BAB II

KAJIAN PUSTAKA

A. Kajian Teoritis

1. Konsep Dasar Sistem

a. Sistem Informasi

Sistem merupakan sekumpulan komponen yang saling berhubungan dan bekerja sama untuk mencapai tujuan tertentu. Ni et al. (2024:78) menjelaskan bahwa sistem tersusun atas elemen atau subsistem yang saling berinteraksi. Adi et al. (2022) menambahkan bahwa setiap komponen memiliki peran yang saling bergantung, sehingga gangguan pada satu bagian dapat memengaruhi kinerja sistem secara keseluruhan.

Sistem informasi mengolah masukan melalui serangkaian proses untuk menghasilkan keluaran yang bermanfaat. Elvarisna et al. (2025) menyatakan bahwa keterpaduan antarkomponen menjadi dasar dalam menghasilkan informasi yang memiliki nilai tambah. Oleh karena itu, perancangan sistem perlu memperhatikan hubungan antara data, proses, pengguna, dan teknologi agar seluruh fungsi dapat berjalan secara terarah.

Dalam penelitian ini, sistem digunakan untuk mengumpulkan, menyimpan, mengolah, dan menampilkan ulasan pengguna aplikasi *Access by KAI*. Setiap komponen bekerja secara terhubung, mulai dari pengambilan data ulasan, proses klasifikasi sentimen, pengelompokan aspek, penyimpanan hasil, hingga penyajian informasi melalui antarmuka sistem

b. Website

Website merupakan media yang memungkinkan pengguna mengakses informasi dan layanan melalui peramban. Darmawan (2023) menjelaskan bahwa sistem berbasis web dijalankan melalui *web browser* dan proses utama dilaksanakan pada server, sehingga pengguna tidak perlu memasang aplikasi tambahan pada perangkatnya.

Lebih lanjut, Siburian & Latifah (2023) menjelaskan bahwa sistem berbasis web menggunakan arsitektur *client-server*. Pihak klien mengirimkan permintaan melalui peramban, kemudian server memproses permintaan dan mengirimkan respons untuk ditampilkan kepada pengguna. Pemisahan antara antarmuka dan logika pemrosesan membuat pengembangan serta pemeliharaan sistem menjadi lebih terstruktur.

Menurut Mahardika et al. (2023) sistem berbasis web memiliki kelebihan berupa kemudahan akses, pemeliharaan yang lebih sederhana, dan kemampuan untuk dikembangkan sesuai kebutuhan. Dalam penelitian ini, website menjadi media utama bagi *Admin* untuk mengakses *Dashboard*, mengelola ulasan, menjalankan analisis, dan melihat masukan pengguna.

c. Aplikasi *Access by KAI*

Access by KAI merupakan aplikasi resmi PT Kereta Api Indonesia (Persero) yang menyediakan layanan perjalanan kereta api secara digital. Putri et al. (2025:2) menjelaskan bahwa aplikasi ini digunakan untuk memfasilitasi pemesanan tiket kereta api secara daring.

Aplikasi *Access by KAI* diluncurkan pada 10 Agustus 2023 sebagai pembaruan dari aplikasi KAI Access.

Menurut Damanhuri & Husein (2024:98) aplikasi tersebut menyediakan layanan pemesanan tiket, pembatalan, dan perubahan jadwal perjalanan. Selain itu, aplikasi juga memiliki fitur pendukung seperti KAIPay, RailFood, serta layanan lain yang berkaitan dengan perjalanan pengguna. Meskipun menyediakan banyak fitur, ulasan di Google Play Store masih menunjukkan berbagai keluhan pengguna. Maulana et al. (2023:186) mencatat adanya keluhan mengenai proses pemuatan yang lambat, waktu pemesanan yang terbatas, dan aplikasi yang sering mengalami kesalahan. Kondisi tersebut menunjukkan bahwa ulasan pengguna perlu dianalisis agar masalah yang sering muncul dapat diketahui dengan lebih cepat.

Berdasarkan penjelasan tersebut, sistem berbasis web memudahkan pengguna mengakses layanan tanpa perlu memasang aplikasi tambahan serta lebih mudah dikembangkan dan dipelihara. Dalam penelitian ini, website digunakan oleh *Admin* untuk mengelola ulasan, melakukan analisis, dan melihat masukan pengguna. Sementara itu, *Access by KAI* menyediakan berbagai layanan perjalanan kereta api secara digital, tetapi masih ditemukan keluhan seperti proses pemuatan yang lambat dan aplikasi yang sering mengalami kesalahan. Oleh karena itu, ulasan pengguna perlu dianalisis agar masalah pada aplikasi dapat diketahui dan diperbaiki.

2. Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan atau *Artificial Intelligence* merupakan bidang ilmu komputer yang bertujuan membuat sistem mampu menjalankan tugas yang biasanya membutuhkan kecerdasan manusia. Menurut Janiesch & Heinrich (2021) kecerdasan buatan memungkinkan komputer mempelajari pola dan ymembantu menyelesaikan tugas yang kompleks. Menurut Mukhamediev et al. (2022) menjelaskan bahwa penerapan kecerdasan buatan banyak didukung oleh *Machine Learning* untuk menangani tugas seperti klasifikasi, pengelompokan, dan prediksi. Dalam penelitian ini, kecerdasan buatan diterapkan melalui model IndoBERT untuk mengklasifikasikan sentimen ulasan pengguna secara otomatis.

a. *Machine Learning*

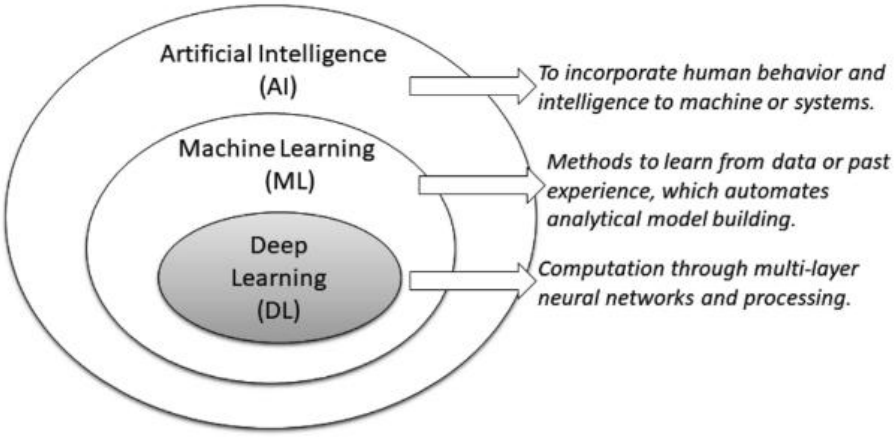
Machine Learning merupakan bagian dari kecerdasan buatan yang memungkinkan sistem mempelajari pola dari data. Niklas et al. (2022) menjelaskan bahwa *Machine Learning* membantu sistem menyelesaikan masalah berdasarkan data yang dipelajari. Pendekatan ini umumnya dibagi menjadi *supervised learning*, *unsupervised learning*, dan *reinforcement learning* sesuai dengan tujuan serta karakteristik datanya.

