh, maka semakin baik kemampuan model dalam melakukan klasifikasi data secara keseluruhan.

2.13 Word Cloud

Word Cloud merupakan salah satu teknik visualisasi pada data teks yang menyajikan kata-kata berdasarkan tingkat frekuensi kemunculannya dalam suatu kumpulan dokumen. Kata yang muncul lebih sering akan ditampilkan dengan ukuran huruf yang lebih besar, sedangkan kata dengan frekuensi yang lebih rendah akan ditampilkan dalam ukuran yang lebih kecil. Melalui visualisasi tersebut, peneliti dapat memperoleh gambaran mengenai kata atau topik yang paling dominan sehingga karakteristik data dapat dipahami secara lebih mudah sebelum dilakukan analisis pada tahap berikutnya [67]. Dalam bidang Natural Language Processing (NLP), *Word Cloud* banyak dimanfaatkan sebagai bagian dari proses eksplorasi data (Exploratory Data Analysis atau EDA). Visualisasi ini membantu peneliti mengenali kata-kata yang paling sering muncul, mengamati kecenderungan topik yang dibahas, serta memahami pola umum dari sekumpulan data teks. Meskipun tidak dapat digunakan untuk menentukan kategori sentimen secara langsung, Word Cloud tetap

memberikan informasi deskriptif yang berguna mengenai distribusi kata pada data yang dianalisis [68].

Pada penelitian analisis sentimen, Word Cloud umumnya diterapkan setelah tahapan preprocessing selesai dilakukan. Proses seperti cleaning, case folding, dan normalisasi bertujuan menghilangkan unsur-unsur yang tidak diperlukan sehingga kata-kata yang divisualisasikan menjadi lebih representatif terhadap isi data. Dengan demikian, hasil visualisasi mampu menggambarkan kata-kata penting yang benar-benar mencerminkan pembahasan pengguna tanpa dipengaruhi oleh karakter khusus, simbol, maupun variasi penulisan yang tidak baku [67], [68]. Dalam penelitian ini, Word Cloud digunakan untuk menampilkan kata-kata yang paling sering muncul pada komentar pengguna TikTok mengenai implementasi sistem Coretax. Visualisasi tersebut dimanfaatkan sebagai analisis awal untuk mengetahui topik-topik yang paling banyak dibahas oleh pengguna sebelum dilakukan proses klasifikasi sentimen menggunakan model IndoBERT. Selain itu, hasil Word Cloud juga berfungsi sebagai pendukung dalam menginterpretasikan hasil analisis sentimen dengan menunjukkan kata-kata yang memiliki frekuensi kemunculan paling tinggi pada dataset penelitian.