BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Sejumlah studi sebelumnya telah mengeksplorasi penerapan kecerdasan buatan dalam ranah analisis sentimen, mulai dari pemrosesan teks bahasa tunggal hingga pendekatan multimodal yang lebih kompleks. Kajian terhadap literatur-literatur ini sangat esensial untuk memetakan batasan teknologi saat ini, khususnya dalam menangani data *e-commerce* berskala masif yang berhadapan dengan tantangan bahasa campuran (*code-mixed*) serta tingginya variasi kualitas visual dari *User-Generated Content*. Melalui tinjauan sistematis terhadap penelitian-penelitian terdahulu ini, berbagai celah penelitian dapat diidentifikasi secara objektif sehingga rancangan arsitektur hibrida yang diusulkan dalam penelitian ini menjadi lebih terarah dan relevan.

Tabel 2.1 *Penelitian Terdahulu[FA1]*

Author	Dataset	Metode	Tujuan Penelitian	Hasils
[9]	20.000 sampel teks ulasan *code-mixed* Banglish yang diekstraksi secara spesifik dari platform Facebook, YouTube, dan situs *e-commerce*.	*Transformer-based Language Model* dengan pra-pelatihan khusus pada korpus teks transliterasi berskala besar.	Mengklasifikasikan sentimen spesifik pada himpunan data teks yang menggunakan bahasa campuran.	Model terbukti sangat andal dan fleksibel dalam membaca pola bahasa campuran (Banglish), namun pendekatannya yang murni unimodal teks menyebabkan model gagal menangkap sentimen negatif jika keluhan hanya terdapat pada foto bukti produk.
[10]	Korpus teks berbahasa Bengali (Bangla) dan *code-mixed* yang bersumber dari media sosial dan platform internet.	*Transformer-driven Methodology* menggunakan arsitektur MuRIL tingkat lanjut untuk pemrosesan konteks multibahasa.	Mengeksplorasi kemampuan MuRIL untuk klasifikasi emosi multibahasa dan teks Bangla campuran.	Arsitektur MuRIL terbukti sangat andal dalam memetakan nuansa emosi pada ejaan Bangla yang kompleks namun implementasinya masih terbatas pada ranah teks (unimodal) dan belum dipadukan dengan *vision encoder*.

Author	**Dataset**	**Metode**	**Tujuan Penelitian**	**Hasil**
[16]	Dataset Bengali-English code-mixed yang berisi kata Bangla, English, named entity, dan unknown word, serta divalidasi menggunakan dataset Borhan untuk evaluasi tambahan.	Mengusulkan Modified BERT-base-NER yang menggabungkan contextual embedding BERT dengan BiLSTM untuk melakukan identifikasi bahasa pada tingkat kata (word-level language identification).	Mengembangkan model yang mampu mengenali kata Bangla, English, named entity, dan unknown word secara lebih akurat pada teks Bengali-English code-mixed dibandingkan metode machine learning dan deep learning sebelumnya.	Model yang diusulkan mencapai akurasi 95% dan F1-score 0,94, secara signifikan mengungguli Naïve Bayes, SVM, CNN-LSTM, dan BiLSTM berdasarkan evaluasi 5-fold cross validation dan uji statistik ($p < 0,0001$).
[17]	Himpunan data teks berstruktur multibahasa yang biasa digunakan untuk tugas ekstraksi sentimen lintas bahasa.	Pemanfaatan model *pretrained* XLM-RoBERTa dan RoBERTa yang di-fine-tune untuk klasifikasi bahasa kompleks.	Mengutilisasi model *pretrained transformer* untuk tugas analisis sentimen lintas bahasa.	XLM-RoBERTa menunjukkan ketangguhan dalam membedah struktur teks lintas bahasa meski utilitasnya belum diekspansi secara multimodal untuk mengekstrak anomali visual.
[18]	Dataset gambar standar berskala besar yang digunakan untuk tugas deteksi objek dan visi komputer.	Arsitektur ConvNeXt yang memanfaatkan *depthwise convolution* dan *layer normalization* untuk ekstraksi spasial.	Mengevaluasi ConvNeXt untuk tugas ekstraksi fitur visual tingkat lanjut.	Mampu mengekstraksi fitur spasial secara jauh lebih efisien dan akurat dibanding CNN klasik namun strukturnya membutuhkan optimasi *hyperparameter* yang cermat agar konvergen optimal.
[19]	Data ulasan produk fesyen berupa pasangan teks dan gambar yang bersumber dari platform Tokopedia	Fusi *feature-level* yang mengintegrasikan *text encoder* BERT dengan mekanisme atensi visual dari DeiT.	Menggabungkan representasi teks dan gambar untuk klasifikasi sentimen multimodal.	Sinergi model sukses memberikan pemahaman konteks ulasan yang kuat, akan tetapi masih bergantung pada gambar artifisial dan belum optimal untuk ejaan *noise* ekstrem.
[7]	Ratusan pangkalan data multi-domain berskala besar yang mencakup masukan teks, gambar, hingga audio.	Menerapkan *deep multimodal fusion* terpusat yang menggabungkan matriks representasi laten dari berbagai lapisan jaringan saraf.	Mengkaji dan mengembangkan teknik fusi data multimodal pada arsitektur *deep learning*.	Metode fusi hibrida terbukti menangkap relasi kontekstual yang lebih kaya dibanding unimodal, walau model fusi rentan bias jika distribusi kelas dataset tidak diseimbangkan (*undersampling*).

Author	**Dataset**	**Metode**	**Tujuan Penelitian**	**Hasil**
[20]	Beragam dataset gambar (*computer vision*) komersial.	Menyatukan keunggulan ekstraksi lokal CNN (ResNet) dengan pemahaman global dari arsitektur *attention* (DeiT/ViT).	Menyelidiki sinergi arsitektur *hybrid* untuk deteksi anomali spasial.	Membuktikan bahwa penggabungan model pemrosesan paralel sangat tangguh, namun menuntut manajemen VRAM (GPU) serta komputasi yang sangat ketat saat dilatih.
[21]	Himpunan observasi yang merupakan gabungan dari beberapa dataset depresi medis (*Combined Datasets*).	Melakukan optimasi *hyperparameter* otomatis menggunakan algoritma pencarian TPE (*Tree-structured Parzen Estimator*) pada *framework* Optuna.	Mengotomatisasi pencarian *hyperparameter* untuk mendongkrak performa klasifikasi.	Algoritma pencarian pada Optuna mempercepat konvergensi parameter terbaik secara efisien dan akurat, namun belum diterapkan secara spesifik pada optimasi lapisan fusi multimodal raksasa.
[15]	Objek tensor masukan umum berupa teks dan matriks gambar di dalam ekosistem PyTorch.	Mengimplementasikan algoritma *Explainable AI* bernama *Integrated Gradients* (IG) untuk mengkalkulasi integral gradien dari *baseline* ke input aktual.	Menyediakan transparansi prediksi dan atribusi fitur pada model *black-box* PyTorch.	*Integrated Gradients* sukses memetakan atribusi keputusan secara transparan ke token teks dan area piksel gambar, meski kalkulasinya menambah beban komputasi memori grafis saat ekstraksi diagnostik.

Berdasarkan tinjauan literatur yang telah dijabarkan sebelumnya dapat disimpulkan bahwa perkembangan teknologi analisis sentimen saat ini masih menyisakan celah penelitian yang mendasar. Beberapa studi sebelumnya [9] berhasil melatih model bahasa menggunakan 20.000 sampel teks campuran Banglish dengan capaian akurasi 69.50%. Riset lain [10] membuktikan keandalan model MuRIL dalam menangani kerumitan ejaan transliterasi lokal dengan tingkat akurasi mencapai 92.00%. Eksperimen lintas bahasa pada penelitian [17] turut menunjukkan ketangguhan arsitektur *transformer* global dalam mengekstrak polaritas sentimen dengan akurasi tertinggi 91.00%. Walaupun memiliki angka

performa yang mengesankan pendekatan tersebut murni beroperasi secara teks tunggal sehingga sangat rentan bias. Model dipastikan buta dan akan gagal memprediksi sentimen negatif jika keluhan konsumen hanya direpresentasikan melalui foto cacat produk tanpa penjelasan teks yang memadai.

Di studi lain transisi menuju arsitektur multimodal [19] berhasil mencapai akurasi tertinggi 93.49% namun masih memiliki limitasi, dimana metodologi yang digunakan bergantung pada penggunaan gambar buatan algoritma untuk mengisi kekosongan data visual ulasan produk fesyen. Pendekatan ini gagal memvalidasi kerusakan produk otentik dari pengguna dan terbukti belum diuji coba untuk bahasa campuran ekstrem. Literatur lain [7] dan [20] juga mengonfirmasi bahwa penggabungan arsitektur hibrida untuk deteksi anomali spasial memang jauh lebih tangguh. Namun pengembangannya menuntut optimasi komputasi yang sangat ketat dan keseimbangan distribusi kelas.