Menurut Jaydip (2021) menyatakan bahwa *Machine Learning* memungkinkan sistem mengembangkan pengetahuan secara otomatis tanpa harus diprogram secara khusus untuk setiap kondisi. Dalam penelitian ini, proses klasifikasi sentimen termasuk dalam *supervised*

learning karena model mempelajari data ulasan yang telah memiliki label sentimen.

b. *Deep Learning*

Deep Learning merupakan bagian dari *Machine Learning* yang menggunakan jaringan saraf dengan banyak lapisan. Iqbal (2021) menjelaskan bahwa *Deep Learning* mampu mempelajari pola yang kompleks dari data dalam jumlah besar dan telah diterapkan pada berbagai bidang, termasuk pengolahan teks. Hubungan antara *Artificial Intelligence*, *Machine Learning*, dan *Deep Learning* ditunjukkan pada Gambar 2.1.



Gambar 2. 1 Hubungan antara Artificial Intelligence, *Machine Learning*, dan *Deep Learning*

Sumber: Sarker, I, (2021)

Gambar 2.1 menunjukkan hubungan bertingkat antara *Artificial Intelligence*, *Machine Learning*, dan *Deep Learning*. Lingkaran paling luar menunjukkan bahwa *Artificial Intelligence* merupakan bidang yang paling luas, yaitu upaya membuat mesin mampu menjalankan tugas yang

membutuhkan kecerdasan. Di dalamnya terdapat *Machine Learning*, yaitu metode yang memungkinkan sistem mempelajari pola dari data atau pengalaman sebelumnya. *Deep Learning* berada di bagian paling dalam karena merupakan bagian khusus dari *Machine Learning* yang menggunakan jaringan saraf berlapis. Susunan tersebut menjelaskan bahwa setiap penerapan *Deep Learning* termasuk *Machine Learning* dan *Artificial Intelligence*, tetapi tidak semua metode *Artificial Intelligence* menggunakan *Deep Learning*.

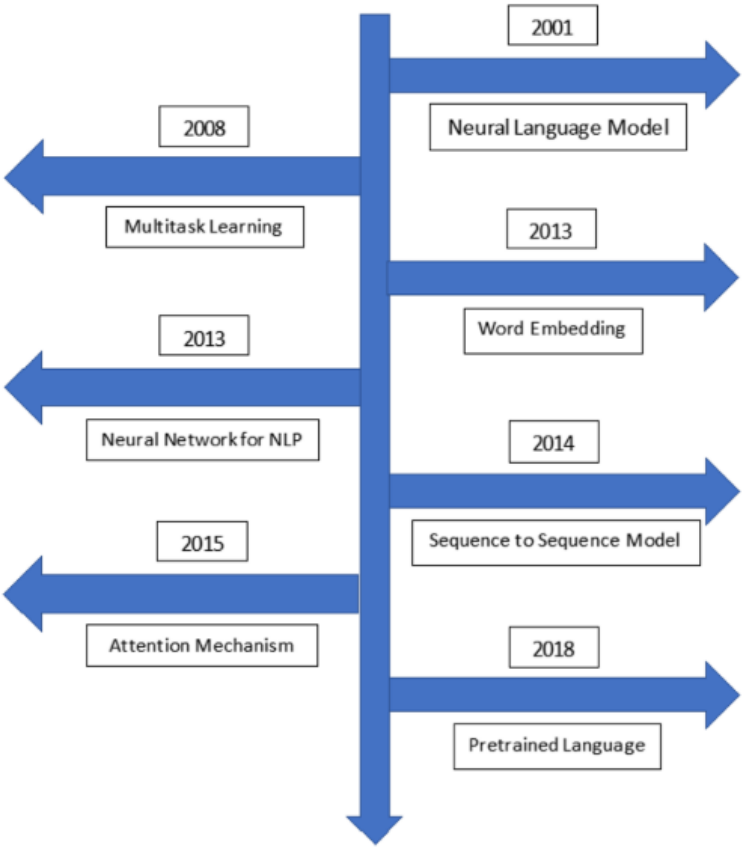
Selanjutnya, Janiesch & Heinrich (2021) menyatakan bahwa *Deep Learning* mampu memberikan hasil yang baik pada berbagai tugas yang sulit ditangani oleh pendekatan tradisional. Dalam penelitian ini, *Deep Learning* diterapkan melalui arsitektur *Transformer* pada model IndoBERT untuk memahami konteks kalimat dalam ulasan berbahasa Indonesia.

c. *Natural Language Processing* (NLP)

Natural Language Processing atau NLP merupakan cabang kecerdasan buatan yang berfokus pada pengolahan bahasa manusia oleh komputer. Khurana et al. (2023) menjelaskan bahwa NLP digunakan pada berbagai tugas, seperti penerjemahan, ekstraksi informasi, peringkasan teks, deteksi spam, dan sistem tanya jawab. Melalui NLP, komputer dapat mengenali pola, makna, dan hubungan antarkata dalam suatu kalimat. Kemampuan tersebut membuat NLP banyak digunakan untuk mengolah

data berbentuk teks yang jumlahnya besar dan sulit dianalisis secara manual.

Kemampuan tersebut membuat NLP banyak digunakan untuk mengolah data berbentuk teks yang jumlahnya besar dan sulit dianalisis secara manual. Perkembangan NLP berlangsung melalui berbagai pendekatan yang terus disempurnakan agar komputer semakin mampu memahami konteks dan makna bahasa secara lebih akurat. Tahapan perkembangan model *Natural Language Processing*dari pendekatan awal hingga model bahasa pralatih ditunjukkan pada Gambar 2.2.



Gambar 2. 2 Perkembangan Model NLP Modern

Sumber: Khurana et al., (2023)

Gambar 2.2 menunjukkan perkembangan model NLP secara bertahap. Perkembangan dimulai dari *neural language model* pada tahun 2001, kemudian dilanjutkan dengan multitask learning pada tahun 2008. Pada tahun 2013, *word embedding* dan *neural network* mulai digunakan untuk mengubah kata menjadi representasi numerik dan mempelajari pola bahasa. *Sequence-to-Sequence* model berkembang pada tahun 2014 untuk memproses urutan masukan menjadi urutan keluaran. *Attention mechanism* pada tahun 2015 membantu model memusatkan perhatian pada bagian teks yang paling penting. Perkembangan tersebut kemudian mengarah pada *pre-trained language model* pada tahun 2018, yaitu model yang terlebih dahulu mempelajari bahasa dari data berukuran besar sebelum disesuaikan untuk tugas tertentu. Tahapan terakhir inilah yang menjadi dasar bagi model BERT dan IndoBERT.

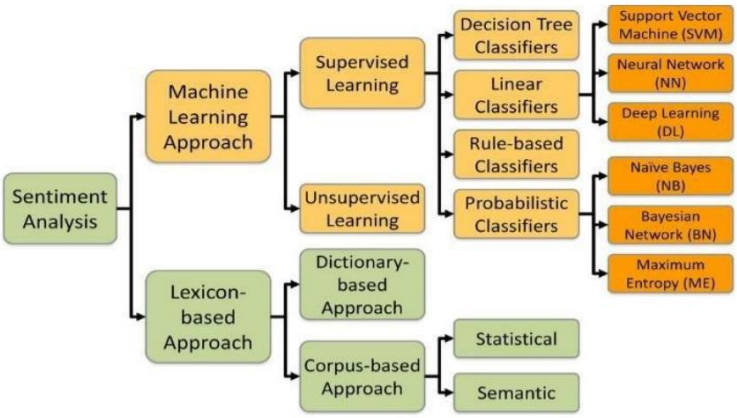
Menurut Tucudean et al. (2024) model berbasis *Transformer*, seperti BERT, banyak digunakan untuk menyelesaikan berbagai tugas NLP. Dalam penelitian ini, NLP digunakan untuk mengolah teks ulasan dan menentukan sentimen yang terkandung di dalamnya. Berdasarkan uraian tersebut, kecerdasan buatan merupakan teknologi yang memungkinkan komputer mempelajari pola dan menyelesaikan tugas yang biasanya membutuhkan kecerdasan manusia. Penerapannya didukung oleh *Machine Learning*, sedangkan *Deep Learning* digunakan untuk mempelajari pola yang lebih kompleks melalui jaringan saraf berlapis. Dalam pengolahan teks, *Natural Language Processing*

membantu komputer memahami makna dan konteks bahasa manusia. Pada penelitian ini, konsep tersebut diterapkan melalui model IndoBERT untuk mengklasifikasikan sentimen ulasan pengguna *Access by KAI* secara otomatis menjadi positif, netral, dan negatif.

3. Analisis Sentimen

Analisis sentimen merupakan proses untuk mengenali dan mengelompokkan pendapat atau perasaan yang terdapat dalam sebuah teks. Proses ini digunakan untuk mengetahui apakah suatu pendapat mengandung tanggapan yang bersifat positif, netral, atau negatif. Wankhade et al. (2022) menjelaskan bahwa analisis sentimen menggunakan NLP dan *text mining* untuk mengidentifikasi informasi subjektif serta menentukan polaritas sentimen yang sesuai.

Menurut Cui et al. (2023) analisis sentimen menjadi salah satu bidang penelitian yang aktif dalam NLP karena dapat diterapkan pada berbagai jenis data opini. Melalui analisis sentimen, ulasan yang sebelumnya berbentuk teks tidak terstruktur dapat diubah menjadi informasi yang lebih mudah dipahami dan digunakan sebagai bahan evaluasi. Pengelompokan pendekatan dan teknik yang dapat digunak an dalam analisis sentimen ditunjukkan pada Gambar 2.3.



Gambar 2. 3 Klasifikasi Teknik Analisis Sentimen
Sumber: (Sayeed et al., 2023)

Gambar 2.3 menunjukkan bahwa analisis sentimen dapat dilakukan melalui dua kelompok pendekatan utama, yaitu *Machine Learning approach* dan *lexicon-based approach*. Pendekatan *Machine Learning* dibagi menjadi *supervised learning* dan *unsupervised learning*. Pada *supervised learning*, model belajar menggunakan data yang telah memiliki label dan dapat memakai *decision tree*, linear *Class*ifier, *rule-based Class*sifier, maupun *probabilistic Classifier*. Contoh algoritmanya meliputi *Support Vector Machine*, *neural network* atau *Deep Learning*, *Naive Bayes*, *Bayesian Network*, dan *Maximum Entropy*. Sementara itu, pendekatan berbasis leksikon menentukan sentimen menggunakan kamus kata atau korpus, kemudian dapat dianalisis secara statistik maupun semantik. Penelitian ini berada pada jalur *supervised learning* karena IndoBERT dilatih menggunakan ulasan yang telah diberi label positif, netral, dan negatif.