Lebih jauh lagi implementasi peningkatan performa melalui optimasi parameter otomatis dan validasi keputusan menggunakan model interpretasi belum diintegrasikan ke dalam arsitektur fusi skala masif. Studi [18] menegaskan bahwa pengekstraksi fitur visual modern membutuhkan penyesuaian parameter yang cermat untuk mendongkrak metrik evaluasi ke angka 77.79% . Pendekatan optimasi menggunakan algoritma pencarian terbukti mempercepat konvergensi parameter terbaik secara efisien dengan perolehan akurasi 93.27% [21]. Sementara itu transparansi keputusan model dapat dipetakan secara jelas menggunakan metode atribusi gradien [15]. Oleh karena itu terdapat urgensi yang mendesak untuk merancang sebuah arsitektur multimodal otentik yang memadukan model representasi bahasa tingkat lanjut dengan pengekstraksi fitur visual modern. Integrasi mekanisme optimasi otomatis dan transparansi prediksi wajib dikembangkan agar sistem yang dihasilkan tidak hanya akurat tetapi juga dapat diandalkan oleh industri *e-commerce*. Kombinasi solusi komprehensif inilah yang menjadi landasan utama bagi perancangan metodologi penelitian ini.

2.2 Tinjauan Teori

2.2.1 *E-commerce*

E-commerce adalah kegiatan transaksi komersial atas barang maupun jasa menggunakan jaringan internet [22]. Pemanfaatan teknologi digital pada sistem ini memfasilitasi proses pertukaran nilai antara pihak penjual dengan pihak pembeli tanpa mensyaratkan pertemuan fisik. Pelanggan memiliki kendali penuh untuk mengeksplorasi katalog produk lalu membuat pesanan dan menyelesaikan pembayaran hingga melacak pengiriman barang langsung melalui perangkat pintar mereka secara fleksibel. Karakteristik paling menonjol dari *e-commerce* terletak pada keberadaan platform daring berbasis situs web maupun aplikasi seluler yang bertindak sebagai etalase virtual bagi pengusaha untuk memajang produk mereka [23]. Ekosistem digital ini menjamin siklus transaksi berjalan jauh lebih cepat dan efisien dengan jangkauan pasar yang melampaui batas geografis perdagangan konvensional. Skala pemanfaatannya tidak lagi terbatas pada korporasi raksasa karena pelaku UMKM hingga pedagang individu kini sangat bergantung pada *e-commerce* untuk memasarkan dagangan mereka secara virtual [24].

Klasifikasi transaksi pada *e-commerce* terbagi menjadi beberapa model utama yaitu *Business to Consumer* (B2C) disusul Business to Business (B2B) dilanjutkan *Consumer to Consumer* (C2C) dan diakhiri Consumer to Business (C2B) [25]. Skema B2C menempati posisi paling dominan karena pihak produsen mendistribusikan barang langsung ke tangan pengguna akhir melalui perantara aplikasi belanja daring populer seperti Shopee atau Tokopedia maupun Bukalapak [8]. Akselerasi pertumbuhan *e-commerce* mencetak rekor fantastis semenjak era pandemi COVID-19 akibat penerapan pembatasan aktivitas fisik yang memaksa masyarakat untuk mengadopsi gaya hidup belanja virtual [25]. Lonjakan jumlah pengguna internet dipadukan dengan kemudahan akses ponsel pintar dan integrasi dompet elektronik ditambah keandalan infrastruktur logistik menjadi katalis utama yang melipatgandakan pertumbuhan sektor industri ini.

2.2.2 *E-commerce Reviews*

Ulasan pelanggan atau customer reviews pada platform *e-commerce* diakui sebagai sumber informasi fundamental yang merepresentasikan tingkat kepuasan serta pengalaman empiris konsumen terhadap suatu produk [8]. Seiring dengan peningkatan kapabilitas interaksi digital pada aplikasi belanja daring, format ulasan tersebut tidak lagi dibatasi pada elemen tekstual saja, melainkan telah diintegrasikan dengan fitur pengunggahan User-Generated Content (UGC) berupa gambar visual [26]. Lampiran gambar produk tersebut difungsikan secara khusus oleh pelanggan untuk menyertakan bukti fisik yang autentik, terutama ketika terjadi ketidaksesuaian, kerusakan, atau kecacatan pada barang yang diterima [19]. Kombinasi informasi antara opini teks dan bukti gambar ini pada akhirnya membentuk himpunan data multi-format tak terstruktur yang sangat besar, sehingga pengekstraksian wawasan kepuasan pelanggan melalui sistem komputasional otomatis menjadi sangat esensial bagi evaluasi kinerja bisnis operasional [27]. Selain itu, keberagaman bahasa yang digunakan dalam ulasan tersebut, termasuk penggunaan bahasa campuran atau *code-mixed*, menuntut adanya sistem analisis yang lebih cerdas untuk memahami konteks ulasan secara utuh [9].

2.2.3 *Sentiment Analysis*

Analisis sentimen merupakan bidang studi komputasional yang berfokus pada evaluasi opini dan emosi maupun sikap manusia terhadap suatu entitas [28]. Entitas tersebut dapat berupa produk maupun layanan jasa yang diekspresikan oleh pengguna melalui tulisan digital [28]. Tujuan utama dari proses ini adalah mengklasifikasikan teks ulasan ke dalam polaritas spesifik seperti positif atau negatif maupun netral. Proses komputasi ini bekerja dengan membedah makna di balik susunan kata yang tidak terstruktur pada ulasan pelanggan untuk memahami kecenderungan pandangan mereka [29].

Secara umum pendekatan analisis sentimen terbagi menjadi beberapa tingkatan evaluasi mulai dari tingkat dokumen hingga tingkat kalimat dan tingkat aspek [30]. Klasifikasi pada tingkat dokumen bertujuan menyimpulkan polaritas keseluruhan dari satu teks ulasan penuh tanpa memecahnya lebih lanjut. Evaluasi

pada tingkat kalimat berfokus pada penentuan sentimen spesifik dari setiap baris pernyataan penyusun paragraf. Sementara itu analisis tingkat aspek bekerja jauh lebih mendalam dengan mengekstraksi opini konsumen terhadap fitur spesifik dari suatu produk [30]. Pendekatan analisis mendalam inilah yang sering menjadi tantangan utama dalam riset pemrosesan bahasa alami.

2.2.4 *Code-mixed*

Penguntukan *code-mixed* language di ruang digital telah diidentifikasi sebagai salah satu tantangan teknis paling signifikan dalam domain pemrosesan bahasa alami [9]. Fenomena ini didefinisikan secara konseptual sebagai praktik penggabungan unit linguistik dari dua bahasa atau lebih secara bergantian dalam satu struktur kalimat yang sama [31]. Karakteristik utama dari teks bahasa campuran ini sering kali ditandai dengan pengabaian kaidah sintaksis baku serta tingginya variasi ejaan informal yang tidak terdaftar dalam kamus formal [32]. Munculnya kata-kata yang bersifat *Out-of-Vocabulary* (OOV) pada ulasan pelanggan menyebabkan proses ekstraksi fitur linguistik menjadi jauh lebih kompleks dibandingkan dengan pemrosesan bahasa tunggal [33].

Untuk mengatasi kompleksitas tersebut, diperlukan implementasi arsitektur text encoder tingkat lanjut yang mampu menangkap korelasi semantik antar bahasa secara mendalam [27]. Model berbasis *transformer* sering kali difungsikan untuk membedah opini pelanggan yang tersembunyi di balik struktur teks yang tidak teratur dan penuh dengan *noise*. Pemahaman terhadap pola *code-mixed* dianggap sangat krusial agar sistem klasifikasi sentimen tidak kehilangan konteks asli saat menghadapi ulasan dengan ambiguitas linguistik yang tinggi [34]. Melalui penggunaan pre-trained models yang bersifat multibahasa, representasi fitur tekstual dapat dihasilkan secara lebih akurat untuk menunjang performa analisis pada tahapan pemodelan selanjutnya.

2.2.5 *Multimodal*

Multimodal merupakan sebuah pendekatan dalam ranah kecerdasan buatan yang mengintegrasikan dan memproses informasi dari berbagai jenis modalitas data yang berbeda, seperti teks, citra, suara, atau video, secara simultan [6]. Struktur

utama dari sistem pembelajaran multimodal umumnya terdiri dari tiga tahapan utama, yaitu ekstraksi fitur independen (unimodal encoders), penyatuan fitur (feature fusion), dan lapisan klasifikasi akhir (classification head) [7]. Tahap fusi fitur sendiri secara struktural dapat dibagi menjadi early fusion (penggabungan di tingkat input dasar), late fusion (penggabungan di tingkat skor keputusan/prediksi), serta *feature-level fusion* atau hybrid fusion yang menyatukan representasi vektor laten dari masing-masing enkoder menggunakan teknik konkatenasi atau mekanisme atensi [35]. Melalui kombinasi interaksi antar-modalitas ini, model mampu menangkap hubungan kontekstual yang lebih kaya dan komprehensif dibandingkan hanya mengandalkan satu jenis sumber data tunggal (unimodal). Karakteristik struktural tersebut menjadikan analisis multimodal sangat efektif untuk menyelesaikan tugas-tugas kompleks, seperti mendeteksi sentimen ulasan *e-commerce* yang melibatkan teks keluhan sekaligus bukti foto produk [35].