Dalam penelitian ini, analisis sentimen digunakan untuk mengelompokkan ulasan pengguna aplikasi *Access by KAI* ke dalam

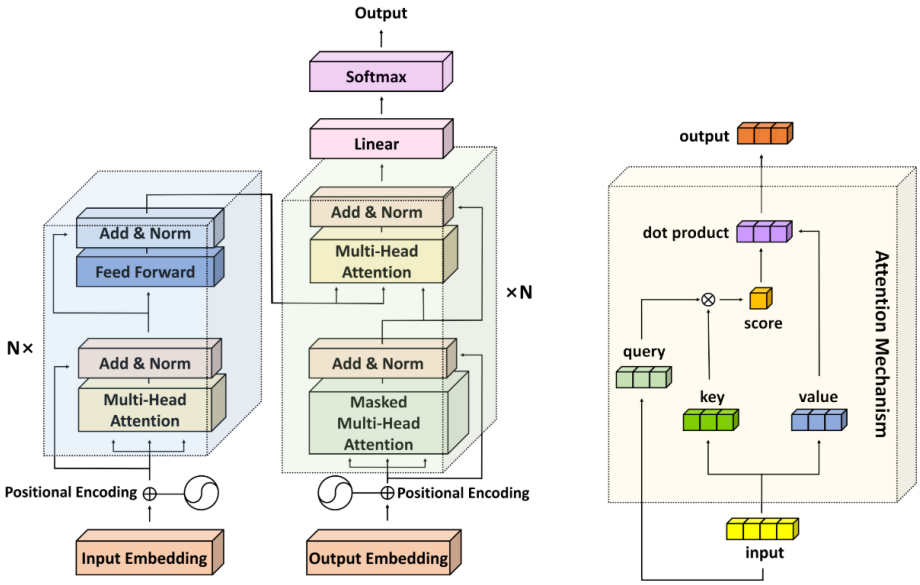
sentimen positif, netral, dan negatif. Hasil klasifikasi tersebut selanjutnya disajikan dalam bentuk jumlah, persentase, dan grafik agar pola opini pengguna dapat diamati dengan lebih mudah.

4. Model *Transformer*

Transformer merupakan arsitektur *Deep Learning* yang banyak digunakan pada pengolahan bahasa alami. Rahali (2023) menjelaskan bahwa *Transformer* memanfaatkan mekanisme *self-attention* untuk memahami hubungan antarelemen dalam suatu urutan, termasuk hubungan antarkata yang letaknya berjauhan dalam kalimat. Menurut Zhang et al. (2023) bahwa *Transformer* secara umum tersusun atas *encoder* dan *decoder* yang memiliki lapisan atensi serta jaringan *feed-forward*. Mekanisme atensi membantu model menentukan bagian teks yang paling relevan ketika memproses suatu masukan.

Pada prosesnya, setiap kata terlebih dahulu diubah menjadi representasi numerik melalui *embedding*. Informasi posisi kata kemudian ditambahkan agar model tetap memahami urutan kata dalam kalimat. Melalui mekanisme *self-attention*, *Transformer* membandingkan hubungan setiap kata dengan kata lainnya dan memberikan perhatian lebih besar pada bagian yang dianggap penting. Hasil pemrosesan tersebut diteruskan ke lapisan berikutnya untuk membentuk pemahaman konteks yang lebih lengkap. Dengan mekanisme ini, model dapat memahami makna kalimat secara keseluruhan, bukan hanya membaca kata satu per satu. Susunan *encoder*,

decoder, dan mekanisme *attention* pada arsitektur *Transformer* ditunjukkan pada Gambar 2.4.



Gambar 2. 4 Arsitektur Dasar *Transformer*s
Sumber: (Vaswani et al., 2017)

Berdasarkan Gambar 2.4 memperlihatkan dua bagian utama *Transformer*, yaitu *encoder* di sebelah kiri dan *decoder* di bagian tengah. Teks masukan terlebih dahulu diubah menjadi input *embedding*, kemudian positional encoding ditambahkan agar model mengetahui urutan kata. Pada *encoder*, multi-head attention membantu model melihat hubungan setiap kata dengan kata lain, sedangkan *feed-forward network* mengolah hasilnya. Bagian *Add* & *Norm* digunakan untuk menjaga informasi dari lapisan sebelumnya dan membuat proses pelatihan lebih stabil. *Decoder* memiliki proses yang hampir sama, tetapi menggunakan masked *multi-head attention* agar model tidak melihat keluaran yang belum seharusnya diketahui.

Bagian kanan gambar menjelaskan mekanisme *attention* menggunakan *query*, *key*, dan value. *Query* dibandingkan dengan *key* untuk

menghasilkan skor perhatian, kemudian skor tersebut digunakan untuk memilih informasi penting dari value. BERT dan IndoBERT hanya memanfaatkan bagian *encoder Transformer* karena keduanya digunakan untuk memahami isi teks, bukan menghasilkan urutan teks baru seperti pada *decoder*. Perhitungan hubungan antara *query*, *key*, dan *value* pada mekanisme *attention* dapat dituliskan melalui rumus 2.1.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.1)$$

Persamaan (2.1) merupakan *scaled dot-product attention*. Q menyatakan query, K menyatakan *key*, V menyatakan value, dan d_k menyatakan dimensi *key*. Perkalian Q dengan K^T menghasilkan skor hubungan antartoken. Nilai tersebut dibagi dengan akar d_k agar skala skor tetap stabil, kemudian fungsi *softmax* mengubahnya menjadi bobot perhatian yang digunakan untuk menggabungkan informasi.

5. *Bidirectional Encoder Representations from Transformers* (BERT)

BERT (*Bidirectional Encoder Representations from Transformers*) merupakan model bahasa berbasis *Transformer* yang digunakan untuk memahami isi teks. Menurut Mudding (2024), BERT membaca kata dari dua arah, yaitu dari kata sebelum dan sesudahnya. Dengan cara ini, BERT dapat memahami hubungan antarkata dan makna kalimat dengan lebih baik. Prosesnya dilakukan melalui tokenisasi, *embedding*, dan *self-attention*, sehingga BERT dapat digunakan untuk klasifikasi teks dan analisis sentimen.

Menurut Sukmawati et al. (2024) BERT dilatih menggunakan data teks dalam jumlah besar agar mampu memahami konteks bahasa. BERT

memiliki dua tahap utama, yaitu pre-training untuk mempelajari bahasa secara umum dan *fine-tuning* untuk menyesuaikan model dengan tugas tertentu. BERT juga menggunakan token khusus, seperti [CLS], [SEP], dan [PAD], untuk membantu proses pengolahan teks.

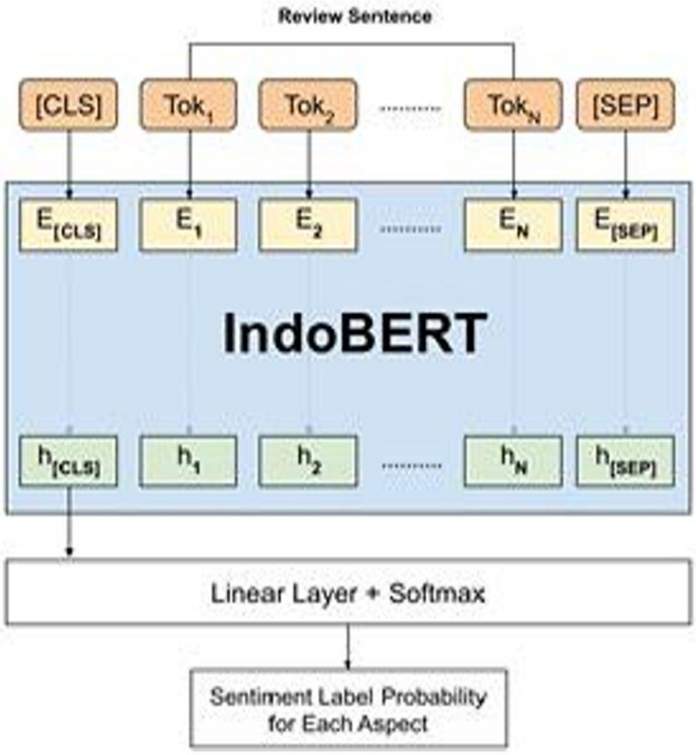
Berdasarkan kedua penjelasan tersebut, BERT dapat disimpulkan sebagai model bahasa yang mampu memahami isi teks dengan memperhatikan konteks kata secara dua arah. Proses tokenisasi, *embedding*, dan *self-attention* membantu BERT mengenali hubungan antarkata, sedangkan proses *fine-tuning* memungkinkan model disesuaikan untuk kebutuhan klasifikasi tertentu. Oleh karena itu, BERT sesuai digunakan sebagai dasar pengembangan IndoBERT dalam penelitian ini untuk mengklasifikasikan sentimen ulasan pengguna aplikasi *Access by KAI* menjadi sentimen positif, netral, dan negatif.

6. IndoBERT

IndoBERT merupakan pengembangan BERT yang dilatih menggunakan korpus berbahasa Indonesia. Koto & Baldwin, (2020) menjelaskan bahwa IndoBERT dikembangkan melalui proyek IndoLEM untuk menyediakan sumber daya NLP bahasa Indonesia yang lebih terstandarisasi. Karena mempelajari teks bahasa Indonesia dalam jumlah besar, model ini memiliki representasi yang lebih sesuai dengan susunan kata dan konteks bahasa Indonesia.

Menurut Nugroho et al. (2020) model BERT yang dilatih menggunakan bahasa Indonesia memberikan hasil yang baik dalam analisis

ulasan aplikasi. Kemampuan memahami konteks membuat IndoBERT sesuai untuk menangani ulasan berbahasa tidak baku, singkatan, kata serapan, dan susunan kalimat yang beragam. Proses tersebut membantu model memahami hubungan antarkata sehingga klasifikasi tetap mempertimbangkan konteks kalimat. Alur pemrosesan kalimat ulasan oleh IndoBERT hingga menghasilkan probabilitas kelas sentimen ditunjukkan pada Gambar 2.5.



Gambar 2. 5 Alur Klasifikasi Sentimen Menggunakan IndoBERT
Sumber: Mujilahwati et al., (2026)

Berdasarkan Gambar 2.5 alur pengolahan sebuah kalimat ulasan menggunakan IndoBERT. Kalimat terlebih dahulu dipecah menjadi Tok1 sampai TokN, kemudian token [CLS] ditambahkan pada awal kalimat dan token [SEP] pada akhir kalimat. Setiap token diubah menjadi *embedding*, yang pada gambar ditandai dengan E[CLS], E1, E2, hingga E[SEP]. Seluruh

embedding diproses oleh IndoBERT untuk menghasilkan representasi kontekstual h[CLS], h1, h2, hingga h[SEP]. Representasi h[CLS] dipakai sebagai ringkasan seluruh kalimat dan diteruskan ke linear layer serta fungsi *softmax*. Linear layer menghasilkan skor untuk setiap kelas, sedangkan *softmax* mengubah skor tersebut menjadi nilai probabilitas. Kelas dengan probabilitas tertinggi dipilih sebagai hasil prediksi. Fungsi *softmax* digunakan untuk mengubah nilai keluaran model atau *logits* menjadi nilai probabilitas pada setiap kelas sentimen. Jumlah seluruh probabilitas yang dihasilkan bernilai satu. Fungsi *softmax* dapat dituliskan pada persamaan 2.2.

$$P\_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{2.2}$$

Pada Persamaan (2.2), z_i merupakan nilai logit untuk kelas ke-i, sedangkan jumlah eksponensial seluruh logit menjadi pembagi. Hasil *softmax* menghasilkan probabilitas untuk kelas negatif, netral, dan positif dengan jumlah keseluruhan bernilai satu. Kelas yang memiliki probabilitas tertinggi dipilih sebagai hasil klasifikasi sentimen.

Pada penelitian ini, probabilitas tersebut digunakan untuk menentukan sentimen positif, netral, atau negatif. Keterangan keluaran per aspek pada gambar sumber tidak digunakan untuk menentukan aspek dalam penelitian ini karena kategori aspek ditetapkan secara terpisah melalui *Rule Based Classification*.

a. Tokenisasi

Tokenisasi merupakan proses memecah teks menjadi unit-unit token yang dapat dibaca oleh model. Pada IndoBERT, token dapat berupa

kata utuh maupun bagian dari kata. Token [CLS] ditambahkan pada awal teks, sedangkan token [SEP] ditempatkan pada akhir teks. Setiap token kemudian diubah menjadi input IDs, sementara attention mask digunakan untuk membedakan token yang perlu diproses dari bagian padding.