2.2.6 *Class Imbalance*

Ketidakseimbangan kelas (class imbalance) merupakan suatu anomali dalam distribusi himpunan data di mana kuantitas observasi pada salah satu kelas secara signifikan melampaui jumlah observasi pada kelas lainnya [36]. Pada domain analisis sentimen ulasan *e-commerce*, ketimpangan ini umumnya terjadi karena konsumen cenderung jauh lebih sering mengunggah ulasan bergambar untuk mengekspresikan kepuasan (kelas positif) dibandingkan dengan keluhan atau kecacatan produk (kelas negatif) [37]. Apabila arsitektur *deep learning* dilatih secara langsung menggunakan distribusi data yang timpang tersebut, jaringan saraf akan rentan mengalami majority bias (bias mayoritas). Dalam kondisi ini, fungsi kerugian (loss function) model akan mendominasi pembaruan bobot untuk mengenali kelas mayoritas secara ekstrem, namun secara bersamaan kehilangan sensitivitasnya dalam mendeteksi fitur-fitur kritis pada kelas minoritas (ulasan negatif) yang justru menjadi tujuan utama evaluasi.

Tantangan terkait distribusi data mentah yang timpang dan penuh dengan derau (*noise*) ini merupakan masalah fundamental dalam pengembangan model kecerdasan buatan berbasis multimodal. Hal ini dipertegas oleh kajian

komprehensif pada pengembangan dataset raksasa LAION-5B [38]. LAION-5B merupakan himpunan data terbuka yang terdiri dari 5,85 miliar pasangan teks dan gambar yang diekstraksi dari arsip web publik (Common Crawl). Penelitian tersebut menyoroti bahwa pengumpulan data bervolume besar dari internet secara natural akan mewarisi bias sosial, representasi kelas yang tidak seimbang, serta kualitas korelasi teks-gambar yang sering kali tidak selaras (noisy). Studi sebelumnya memperingatkan bahwa pemanfaatan data mentah ini tanpa adanya mekanisme penyaringan atau intervensi kurasi yang ketat akan menyebabkan model mereproduksi bias tersebut, sehingga mendegradasi kualitas generalisasi model pada skenario dunia nyata [38].

2.2.7 *Text Validation*

Text validation merupakan salah satu tahapan awal dalam proses pembersihan data yang bertujuan memastikan bahwa data tekstual memenuhi kualitas minimum sebelum diproses lebih lanjut oleh model machine learning maupun deep learning [38]. Data teks yang dikumpulkan dari berbagai sumber sering kali mengandung karakter yang tidak relevan, simbol berlebihan, teks yang terlalu pendek, maupun kombinasi karakter yang tidak membentuk informasi semantik yang bermakna [31]. Kondisi tersebut dapat menurunkan kualitas representasi fitur serta meningkatkan tingkat noise pada proses pembelajaran model. Oleh karena itu, validasi teks dilakukan untuk mempertahankan hanya data yang dianggap mampu merepresentasikan informasi yang dibutuhkan dalam proses analisis.

Beberapa teknik text validation yang umum diterapkan meliputi pemeriksaan panjang minimum teks (*minimum text length*), validasi karakter menggunakan *regular expression* (regex), serta penyaringan terhadap karakter yang tidak sesuai dengan bahasa target [39]. Penggunaan batas minimum jumlah karakter bertujuan menghilangkan ulasan yang terlalu singkat sehingga tidak memiliki informasi sentimen yang memadai. Selain itu, *regular expression* banyak digunakan untuk memverifikasi keberadaan pola karakter tertentu, seperti huruf alfabet maupun kombinasi karakter yang sesuai dengan kebutuhan penelitian [40]. Melalui proses

validasi tersebut, kualitas data tekstual dapat ditingkatkan sehingga proses ekstraksi fitur pada tahapan berikutnya menjadi lebih optimal.

2.2.8 *Sentiment Labeling Based on Rating*

Pada berbagai penelitian analisis sentimen, label kelas sering diperoleh secara otomatis menggunakan nilai rating yang diberikan oleh pengguna [41]. Pendekatan ini dikenal sebagai *weak labeling*, yaitu proses pembentukan label berdasarkan informasi yang telah tersedia tanpa memerlukan anotasi manual. Penggunaan rating sebagai representasi sentimen didasarkan pada asumsi bahwa nilai rating yang tinggi menunjukkan sentimen positif, sedangkan nilai rating yang rendah merepresentasikan sentimen negatif [8]. Pendekatan tersebut banyak digunakan karena mampu menghasilkan dataset berukuran besar dengan proses pelabelan yang relatif cepat dan konsisten.

Meskipun demikian, penggunaan rating sebagai label sentimen memiliki beberapa keterbatasan. Salah satunya adalah keberadaan rating netral yang sering kali menimbulkan ambiguitas karena tidak secara jelas merepresentasikan polaritas positif maupun negatif [42]. Oleh sebab itu, pada penelitian klasifikasi sentimen biner, rating netral umumnya tidak digunakan sehingga model hanya mempelajari dua kelas sentimen yang memiliki batas keputusan lebih jelas. Pendekatan ini membantu meningkatkan konsistensi proses pelatihan serta mempermudah evaluasi performa model klasifikasi[42].

2.2.9 *Data Deduplication*

Data deduplication merupakan proses identifikasi dan penghapusan data yang memiliki informasi identik atau sangat mirip di dalam suatu dataset. Duplikasi data dapat muncul akibat kesalahan proses pengumpulan data, sinkronisasi antar sistem, maupun aktivitas pengguna yang menghasilkan entri berulang. Keberadaan data duplikat dapat menyebabkan distribusi data menjadi tidak representatif sehingga model cenderung mempelajari pola tertentu secara berlebihan (*over-representation*).

Secara umum, proses deduplikasi dilakukan menggunakan beberapa pendekatan, seperti pencocokan berdasarkan nilai atribut tertentu (*key-based matching*), pencocokan keseluruhan isi data (*exact duplicate*), maupun pencocokan berdasarkan tingkat kemiripan (*near duplicate detection*). Penghapusan data duplikat bertujuan menjaga kualitas dataset, mengurangi redundansi informasi, serta meningkatkan kemampuan generalisasi model ketika dihadapkan pada data yang belum pernah diproses sebelumnya.

2.2.10 *Threshold Calibration*

Threshold calibration merupakan proses penentuan nilai ambang (*threshold*) yang digunakan sebagai batas keputusan pada suatu proses klasifikasi maupun penyaringan data. Nilai threshold berperan dalam membedakan apakah suatu objek memenuhi kriteria tertentu berdasarkan hasil pengukuran yang diperoleh. Penentuan threshold yang kurang tepat dapat menyebabkan data berkualitas baik tereliminasi atau sebaliknya, data berkualitas rendah tetap dipertahankan sehingga memengaruhi performa model.

Secara umum, threshold dapat ditentukan menggunakan dua pendekatan, yaitu *fixed threshold* dan *data-driven threshold*. Pendekatan *fixed threshold* menggunakan nilai ambang yang telah ditentukan sebelumnya berdasarkan literatur atau eksperimen terdahulu. Sebaliknya, *data-driven threshold* menentukan nilai ambang berdasarkan karakteristik distribusi data, misalnya menggunakan nilai persentil (*percentile*) atau statistik tertentu. Pendekatan ini dinilai lebih adaptif karena mampu menyesuaikan batas keputusan terhadap karakteristik dataset yang digunakan.

2.2.11 *Feature-Level Fusion (Concatenation)*

Feature-level fusion merupakan salah satu pendekatan multimodal yang menggabungkan representasi fitur dari dua atau lebih modalitas sebelum proses klasifikasi dilakukan [7]. Berbeda dengan *decision-level fusion* yang mengombinasikan hasil prediksi dari masing-masing model, *feature-level fusion* mengintegrasikan informasi sejak tahap representasi fitur sehingga model dapat mempelajari hubungan antar modalitas secara langsung [43]. Pendekatan ini

banyak digunakan pada penelitian multimodal karena mampu menghasilkan representasi yang lebih kaya dibandingkan penggunaan satu modalitas saja.

Salah satu teknik *feature-level fusion* yang paling umum digunakan adalah *concatenation*, yaitu proses menggabungkan dua vektor fitur dengan menempatkannya secara berurutan menjadi satu vektor berdimensi lebih besar [7]. Vektor hasil concatenation kemudian digunakan sebagai masukan bagi lapisan klasifikasi untuk menghasilkan prediksi akhir [7]. Pendekatan ini relatif sederhana, tidak memerlukan parameter tambahan pada proses penggabungan fitur, serta mampu mempertahankan seluruh informasi yang diekstraksi dari masing-masing modalitas sehingga banyak diterapkan pada berbagai penelitian analisis sentimen multimodal.

2.2.12 *Metrik Evaluasi*

Metrik evaluasi berperan sebagai indikator krusial untuk menilai performa suatu model komputasional pada berbagai tugas seperti klasifikasi dan regresi sentimen [44]. Penguntukan metrik ini bertujuan untuk mengukur tingkat ketepatan prediksi yang dihasilkan model serta kemampuannya dalam melakukan generalisasi terhadap data baru yang belum pernah dilihat sebelumnya [8]. Pemilihan metrik yang tepat sangat menentukan validitas dari hasil komparasi antar arsitektur model *deep learning* yang sedang diuji dalam sebuah eksperimen analitik [36]. Dalam konteks analisis opini produk, metrik ini memberikan representasi numerik yang objektif mengenai sejauh mana model dapat membedakan polaritas emosi pelanggan dengan benar pada bahasa yang kompleks [27].

Implementasi evaluasi yang terstruktur memungkinkan peneliti untuk mengidentifikasi kekuatan maupun kelemahan spesifik dari algoritma gabungan yang diusulkan. Proses pengujian ini tidak hanya berfokus pada keberhasilan prediksi secara umum, tetapi juga menyoroti keandalan model saat dihadapkan pada himpunan data multi-modalitas yang tidak seimbang [39]. Oleh karena itu, penggunaan kombinasi beberapa metrik pengukuran sangat direkomendasikan untuk menghasilkan validasi performa yang komprehensif. Adapun beberapa

metrik evaluasi utama yang sering diterapkan dalam arsitektur klasifikasi *e-commerce* meliputi akurasi, presisi, recall, dan F1-Score [45].

1. Akurasi

   Akurasi merupakan salah satu indikator evaluasi yang digunakan untuk menilai tingkat ketepatan prediksi model secara keseluruhan. Metrik ini menunjukkan proporsi prediksi yang sesuai dengan label sebenarnya terhadap keseluruhan prediksi yang dihasilkan, mencakup prediksi benar pada kelas positif maupun negatif. Nilai akurasi yang tinggi mencerminkan kemampuan model dalam melakukan klasifikasi secara umum dengan baik pada data uji. Perhitungan akurasi pada penelitian ini mengacu pada Rumus 2.1

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN}$$

Rumus 2.1. Akurasi [46]

   Keterangan:

   a. *True Positive* (TP): Banyaknya data berlabel positif yang berhasil diprediksi secara tepat oleh model.
   b. *True Negative* (TN): Banyaknya data berlabel negatif yang diprediksi dengan benar oleh model.
   c. *False Positive* (FP): Banyaknya data berlabel negatif yang keliru diprediksi sebagai positif.
   d. *False Negative* (FN): Banyaknya data berlabel positif yang keliru diprediksi sebagai negatif

2. Presisi

   Presisi adalah metrik yang menilai ketepatan model dalam memprediksi kelas positif. Dengan kata lain, presisi menunjukkan proporsi prediksi positif yang benar dari seluruh prediksi positif yang dihasilkan model. Metrik ini menjadi sangat penting ketika kesalahan dalam memprediksi positif palsu (false positive) memiliki dampak yang signifikan pada penilaian kepuasan konsumen. Semakin tinggi nilai presisi, semakin sedikit kesalahan model dalam melabeli ulasan

negatif sebagai ulasan positif. Perhitungan presisi dapat dilakukan menggunakan Rumus 2.2 berikut.

$$Presisi = \frac{TP}{TP + FP}$$

Rumus 2.2. Presisi [46]

Keterangan:

a. *True Positive* (TP): Banyaknya data berlabel positif yang berhasil diprediksi secara tepat oleh model.
b. *False Positive* (FP): Banyaknya data berlabel negatif yang keliru diprediksi sebagai positif.

3. Recall

Recall adalah metrik yang mengukur kemampuan model dalam mendeteksi seluruh contoh positif yang ada pada dataset. Dengan kata lain, recall menunjukkan proporsi kasus positif yang berhasil diidentifikasi dengan benar oleh model dari total data yang seharusnya. Metrik ini menjadi krusial ketika kesalahan dalam melewatkan kasus positif (false negative) memiliki konsekuensi yang besar bagi sistem evaluasi. Dalam kasus sentimen, recall yang tinggi berarti model tidak melewatkan banyak ulasan penting. Perhitungan recall dilakukan menggunakan Rumus 2.3.

$$Recall = \frac{TP}{TP + FN}$$

Rumus 2.3. Recall [46]

Keterangan:

a. *True Positive* (TP): Banyaknya data berlabel positif yang berhasil diprediksi secara tepat oleh model.
b. *False Negative* (FN): Banyaknya data berlabel positif yang keliru diprediksi sebagai negatif

4. F1-Score

F1-Score merupakan metrik yang menggabungkan nilai presisi dan recall dalam satu ukuran harmonis. Metrik ini memberikan penilaian yang seimbang antara presisi dan recall, sehingga sangat berguna ketika terdapat ketidakseimbangan antara keduanya di dalam distribusi kelas. Mengingat data ulasan gambar sering kali memiliki jumlah kelas yang tidak seimbang (class imbalance), penggunaan akurasi saja tidaklah cukup. F1-Score sering digunakan untuk menilai kinerja model secara keseluruhan ketika baik presisi maupun recall sama-sama penting untuk diperhatikan. Perhitungan F1-Score dilakukan menggunakan Rumus 2.4

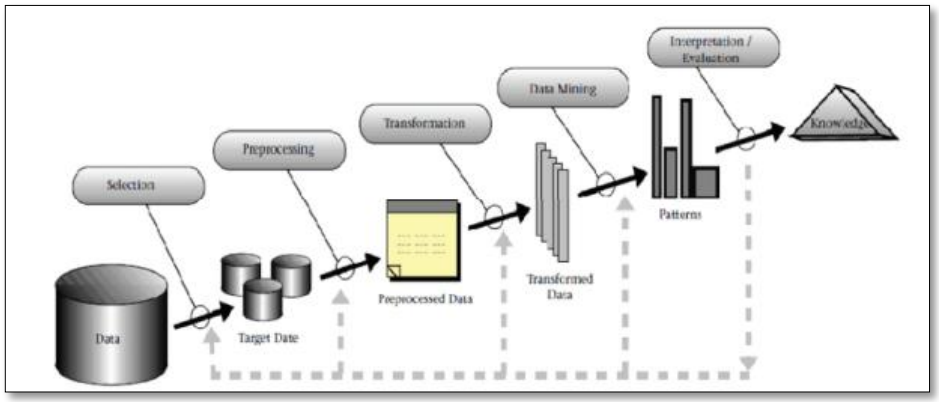
$$F1 - Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall}$$

Rumus 2.4. F1-Score [46]

2.3 Algoritma dan *Framework*

2.3.1 *Knowledge Discovery in Database* (KDD)

Knowledge Discovery in Databases (KDD) merupakan proses sistematis untuk mengekstraksi pengetahuan yang bernilai dari kumpulan data dalam jumlah besar. Konsep KDD diperkenalkan oleh Fayyad et al. sebagai suatu rangkaian aktivitas yang mencakup pemilihan data, pembersihan data, transformasi data, proses data *mining*, serta interpretasi hasil yang diperoleh [47]. Tujuan utama KDD adalah mengubah data mentah menjadi informasi yang dapat digunakan untuk mendukung pengambilan keputusan [48].



Gambar 2.1 [FA2]Diagram Tahapan KDD

Sumber: [49]

KDD terdiri atas lima tahapan utama, yaitu *Selection*, preprocessing, *transformation*, data *mining*, dan interpretation/*Evaluation [49]*.

1. *Selection*

   Langkah awalan ini berfokus pada penentuan data target dari sekumpulan pangkalan data operasional mentah [48]. Peneliti harus mengidentifikasi atribut variabel yang paling relevan dengan tujuan analitik sebelum memulai proses komputasi. Hasil ekstraksi dari tahap ini akan menghasilkan himpunan data spesifik yang menjadi fokus utama untuk diproses pada tahapan pembersihan tanpa perlu membebani memori sistem dengan pemrosesan data yang tidak berguna [47].

2. *Pre-processing*

   Himpunan data terpilih pada umumnya masih mengandung banyak derau atau nilai kosong yang berpotensi merusak keandalan model prediksi [47]. Proses pembersihan ini bertujuan mengeliminasi berbagai anomali tersebut supaya kualitas landasan data menjadi jauh lebih baik dan terstruktur. Teknik penanganannya mencakup pengisian nilai atribut kosong dan penghapusan anomali pencilan atau pembuangan baris data duplikat agar algoritma dapat memetakan pola dengan sangat optimal pada tahap selanjutnya [19].

3. *Transformation*

   Landasan data bersih selanjutnya diubah ke dalam representasi matematis yang paling ideal untuk dieksekusi oleh mesin komputasi [47]. Langkah modifikasi ini mencakup normalisasi rentang nilai numerik atau reduksi dimensi atau penggabungan beberapa atribut menjadi satu komponen fitur baru yang lebih informatif. Transformasi arsitektur data ini amat vital untuk melipatgandakan tingkat efisiensi algoritma pada tahap penambangan pengetahuan [47].

4. Modeling

   Penambangan data merupakan tahapan inti fungsional dari keseluruhan rangkaian proses KDD [47]. Berbagai metode algoritmik canggih diterapkan secara simultan pada tahap ini untuk mengekstraksi struktur pola laten di dalam pangkalan data yang telah ditransformasi. Pendekatan komputasi yang

diaplikasikan wajib disesuaikan dengan tujuan utama penelitian seperti analisis klaster atau klasifikasi sentimen maupun deteksi anomali menggunakan teknik pemelajaran mesin mutakhir [48].