b. *Fine-tuning*

Fine-tuning merupakan proses menyesuaikan model yang telah melalui tahap pre-training menggunakan dataset yang lebih khusus. Jayadianti et al. (2022) menegaskan bahwa *fine-tuning* pada IndoBERT dapat meningkatkan kemampuan model dalam memahami teks ulasan berbahasa Indonesia. Dany et al. (2024) juga menunjukkan bahwa IndoBERT mampu mengenali hubungan konteks dalam kalimat secara lebih terperinci. Selama proses *fine-tuning*, kesalahan prediksi model dapat dihitung menggunakan fungsi *cross-entropy loss*. Fungsi ini membandingkan probabilitas hasil prediksi dengan label sentimen yang sebenarnya. Semakin kecil nilai *loss*, semakin dekat hasil prediksi model dengan label aktual. Rumus *cross-entropy loss* dapat dituliskan pada persamaan 2.3.

$$L = -\sum_{i=1}^{C} \quad y_i ln(p_i) \tag{2. 3}$$

Pada Persamaan (2.3), y_i menyatakan label aktual dan p_i menyatakan probabilitas hasil prediksi untuk kelas ke-i. Nilai *cross-entropy loss* akan mengecil ketika probabilitas pada kelas yang benar semakin besar. Oleh karena itu, penurunan loss selama pelatihan menunjukkan bahwa prediksi model semakin mendekati label sentimen yang sebenarnya. Dalam penelitian

ini, IndoBERT disesuaikan menggunakan data ulasan berlabel positif, netral, dan negatif. Model hasil pelatihan kemudian diekspor ke format ONNX agar dapat dijalankan pada *backend* sistem dengan proses inferensi yang lebih ringan.

7. Klasifikasi Berbasis Aturan (*Rule-Bases Classification*)

Klasifikasi berbasis aturan merupakan metode yang menentukan kelas berdasarkan aturan yang telah disusun sebelumnya. Suhwan et al. (2022) menjelaskan bahwa pendekatan berbasis aturan menghasilkan model yang mudah dipahami karena keputusan klasifikasi dapat ditelusuri melalui aturan yang digunakan. Selanjutnya Bishwamittra et al. (2022) menjelaskan bahwa *rule-based Classifier* dapat merepresentasikan keputusan melalui aturan IF-THEN yang tersusun dari fitur masukan. Pada data teks, aturan dapat dibuat menggunakan kata kunci atau pola tertentu. Jika sebuah ulasan mengandung kata yang berhubungan dengan pembayaran, misalnya, sistem dapat memasukkannya ke kategori Pembayaran.

Dalam penelitian ini, klasifikasi berbasis aturan digunakan untuk mengelompokkan ulasan ke dalam kategori *Login*, Sistem, Pembayaran, UI/UX, Performa, dan Lainnya. Pendekatan yang sama juga digunakan untuk mendeteksi ulasan yang mengandung masukan melalui kata-kata seperti “tolong”, “harap”, “sebaiknya”, atau bentuk permintaan perbaikan lainnya. Metode ini dipilih karena aturan dapat diperiksa dan disesuaikan tanpa harus melatih ulang model IndoBERT.

8. Pengumpulan dan Pengolahan Data Ulasan

a. Web Scraping

Web scraping merupakan proses pengambilan data dari halaman web secara otomatis. Khder (2021) menjelaskan bahwa *web scraping* dapat mengubah data pada halaman web menjadi bentuk yang lebih terstruktur agar dapat disimpan dan diolah lebih lanjut. Selanjutnya, Yasin et al. (2024) menggunakan proses scraping berbasis Python untuk mengambil ulasan dari *Google Play Store*. Bernanda et al. (2024) juga memanfaatkan pustaka *google-play-scraper* untuk mengumpulkan ulasan dan rating sebagai dataset analisis sentimen berbasis *Transformer*. Pendekatan tersebut menunjukkan bahwa scraping dapat membantu mengumpulkan data dalam jumlah besar secara lebih efisien dibandingkan pencatatan manual.

Dalam penelitian ini, pustaka *google-play-scraper* digunakan untuk mengambil teks ulasan, nama pengguna, rating, tanggal, dan metadata lain dari aplikasi *Access by KAI*. Data yang berhasil diambil kemudian disimpan ke dalam basis data dan digunakan sebagai masukan untuk proses klasifikasi.

b. *Preprocessing* Teks

Preprocessing teks merupakan tahap penyiapan data sebelum masuk ke model. Tahap ini bertujuan merapikan teks dan mengurangi unsur yang tidak diperlukan agar proses tokenisasi dapat berjalan secara konsisten.

Proses yang dilakukan pada penelitian ini mencakup pembersihan karakter yang tidak relevan, perapian spasi, penyesuaian format teks, serta penyiapan masukan sesuai kebutuhan tokenizer IndoBERT. Preprocessing dilakukan secara hati-hati agar informasi penting dalam ulasan tidak hilang, terutama karena model IndoBERT memanfaatkan konteks kalimat. Setelah teks disiapkan, tokenizer mengubah ulasan menjadi token, input IDs, dan attention mask. Hasil tersebut kemudian diteruskan ke model untuk menghasilkan prediksi sentimen.

c. *Word cloud*

Word cloud merupakan teknik visualisasi teks yang menampilkan sekumpulan kata berdasarkan frekuensi kemunculannya. Kata yang memiliki frekuensi lebih tinggi ditampilkan dengan ukuran yang lebih besar sehingga kata atau topik yang dominan dalam suatu kumpulan teks dapat dikenali secara lebih cepat. Wibowo et al. (2024) menjelaskan bahwa *word cloud* banyak digunakan dalam pengolahan teks karena mampu memberikan representasi visual yang tetap informatif terhadap kata-kata yang dianalisis.

Pada penelitian ini, *word cloud* digunakan untuk menampilkan kata-kata yang paling sering muncul dalam ulasan pengguna aplikasi *Access by KAI* setelah melalui tahap pembersihan teks dan penyaringan kata umum yang kurang memberikan informasi. Hasil visualisasi dikelompokkan ke dalam kategori Semua, Negatif, Netral, dan Positif agar Admin dapat membandingkan kata dominan pada setiap kelompok

sentimen. Pengelompokan tersebut sejalan dengan penelitian Sitti et al. (2025) yang memanfaatkan *word cloud* untuk mengidentifikasi kata dominan pada data teks yang telah dikelompokkan ke dalam sentimen netral, positif, dan negatif.