5. *Evaluation*

   Tahap *Evaluation* and Interpretation merupakan proses mengevaluasi hasil yang diperoleh dari tahap *data mining* untuk memastikan bahwa pola atau model yang dihasilkan memiliki kualitas yang baik dan sesuai dengan tujuan analisis [46]. Evaluasi umumnya dilakukan menggunakan berbagai metrik sesuai jenis permasalahan, seperti akurasi, presisi, *recall*, *F1-score*, maupun metrik evaluasi lainnya [46]. Selain mengukur performa model, tahap ini juga mencakup proses interpretasi terhadap hasil yang diperoleh sehingga pengetahuan yang dihasilkan dapat dipahami dan dimanfaatkan dalam proses pengambilan keputusan. Dengan demikian, tahap evaluasi tidak hanya berfungsi sebagai pengukuran performa, tetapi juga memastikan bahwa informasi yang diekstraksi dari data memiliki nilai praktis dan dapat dipertanggungjawabkan.

2.3.2 *OpenCLIP*

OpenCLIP merupakan implementasi terbuka dari Contrastive Language–Image Pretraining (CLIP) yang dikembangkan untuk mempelajari representasi bersama antara data teks dan gambar. Model ini menggunakan pendekatan *contrastive learning*, yaitu proses pembelajaran yang bertujuan mendekatkan representasi vektor dari pasangan teks dan gambar yang saling berkaitan, serta menjauhkan representasi pasangan yang tidak memiliki hubungan semantik. Melalui mekanisme tersebut, OpenCLIP mampu menghasilkan ruang representasi (*embedding space*) yang memungkinkan data dari dua modalitas berbeda dibandingkan secara langsung.

Dalam proses pencocokan pasangan teks dan gambar, OpenCLIP menghasilkan embedding untuk masing-masing modalitas menggunakan encoder yang berbeda, kemudian mengukur tingkat kemiripannya menggunakan *cosine similarity*. Semakin tinggi nilai cosine similarity, semakin besar kemungkinan

bahwa gambar dan teks menggambarkan konteks yang sama. Oleh karena itu, OpenCLIP banyak dimanfaatkan pada berbagai tugas multimodal seperti image retrieval, image-text matching, visual question answering, serta pemilihan gambar yang paling relevan terhadap suatu deskripsi tekstual.

Tabel 2.2 *Pseudocode* OpenCLIP

No	Tahapan
1	Input pasangan teks ulasan (text) dan kumpulan gambar kandidat (candidate images).
2	Lakukan preprocessing pada teks menggunakan tokenizer bawaan OpenCLIP.
3	Lakukan preprocessing pada setiap gambar menggunakan transformasi citra bawaan OpenCLIP.
4	Masukkan teks ke dalam Text Encoder untuk memperoleh representasi embedding teks (text embedding).
5	Masukkan setiap gambar ke dalam Image Encoder untuk memperoleh representasi embedding gambar (image embedding).
6	Normalisasi seluruh embedding menggunakan normalisasi L2.
7	Hitung nilai Cosine Similarity antara embedding teks dan embedding masing-masing gambar.
8	Urutkan seluruh gambar berdasarkan nilai similarity dari yang tertinggi hingga terendah.
9	Pilih gambar dengan nilai similarity tertinggi sebagai gambar yang paling relevan terhadap teks ulasan.
10	Output berupa pasangan teks–gambar dengan tingkat relevansi tertinggi.

2.3.3 Optuna

Diperkenalkan pertama kali oleh Akiba et al., Optuna merupakan sebuah *framework* yang dirancang secara khusus untuk mengotomatisasi pencarian konfigurasi *hyperparameter* terbaik pada arsitektur machine learning maupun *deep learning* [14]. Keunggulan utama dari kerangka kerja ini terletak pada tingkat efisiensi proses optimasinya, di mana algoritma pencarian yang diusung mampu beroperasi secara adaptif sehingga dapat secara signifikan meminimalkan jumlah percobaan yang dibutuhkan untuk menemukan kombinasi parameter paling optimal. Secara prosedural, alur kerja dan mekanisme komputasi yang dijalankan oleh Optuna dalam mengevaluasi model dapat dijabarkan melalui *pseudocode* pada Tabel 2.2 berikut.

Tabel 2.3 *Pseudocode* Optuna

No	Tahapan
1	Input himpunan data latih (*Train Set*) dan data validasi (*Validation Set*).
2	Buat objek Study dan tetapkan arah optimasi target (contoh: direction = 'maximize' untuk metrik *F1-Score*). Tetapkan algoritma *Tree-structured Parzen Estimator* (TPE) sebagai Sampler.
3	Buat fungsi objektif *objective(trial)* yang akan dieksekusi secara berulang.
4	Di dalam fungsi objektif, definisikan batas pencarian parameter menggunakan objek trial. - *Contoh:* lr = trial.suggest_loguniform('learning_rate', 1e-5, 1e-3)
5	Bangun arsitektur *model* dan terapkan parameter yang disarankan oleh iterasi trial saat ini ke dalam model serta *optimizer*.

Berbeda dengan metode tradisional seperti Grid Search dan Random Search, Optuna menggunakan pendekatan berbasis optimasi Bayesian yang lebih adaptif terhadap hasil percobaan sebelumnya [50]. Secara bawaan, Optuna menerapkan algoritma Tree-structured Parzen Estimator (TPE) untuk menentukan kombinasi *hyperparameter* yang akan dievaluasi pada trial berikutnya. Pendekatan ini memungkinkan proses pencarian lebih terarah dibandingkan pencarian acak karena mempertimbangkan performa dari trial yang telah dilakukan sebelumnya [21].

Struktur utama Optuna terdiri atas beberapa komponen, yaitu Study, Trial, Sampler, dan Pruner [14]. Study merepresentasikan keseluruhan proses optimasi yang dijalankan. Trial merupakan satu kali eksperimen dengan kombinasi *hyperparameter* tertentu. Sampler bertugas memilih kombinasi *hyperparameter* yang akan diuji berdasarkan strategi optimasi yang digunakan. Sementara itu, Pruner berfungsi menghentikan trial yang diperkirakan tidak akan menghasilkan performa yang baik sehingga penggunaan sumber daya komputasi dapat dioptimalkan [14].

Sejumlah penelitian menunjukkan bahwa Optuna mampu menghasilkan performa model yang lebih baik dibandingkan konfigurasi *hyperparameter* yang ditentukan secara manual [51]. Selain meningkatkan akurasi model, penggunaan Optuna juga dapat mengurangi waktu pencarian *hyperparameter* karena proses optimasi dilakukan secara otomatis dan adaptif berdasarkan hasil evaluasi setiap trial. Untuk alasan tersebut, Optuna telah banyak digunakan pada penelitian modern

yang melibatkan model *deep learning* dengan jumlah *hyperparameter* yang besar dan kompleks.

2.3.4 *XLM-RoBERTa*

XLM-RoBERTa (*Cross-lingual Language Model*–RoBERTa) merupakan model bahasa multibahasa berbasis arsitektur *Transformer* yang dikembangkan oleh Facebook AI *Research* sebagai pengembangan dari RoBERTa untuk mendukung *cross-lingual understanding* [52]. Model ini dilatih menggunakan lebih dari dua terabyte korpus multibahasa yang berasal dari *CommonCrawl* dan mencakup 100 bahasa, sehingga mampu mempelajari representasi semantik dari berbagai bahasa secara bersamaan tanpa memerlukan proses penerjemahan [17]. Berbeda dengan multilingual BERT yang menggunakan kombinasi beberapa objektif pelatihan, XLM-RoBERTa hanya mengadopsi metode *Masked Language Modeling* (MLM) dengan strategi pelatihan yang lebih besar, jumlah data yang lebih banyak, serta optimisasi yang lebih baik. Pendekatan tersebut memungkinkan model menghasilkan representasi kontekstual yang lebih kaya sehingga mampu memahami hubungan antar kata berdasarkan konteks kalimat, bukan hanya berdasarkan arti leksikalnya [53].

Kemampuan utama XLM-RoBERTa terletak pada representasi lintas bahasa, yaitu kemampuan memetakan kata maupun kalimat dari berbagai bahasa ke dalam *embedding space* yang sama [54]. Dengan mekanisme tersebut, kata-kata yang memiliki makna serupa meskipun berasal dari bahasa yang berbeda akan memiliki representasi vektor yang berdekatan. Karakteristik ini membuat XLM-RoBERTa banyak digunakan pada berbagai tugas *Natural Language Processing* (NLP), seperti klasifikasi teks, analisis sentimen, named entity recognition, question answering, hingga *machine translation* [17]. Selain itu, kemampuan memahami teks multibahasa menjadikan model ini efektif dalam menangani data *code-mixed* maupun code-switching, yaitu kondisi ketika lebih dari satu bahasa digunakan secara bersamaan dalam satu kalimat [17].

Secara komputasional, alur kerja XLM-RoBERTa dalam memproses teks mentah menjadi vektor fitur yang siap diklasifikasikan dapat dijabarkan melalui *pseudocode* pada Tabel 2.3:

Tabel 2.4 Pseudocode XLM-RoBERTa

No	Tahapan
1	Menerima deretan string teks ulasan pelanggan yang telah melewati tahap prapemrosesan pembersihan awal.
2	Memecah teks string menjadi unit-unit sub-kata (token) menggunakan *AutoTokenizer* bawaan XLM-RoBERTa, seperti mengonversi teks menjadi representasi numerik (*input_ids*) dan membuat *attention_mask* bernilai 1 untuk token teks asli dan bernilai 0 untuk token *padding*
3	Menyelaraskan panjang deretan token numerik pada setiap ulasan ke batas maksimal seragam (misalnya: max_length = 128 token).
4	Mengonversi token matriks diskrit (input_ids) ke dalam *dense vector space*.
5	Meneruskan vektor input melewati lapisan *transformer* secara berurutan. Mekanisme *attention* menghitung bobot relevansi antara setiap token di dalam kalimat untuk menangkap makna kontekstual yang utuh dari urutan tata bahasa campuran.
6	Mengambil fitur matriks keluaran dari *last hidden state*. Secara spesifik, vektor dari token awal <s> (ekuivalen dengan token [CLS] pada BERT standar) diekstrak karena menyimpan agregasi representasi semantik dari keseluruhan ulasan teks tersebut.
7	Menghasilkan vektor fitur tekstual satu dimensi (umumnya berukuran 768 dimensi) yang merepresentasikan informasi sentimen kalimat. Vektor ini siap untuk diteruskan ke lapisan *feature-level fusion* bersama dengan vektor gambar.

a. Prinsip Matematis

Proses pelatihan XLM-RoBERTa menggunakan pendekatan *Masked Language Modeling* (MLM), yaitu sebagian token pada suatu kalimat disembunyikan (*masked*) secara acak, kemudian model dilatih untuk memprediksi token asli berdasarkan konteks token-token lainnya. Berbeda dengan model autoregresif yang hanya memperhatikan token sebelumnya, MLM memungkinkan *Transformer* memanfaatkan informasi dari kedua arah (*bidirectional context*) sehingga hubungan antar kata dapat dipelajari secara lebih komprehensif. Tujuan optimasi model adalah memaksimalkan probabilitas token asli pada posisi yang disembunyikan melalui fungsi objektif sebagai berikut.

$$\mathcal{L}_{MLM} = -\sum_{i \in M} \quad log\ P(x_i \mid x \backslash M)$$

Rumus 2.5. *Masked language modeling* [17]

Keterangan

a. M merupakan himpunan token yang disembunyikan (*masked tokens*),
b. $x_i$adalah token asli yang harus diprediksi,
c. $x_{\setminus M}$merupakan seluruh token yang tidak disembunyikan,
d. $P(x_i \mid x_{\setminus M})$adalah probabilitas prediksi token pada posisi ke-i.

Untuk memperoleh representasi kontekstual setiap token, XLM-RoBERTa memanfaatkan mekanisme *Self-Attention* yang menghitung hubungan antar seluruh token dalam satu kalimat secara bersamaan. Setiap token terlebih dahulu diproyeksikan menjadi tiga representasi, yaitu *Query* (Q), *Key* (K), dan *Value* (V). Selanjutnya bobot perhatian dihitung menggunakan persamaan *Scaled Dot-Product Attention* berikut.

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

Rumus 2.6. $Scaled\ Dot - Product\ Attention$ [52]

dengan $d_k$menyatakan dimensi vektor *key*. Pembagian terhadap $\sqrt{d_k}$bertujuan menjaga stabilitas nilai gradien ketika dimensi representasi semakin besar. Melalui mekanisme tersebut, model mampu menentukan token mana yang paling relevan ketika membentuk representasi suatu kata berdasarkan konteks keseluruhan kalimat.

b. Arsitektur

Arsitektur XLM-RoBERTa dibangun berdasarkan *Transformer Encoder* yang terdiri atas beberapa lapisan encoder identik. Setiap *encoder* mengombinasikan mekanisme *Multi-Head Self-Attention* (MHSA) dan *Feed Forward Network* (FFN) yang dilengkapi dengan *Residual Connection* serta *Layer Normalization* untuk menghasilkan representasi kontekstual yang stabil [17]. Proses pemrosesan diawali dengan tokenisasi menggunakan *SentencePiece*, kemudian setiap token diubah menjadi embedding dan diproses secara bertahap melalui seluruh lapisan encoder untuk mempelajari hubungan semantik antar kata. Representasi kontekstual yang dihasilkan

selanjutnya dimanfaatkan sebagai fitur pada berbagai tugas *Natural Language Processing* (NLP), seperti klasifikasi teks, analisis sentimen, maupun *named entity recognition [17]*.

2.3.5 *MuRIL (Multilingual Representations for Indian Languages)*

MuRIL (*Multilingual Representations for Indian Languages*) merupakan model bahasa multibahasa berbasis arsitektur *Transformer* yang dikembangkan oleh Google Research untuk meningkatkan pemahaman bahasa-bahasa yang digunakan di India [55]. Model ini dirancang karena sebagian besar model multibahasa sebelumnya masih memiliki keterbatasan dalam memahami karakteristik bahasa lokal India yang memiliki variasi penulisan, transliterasi, maupun penggunaan bahasa campuran (*code-mixed text*) [55]. Untuk mengatasi permasalahan tersebut, MuRIL dilatih menggunakan korpus multibahasa yang mencakup 16 bahasa India, bahasa Inggris, serta pasangan kalimat hasil transliterasi dan terjemahan sehingga mampu mempelajari hubungan semantik antarbahasa secara lebih efektif [55]. Dengan pendekatan tersebut, MuRIL menghasilkan representasi kontekstual yang lebih baik untuk berbagai tugas *Natural Language Processing* (NLP), seperti klasifikasi teks, analisis sentimen, *named entity recognition*, serta *question answering* pada lingkungan multibahasa [56].

Salah satu karakteristik utama MuRIL adalah kemampuannya memahami hubungan antara teks asli, teks hasil transliterasi, dan teks hasil terjemahan selama proses *pre-training[10]*. Pendekatan ini memungkinkan model menghasilkan representasi yang lebih konsisten terhadap kata atau kalimat yang memiliki makna sama meskipun ditulis menggunakan bahasa maupun sistem penulisan yang berbeda. Oleh karena itu, MuRIL banyak digunakan pada berbagai penelitian yang melibatkan data multibahasa maupun *code-mixed[56]*, khususnya pada bahasa-bahasa dengan sumber daya (*low-resource languages*) yang sebelumnya kurang terwakili pada model bahasa multibahasa berskala besar.

Secara komputasional, alur kerja MuRIL dalam memproses teks mentah menjadi vektor fitur yang siap diklasifikasikan dapat dijabarkan melalui *pseudocode* pada Tabel 2.4:

Tabel 2.5 *Pseudocode* MuRIL [52]

No	Tahapan
1	Menerima deretan string teks ulasan pelanggan yang telah melewati tahap prapemrosesan pembersihan awal.
2	Memecah teks string menjadi unit-unit sub-kata (token) menggunakan *AutoTokenizer* bawaan MuRIL, seperti mengonversi teks menjadi representasi numerik (*input_ids*) dan membuat *attention_mask* bernilai 1 untuk token teks asli dan bernilai 0 untuk token *padding*
3	Menyelaraskan panjang deretan token numerik pada setiap ulasan ke batas maksimal seragam (misalnya: max_length = 128 token).
4	Mengonversi token matriks diskrit (input_ids) ke dalam *dense vector space*.
5	Meneruskan vektor input melewati lapisan *transformer* secara berurutan. Mekanisme *attention* menghitung bobot relevansi antara setiap token di dalam kalimat untuk menangkap makna kontekstual yang utuh dari urutan tata bahasa campuran.
6	Mengambil fitur matriks keluaran dari *last hidden state*. Secara spesifik, vektor dari token awal <s> (ekuivalen dengan token [CLS] pada BERT standar) diekstrak karena menyimpan agregasi representasi semantik dari keseluruhan ulasan teks tersebut.
7	Menghasilkan vektor fitur tekstual satu dimensi (umumnya berukuran 768 dimensi) yang merepresentasikan informasi sentimen kalimat. Vektor ini siap untuk diteruskan ke lapisan *feature-level fusion* bersama dengan vektor gambar.

a. Prinsip Matematis

Seperti model BERT pada umumnya, MuRIL memanfaatkan Masked Language Modeling (MLM) sebagai salah satu objektif utama selama proses pelatihan. Pada metode ini, sebagian token dalam suatu kalimat disembunyikan (masked) secara acak, kemudian model dilatih untuk memprediksi token asli berdasarkan konteks token yang tersisa. Fungsi objektif MLM dapat dinyatakan dengan Rumus 2.7 berikut.

$$\mathcal{L}_{MLM} = -\sum_{i \in M} \quad log\ P(x_i \mid x \backslash M)$$

Rumus 2.7. *Masked Language Modeling* [52]

Keterangan

a. M merupakan himpunan token yang disembunyikan (*masked tokens*),
b. $x_i$adalah token asli yang harus diprediksi,
c. $x_{\backslash M}$merupakan seluruh token yang tidak disembunyikan,
d. $P(x_i \mid x_{\backslash M})$adalah probabilitas prediksi token pada posisi ke-i.