Berdasarkan penelitian tersebut dapat disimpulkan bahwa *word cloud* dapat digunakan sebagai visualisasi pendukung untuk mengetahui persebaran kata dan topik yang menjadi ciri pada masing-masing kelas sentimen. Ukuran kata pada *word cloud* menunjukkan tingkat frekuensi kemunculan, tetapi tidak secara langsung menentukan makna atau sentimen suatu ulasan. Oleh karena itu, visualisasi tersebut digunakan untuk membantu interpretasi hasil, sedangkan penentuan sentimen tetap didasarkan pada hasil klasifikasi model IndoBERT dan konteks keseluruhan ulasan.

9. *Framework*

Framework merupakan kerangka kerja yang menyediakan struktur serta komponen siap pakai untuk membantu proses pengembangan perangkat lunak. Bartlomiej & Marek, (2025) menjelaskan bahwa framework mendukung penggunaan ulang kode dan membantu meningkatkan kualitas aplikasi karena berbagai komponennya telah dibangun dan diuji sebelumnya. Anuar et al. (2022) menambahkan bahwa framework web mempermudah pengembang dalam membangun fungsi pada sisi *backend* maupun *Frontend*.

Dalam penelitian ini, teknologi utama yang digunakan adalah FastAPI untuk *backend*, Vue.js untuk *Frontend*, dan ONNX Runtime untuk

menjalankan model. Tailwind CSS digunakan untuk membantu pengaturan tampilan, sedangkan Chart.js digunakan untuk menyajikan data dalam bentuk grafik.

a. FastAPI

FastAPI merupakan framework Python yang digunakan untuk membangun Application Programming Interface atau API. Rasulov (2024) menjelaskan bahwa FastAPI memanfaatkan *type hints* Python dan mendukung validasi data serta dokumentasi OpenAPI secara otomatis. Menurut Bartlomiej & Marek (2025) FastAPI memberikan performa terbaik pada hampir seluruh skenario pengujian permintaan HTTP dasar dan sebagian besar operasi CRUD. Dalam penelitian ini, FastAPI menangani proses autentikasi *Admin*, pengambilan data ulasan, akses basis data, pemanggilan model IndoBERT, serta pengiriman hasil analisis ke *Frontend*.

c. Vue JS

Vue.js merupakan framework JavaScript yang digunakan untuk membangun antarmuka web yang reaktif. Jakub & Artur (2023) menyatakan bahwa keberlangsungan sebuah framework dipengaruhi oleh kemampuan teknologi tersebut dalam menyesuaikan diri dengan kebutuhan pengembangan modern.

Menurut Bielak et al. (2022) dependensi komponen pada Vue dilacak secara otomatis ketika proses rendering sehingga sistem dapat menentukan komponen yang perlu direnderer ulang saat terjadi perubahan

state.. Piastou, (2023) juga menyebutkan bahwa Vue.js memiliki penggunaan sumber daya yang relatif ringan. Dalam penelitian ini, Vue 3 digunakan untuk membangun halaman *Login*, *Dashboard*, Live Scraping, AI Pipeline, dan Masukan Ulasan.

10. Database

Database digunakan untuk menyimpan data secara terstruktur agar dapat ditambahkan, diperbarui, dicari, dan ditampilkan kembali. Patel et al. (2023) menjelaskan bahwa teknologi basis data terus berkembang mengikuti kebutuhan pengelolaan data. Khan et al. (2023) membagi sistem basis data ke dalam beberapa kelompok, termasuk basis data relasional, NoSQL, dan NewSQL.

Penelitian ini menggunakan basis data relasional karena data pengguna, ulasan, hasil sentimen, kategori aspek, dan status ulasan memiliki hubungan yang jelas. Penyimpanan yang terstruktur membantu sistem menjaga konsistensi data selama proses scraping, analisis, dan visualisasi.

a. PostgreSQL

PostgreSQL merupakan sistem manajemen basis data relasional yang bersifat *open source*. Salunke & Ouda, (2024) menjelaskan bahwa PostgreSQL memiliki performa yang baik dalam operasi pembacaan data berskala besar. Trillo-montero et al. (2023) menambahkan bahwa PostgreSQL dapat diintegrasikan dengan perangkat pengelolaan basis data seperti pg*Admin*.

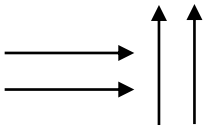
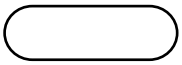
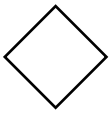
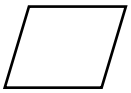
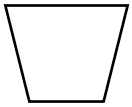
Dalam penelitian ini, PostgreSQL digunakan untuk menyimpan akun *Admin*, data ulasan, hasil klasifikasi sentimen, kategori aspek, status penanganan ulasan, dan informasi pendukung lainnya. Data yang tersimpan kemudian digunakan oleh *backend* untuk menyusun ringkasan dan grafik pada *Dashboard*.

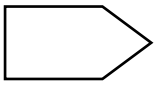
11. Flowchart

Flowchart atau bagan alir merupakan representasi grafis yang menunjukkan urutan proses dalam suatu sistem. Hartono, (2021) menjelaskan bahwa flowchart digunakan untuk menggambarkan alur program atau prosedur secara logis. Listyoningrum et al., (2023) menambahkan bahwa flowchart membantu pengembang memahami pekerjaan yang kompleks secara lebih terstruktur. Melalui flowchart, setiap tahapan dapat dilihat mulai dari proses awal, masukan, pengolahan data, pengambilan keputusan, hingga keluaran yang dihasilkan. Penggunaan simbol yang berbeda pada setiap proses juga membantu pembaca mengenali fungsi dan hubungan antarbagian dalam sistem.