Untuk memperoleh representasi kontekstual setiap token, XLM-RoBERTa memanfaatkan mekanisme *Self-Attention* yang menghitung hubungan antar seluruh token dalam satu kalimat secara bersamaan. Setiap token terlebih dahulu diproyeksikan menjadi tiga representasi, yaitu *Query* (Q), *Key* (K), dan *Value* (V). Selanjutnya bobot perhatian dihitung menggunakan Rumus 2.8 berikut.

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

Rumus 2.8. $Scaled\ Dot - Product\ Attention$ [52]

dengan d_k menyatakan dimensi vektor *key*. Pembagian terhadap $\sqrt{d_k}$ bertujuan menjaga stabilitas nilai gradien ketika dimensi representasi semakin besar. Melalui mekanisme tersebut, model mampu menentukan token mana yang paling relevan ketika membentuk representasi suatu kata berdasarkan konteks keseluruhan kalimat.

b. Arsitektur

Arsitektur XLM-RoBERTa dibangun berdasarkan *Transformer Encoder* yang terdiri atas beberapa lapisan encoder identik. Setiap *encoder* mengombinasikan mekanisme *Multi-Head Self-Attention* (MHSA) dan *Feed Forward Network* (FFN) yang dilengkapi dengan *Residual Connection* serta *Layer Normalization* untuk menghasilkan representasi kontekstual yang stabil [57]. Proses pemrosesan diawali dengan tokenisasi menggunakan *SentencePiece*, kemudian setiap token diubah menjadi embedding dan diproses secara bertahap melalui seluruh lapisan encoder untuk mempelajari hubungan semantik antar kata. Representasi kontekstual yang dihasilkan selanjutnya dimanfaatkan sebagai fitur pada berbagai tugas *Natural Language Processing* (NLP), seperti klasifikasi teks, analisis sentimen, maupun *named entity recognition [55].*

2.3.6 *Residual Network (ResNet-18)*

Residual Network atau ResNet merupakan arsitektur jaringan saraf konvolusional tingkat lanjut yang dirancang untuk mengatasi fenomena *vanishing gradient* pada jaringan saraf yang sangat dalam [57]. Arsitektur ini memperkenalkan konsep *residual learning* yang memungkinkan model untuk

mempelajari pemetaan residu alih-alih mencoba mempelajari pemetaan dasar secara langsung. ResNet-18 secara spesifik terdiri dari delapan belas lapisan pemrosesan yang mencakup lapisan konvolusi, *batch normalization*, dan fungsi aktivasi yang disusun secara sistematis. Penguntukan lapisan yang dalam ini bertujuan untuk mengekstraksi fitur visual yang lebih abstrak dan kompleks dari sebuah gambar ulasan produk [11]. Keunggulan utama dari model ini terletak pada stabilitasnya saat proses pelatihan meskipun jumlah parameternya cukup besar untuk ukuran model konvolusi klasik.

1. *Residual Block dan Skip Connection*

   Komponen paling krusial dalam ResNet adalah residual block yang dilengkapi dengan mekanisme skip connection atau koneksi lewati [42]. Mekanisme ini bekerja dengan melewatkan informasi dari satu lapisan langsung ke lapisan berikutnya tanpa melewati operasi konvolusi di antaranya. Hal ini memastikan bahwa gradien perhitungan tetap dapat mengalir dengan baik ke lapisan-lapisan awal selama fase *backpropagation*. Secara matematis, keluaran dari sebuah blok residual didefinisikan sebagai penjumlahan antara hasil pemetaan konvolusi dengan masukan aslinya.

Tabel 2.6 Pseudocode *Residual Block dan Skip Connection [42]*

No	Tahapan
1	Input: Ambil nilai masukan x dari peta fitur yang dihasilkan oleh lapisan konvolusi.
2	Periksa nilai masukan: Evaluasi apakah nilai x berada di atas angka nol atau tidak.
3	Penerapan ReLU: Jika $x > 0$, maka hasil $f(x) = x$. Jika $x \leq 0$, maka hasil $f(x) = 0$.
4	Output: Simpan hasil evaluasi sebagai representasi fitur non-linear baru.
5	Ulangi langkah 2 hingga 4 untuk setiap piksel elemen di dalam peta fitur tersebut
6	Akhiri proses setelah seluruh elemen matriks selesai divalidasi oleh fungsi aktivasi

2. Arsitektur ResNet-18

   Secara struktural, ResNet-18 diawali dengan lapisan konvolusi tunggal berukuran tujuh kali tujuh piksel dengan nilai *stride* dua yang diikuti oleh lapisan *max pooling*. Setelah itu, jaringan disusun oleh delapan buah blok residual yang masing-masing berisi dua lapisan konvolusi berukuran tiga kali tiga piksel. Pada bagian akhir jaringan, diterapkan lapisan *Global Average Pooling* untuk mengubah peta fitur menjadi vektor tunggal sebelum diproses

oleh lapisan fully connected. Lapisan terakhir ini bertanggung jawab untuk menghasilkan distribusi probabilitas yang menentukan apakah sebuah gambar ulasan termasuk dalam kategori sentimen positif atau negatif.

Penelitian sebelumnya mencatat bahwa optimalisasi pada arsitektur berbasis residual mampu memberikan efisiensi komputasi yang sangat konsisten pada berbagai dataset gambar [20]. Model konvolusional dengan koneksi lewati terbukti mampu mempertahankan akurasi klasifikasi yang stabil saat dihadapkan pada variasi data ulasan yang memiliki kualitas visual beragam. ResNet-18 diposisikan sebagai variabel bebas utama yang mewakili metode pemrosesan fitur lokal berbasis konvolusi dalam kerangka komparasi ini. Alur pemikirannya menghubungkan ketangguhan mekanisme *skip connection* dengan kemampuan model dalam mendeteksi detail kerusakan produk sebagai indikator sentimen negatif.

2.3.7 *Data-efficient Image Transformers (DeiT)*

Data-efficient Image Transformers atau DeiT merupakan adaptasi inovatif dari teknologi mekanisme yang dirancang khusus untuk ranah *computer vision* tingkat lanjut. Berbeda dengan pendekatan konvolusi konvensional yang memproses gambar secara bertahap melalui jendela lokal, DeiT memindai keseluruhan gambar masukan secara komprehensif dalam satu waktu komputasi. Arsitektur ini membuktikan bahwa ketergantungan model berbasis *Transformer* terhadap kumpulan data pelatihan yang berskala besar dapat diatasi melalui strategi penyulingan pengetahuan secara terarah. Mekanisme ini dirancang sedemikian rupa agar mampu mencapai akurasi deteksi maksimal meskipun hanya dilatih menggunakan pangkalan data yang jumlahnya jauh lebih terbatas. Kemampuan untuk memahami konteks spasial secara global sejak lapisan komputasi pertama menjadikan model ini sangat tangguh dalam menangkap detail anomali yang tersebar di berbagai sudut gambar. Proses pemetaan visual pada arsitektur ini bergantung pada tiga komponen operasional utama berikut:

1. *Patch Embedding*

Langkah paling awal dalam arsitektur ini adalah memecah gambar dua dimensi menjadi serangkaian petak atau patches yang tidak saling tumpang tindih. Setiap petak kemudian diratakan menjadi vektor satu dimensi dan diproyeksikan ke dalam ruang dimensi laten agar menyerupai representasi urutan kata pada pemrosesan bahasa alami. Model juga menambahkan sematan posisi (positional embedding) pada setiap vektor agar jaringan saraf tetap dapat mengenali urutan dan lokasi asli petak tersebut di dalam gambar utuh. Secara matematis, jumlah petak yang dihasilkan dari sebuah gambar dihitung menggunakan persamaan pada Rumus 2.9.

$$N = \frac{H \times W}{P^2}$$

Rumus 2.9. $Scaled\ Dot - Product\ Attention$ [12]

Keterangan:

a. N: Total jumlah petak (*patches*) yang dihasilkan.

b. $H \times W$: Resolusi tinggi dan lebar dari gambar masukan asli.

c. P: Ukuran resolusi sisi petak yang telah ditentukan (misalnya 16 piksel).

Tabel 2.7 *Pseudocode Patch Embedding [12]*

No	Tahapan
1	Menerima gambar masukan utuh berukuran panjang dan lebar tertentu beserta jumlah saluran warnanya.
2	Potong gambar tersebut menjadi sekumpulan matriks bujur sangkar berukuran enam belas kali enam belas piksel.
3	Ratakan setiap matriks petak tersebut menjadi sebuah urutan vektor berdimensi satu.
4	Kalikan urutan vektor tersebut dengan matriks proyeksi linier untuk menghasilkan sematan dimensi laten.
5	Tambahkan nilai sematan posisi spasial ke dalam setiap vektor agar model mengenali koordinat asli petak.
6	Masukkan kumpulan vektor yang telah merepresentasikan potongan gambar ini ke dalam lapisan blok *Transformer*.

2. *Multi-Head Self-Attention (MHSA)*

Mekanisme ini merupakan inti perhitungan dari arsitektur *Transformer* yang bertugas menemukan korelasi tingkat kepentingan antar setiap petak gambar. Proses ini melibatkan tiga matriks utama yaitu *Query*, *Key*, dan *Value* yang dihitung dari perkalian masukan dengan bobot lapisan linier. Atensi mandiri

mengevaluasi seberapa besar fokus yang harus diberikan oleh suatu petak terhadap petak lainnya saat memproses informasi visual tertentu. Perhitungan komputasi untuk mendapatkan bobot atensi ini dirumuskan secara aljabar pada Rumus 2.10.

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

Rumus 2.10. $Scaled\ Dot - Product\ Attention$ [12]

Keterangan:

a. Q: Matriks *Query* yang mewakili representasi elemen yang sedang mencari informasi.
b. K^T: Matriks *Key* transposisi yang mewakili kunci informasi dari elemen lain.
c. d_k: Dimensi dari vektor *Key* yang berfungsi sebagai faktor penskalaan untuk menstabilkan gradien.
d. V: Matriks *Value* yang berisi nilai representasi fitur sesungguhnya.

Tabel 2.8 *Pseudocode MHSA [12]*

No	Tahapan
1	Input: Terima kumpulan vektor petak dari lapisan penyematan atau lapisan sebelumnya.
2	Hitung matriks *Query*, *Key*, dan *Value* melalui operasi perkalian vektor masukan dengan bobot linier.
3	Lakukan perkalian titik (*dot product*) antara matriks *Query* dan matriks *Key* transposisi.
4	Bagi hasil perkalian tersebut dengan nilai akar kuadrat dari dimensi fitur untuk menstabilkan komputasi.
5	Terapkan fungsi *softmax* untuk mendapatkan distribusi probabilitas atau bobot atensi dari setiap petak.
6	Kalikan bobot probabilitas tersebut dengan matriks *Value* untuk mendapatkan representasi fitur akhir.
7	*Output*: Teruskan matriks hasil perhitungan atensi ke lapisan jaringan saraf feed-forward.

3. *Distillation Token*

Keunikan sejati dari DeiT terletak pada penggunaan token penyulingan atau distillation token yang ditambahkan ke dalam urutan petak gambar bersama dengan token klasifikasi utama [58]. Token ini berinteraksi dengan seluruh petak gambar melalui mekanisme atensi mandiri untuk menyerap pengetahuan dari sebuah model guru atau teacher model yang umumnya berupa jaringan

konvolusional tingkat lanjut. Melalui perhitungan token penyulingan ini, DeiT dapat meniru cara model konvolusi dalam mengekstraksi fitur spasial tanpa harus memiliki lapisan konvolusi itu sendiri [12].

Kajian literatur sebelumnya menyoroti bahwa pemanfaatan arsitektur berbasis mekanisme atensi telah membuka standar akurasi baru berkat kemampuannya memindai keseluruhan gambar secara serentak [20]. Penelitian tersebut sangat mendukung penerapan arsitektur *Transformer* sebagai penantang kuat bagi dominasi model konvolusional klasik pada tugas analisis gambar komersial. DeiT diposisikan sebagai variabel pembanding tunggal yang mewakili pendekatan perhitungan matriks global dalam mengklasifikasikan sentimen ulasan. Pengujian komparatif ini dirancang secara khusus untuk membuktikan hipotesis bahwa perhitungan hubungan spasial secara komprehensif akan memberikan akurasi deteksi yang lebih superior pada data gambar ulasan nyata.

2.3.8 *ConvNeXt*

ConvNeXt (*Convolutional Network*) merupakan arsitektur *Convolutional Neural Network* (CNN) modern yang dikembangkan oleh Meta AI sebagai hasil modernisasi arsitektur ResNet dengan mengadopsi berbagai desain yang sebelumnya diperkenalkan pada *Vision Transformer* (ViT) [59]. Pengembangan ConvNeXt bertujuan menunjukkan bahwa model berbasis konvolusi masih mampu bersaing dengan arsitektur *Transformer* apabila dilakukan penyempurnaan pada struktur jaringan, strategi pelatihan, serta konfigurasi blok konvolusi [18]. Berbeda dengan CNN konvensional, ConvNeXt menerapkan beberapa perubahan penting seperti penggunaan *large kernel convolution*, *depthwise convolution*, *Layer Normalization*, serta aktivasi *Gaussian Error Linear Unit* (GELU) untuk meningkatkan kemampuan ekstraksi fitur visual [59].

1. Arsitektur Model

   Secara struktural, ConvNeXt terdiri atas empat tahapan (stages) utama yang masing-masing diawali dengan proses *downsampling* untuk mengurangi resolusi citra sekaligus meningkatkan jumlah kanal fitur. Setiap stage tersusun

atas beberapa ConvNeXt *Block* yang mengombinasikan *depthwise convolution* berukuran 7×7, *Layer Normalization*, *pointwise convolution*, serta fungsi aktivasi Gaussian Error Linear Unit (GELU) [59]. Berbeda dengan CNN klasik yang umumnya menggunakan *Batch Normalization* dan ReLU, ConvNeXt mengadopsi *Layer Normalization* dan GELU untuk menghasilkan proses optimasi yang lebih stabil, terutama ketika dilatih menggunakan dataset berskala besar [60]. Pada bagian akhir jaringan diterapkan *Global Average Pooling* untuk mengubah peta fitur menjadi sebuah vektor representasi sebelum diteruskan menuju *fully connected layer* yang menghasilkan probabilitas setiap kelas [60].

Tabel 2.9 *Pseudocode* Arsitektur Blok ConvNeXt

No	Tahapan
1	Terima peta fitur masukan X berdimensi [C, H, W] dari lapisan sebelumnya, lalu simpan sebagai identitas residu awal (RESIDU = X).
2	Terapkan operasi konvolusi depthwise berukuran kernel 7 x 7 secara terpisah pada setiap kanal fitur X untuk mengekstraksi informasi spasial lokal secara efisien.
3	Lakukan normalisasi statistik (Layer Normalization) pada hasil konvolusi di sepanjang dimensi kanal fitur guna menstabilkan gradien.
4	Terapkan konvolusi pointwise berukuran 1 x 1 pertama untuk memperluas dimensi kanal fitur menjadi empat kali lipat (C menjadi 4C).
5	Lewatkan peta fitur yang telah diperluas ke dalam fungsi aktivasi Gaussian Error Linear Unit (GELU) untuk memperoleh pemetaan non-linear yang halus.
6	Terapkan konvolusi pointwise berukuran 1 x 1 kedua untuk menyusutkan kembali dimensi kanal fitur ke ukuran semula (4C menjadi C).
7	Kalikan matriks hasil proyeksi secara aljabar dengan parameter skala pembelajar (GAMMA) yang diinisialisasi pada nilai sangat kecil (10^{-6}).
8	Terapkan regularisasi stochastic depth guna menonaktifkan jalur pemrosesan blok secara acak selama fase pelatihan demi memitigasi overfitting.
9	Jumlahkan matriks hasil pemrosesan tersebut dengan identitas residu awal secara elemen-per-elemen (OUTPUT = RESIDU + F(X)).
10	Teruskan tensor OUTPUT ke blok jaringan berikutnya.

2. Tahapan pemrosesan

   Operasi utama pada ConvNeXt tetap didasarkan pada proses konvolusi untuk mengekstraksi pola visual dari citra masukan. Proses konvolusi dua dimensi dapat dinyatakan sebagai berikut.

$$Y(i,j) = \sum_m \quad \sum_n X(i+m, j+n) \times K(m,n)$$

Rumus 2.11. *Konvolusi dua dimensi* [12]

Keterangan:

a. X merupakan citra masukan (*input feature map*),

b. K merupakan kernel konvolusi,

c. Y merupakan hasil ekstraksi fitur (*output feature map*).

Berbeda dengan CNN konvensional, ConvNeXt memanfaatkan *Depthwise Convolution* yang menerapkan proses konvolusi secara terpisah pada setiap *channel* sehingga jumlah parameter dan kompleksitas komputasi dapat dikurangi tanpa menghilangkan kemampuan ekstraksi fitur. Setelah proses konvolusi selesai dilakukan, hasil keluaran diteruskan menuju fungsi aktivasi *Gaussian Error Linear Unit (GELU)* yang dinyatakan pada Rumus 2.12 berikut:

$$\mathrm{GELU}(x) = x\Phi(x)$$

Rumus 2.12. *Gaussian Error Linear Unit* $(GELU)$ [12]

dengan $\Phi(x)$ merupakan fungsi distribusi kumulatif (*Cumulative Distribution Function*) dari distribusi normal standar. Penggunaan GELU menghasilkan proses aktivasi yang lebih halus dibandingkan ReLU sehingga mampu meningkatkan stabilitas proses pelatihan dan kualitas representasi fitur yang dihasilkan.