Dengan demikian, flowchart dapat mempermudah proses perancangan, pemeriksaan alur, serta penyampaian cara kerja sistem kepada pihak lain. Simbol-simbol yang digunakan pada flowchart beserta fungsi masing-masing ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Simbol Flowchart

Simbol	Nama Simbol	Keterangan
	*Flow*	Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini juga disebut dengan *Connection Line*
	*On-Page Reference*	Simbol untuk keluar – masuk atau penyambung proses dalam lembar kerja yang sama
	*Off-Page Reference*	Simbol untuk keluar – masuk atau penyambung proses dalam lembar kerja yang berbeda
	*Terminator*	Simbol yang menyatakan awal atau akhir suatu program
	*Process*	Simbol yang menyatakan proses yang dilakukan komputer
	*Decision*	Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu iya atau tidak
	*Input Output*	Simbol yang menyatakan proses *input* atau *output* tanpa bergantung dengan peralatan
	*Manual Operation*	Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer
	*Document*	Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu di cetak
	*Predifine Proses*	Simbol untuk suatu bagian (sub bab program) atau prosedur

	*Display*	Simbol yang menyatkan peralatan *output* yang digunakan
	*Preparation*	Simbol yang menyatkan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal

Sumber: Putri (2024)

Tabel 2.1 menunjukkan nama simbol flowchart dan kegunaannya dalam menggambarkan alur proses. Simbol Flow digunakan untuk menghubungkan satu langkah dengan langkah lain. On-Page Reference dan Off-Page Reference digunakan ketika alur perlu disambungkan pada bagian atau halaman lain. Terminator menunjukkan awal dan akhir, Process menunjukkan kegiatan yang dikerjakan sistem, sedangkan Decision menunjukkan percabangan berdasarkan suatu kondisi. Simbol Input/Output digunakan untuk data yang masuk atau keluar. Simbol lainnya, seperti Document, Predefined Process, Display, dan Preparation, membantu menjelaskan sumber dokumen, subproses, keluaran, dan tahap persiapan. Dalam penelitian ini, simbol-simbol tersebut digunakan untuk menggambarkan proses dari pengumpulan ulasan hingga hasil analisis ditampilkan.

12. *Unified Modeling Language* (UML)

Unified Modeling Language atau UML merupakan bahasa pemodelan visual yang digunakan untuk menggambarkan kebutuhan, struktur, dan perilaku sistem. Sandy et al. (2024:3) menjelaskan bahwa UML digunakan dalam perancangan perangkat lunak berorientasi objek.

Syahrizal Syahrizal, (2025:155) menambahkan bahwa UML membantu menggambarkan struktur dan cara kerja sistem secara visual.

Dalam penelitian ini, UML digunakan untuk menjelaskan fungsi sistem, alur aktivitas, komunikasi antarkomponen, dan struktur data. Diagram yang digunakan meliputi *Use Case* Diagram, *Activity* Diagram, *Sequence* Diagram, dan *Class* Diagram.

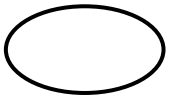
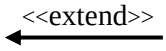
a. *Use Case* Diagram

Use Case Diagram menggambarkan hubungan antara aktor dan fungsi yang disediakan oleh sistem. Sandy et al. (2024:4) menjelaskan bahwa diagram ini digunakan untuk memvisualisasikan interaksi pengguna dengan sistem. Syahrizal Syahrizal (2025:155-156) menambahkan bahwa *Use Case* membantu menunjukkan fungsi yang tersedia dan aktor yang berhak menggunakannya.

Dalam penelitian ini, *Use Case* Diagram menggambarkan interaksi *Admin* dengan fitur *Login*, *Dashboard*, Live Scraping, AI Pipeline, Masukan Ulasan, dan *Logout* Simbol dan hubungan yang digunakan dalam *Use Case* Diagram ditunjukkan pada Tabel 2.2.

Tabel 2. 2 Simbol *Use Case* Diagram

Simbol	Elemen	Keterangan
	Aktor	rang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.

	Use Case	Deskripsi dari urutan aksi aksi yang ditampilkansistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
	Generalization	Hubungan generalisasi dan spesialisasi (umum- khusus) antar dua buah Use Case dimana fungsi yangsatu adalah fungsi yang lebih umum dari yang lainnya.
	Include	Relasi Use Case tambahan ke sebuah Use Case dimana Use Case yang ditambahkan memerlukan Use Case ini untuk menjalankan fungsinya.
	Extend	Relasi Use Case tambahan ke sebuah Use Case, dimana Use Case yang ditambahkan dapat berdirisendiri.
	Association	Komunikasi antar aktor dan use case yang berpartisipasi pada Use Case atau Use Case memiliki interaksi dengan aktor.

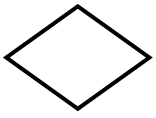
Sumber: Sandy et al. (2024:5)

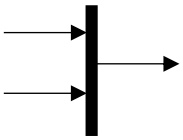
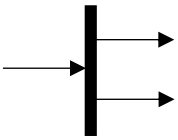
Tabel 2.2 menunjukkan bahwa aktor mewakili pihak yang berinteraksi dengan sistem, sedangkan *Use Case* mewakili fungsi yang dapat dijalankan. Association menghubungkan aktor dengan fungsi yang digunakannya. Generalization menunjukkan hubungan umum dan khusus, include menunjukkan fungsi yang selalu membutuhkan fungsi lain, dan extend menunjukkan fungsi tambahan yang hanya dijalankan pada kondisi tertentu. Simbol-simbol tersebut digunakan agar batas akses *Admin* dan fungsi utama sistem dapat dibaca dengan jelas.

b. *Activity* Diagram

Activity Diagram menggambarkan alur kerja suatu proses dari awal hingga akhir. Syahrizal Syahrizal (2025:157) menjelaskan bahwa *Activity* diagram dapat digunakan untuk menunjukkan aliran kerja sistem, proses bisnis, maupun menu perangkat lunak. Dalam penelitian ini, *Activity* Diagram digunakan untuk menggambarkan urutan aktivitas pada proses *Login*, *Dashboard*, Live Scraping, AI Pipeline, Masukan Ulasan, dan *Logout*. Simbol-simbol yang digunakan untuk menggambarkan alur aktivitas ditunjukkan pada Tabel 2.3.

Tabel 2. 3 Simbol Activity Diagram

Simbol	Elemen	Keterangan
	*Activity*	Memperlihatkan bagaimana masing - masing kelas antarmuka saling berinteraksi satu sama lain.
	Control Flow	Menunjukan urutan eksekusi
	*Start Poimt*	Menyatakn bahwa sebuah objek dari sebuah action atau *Activity* ke action
	*End Point*	Menyatakn bahwa sebuah objek dibentuk atau diawali
	*Decision*	Menunjukan penggambaran suatu keputusan atau tindakan yang harus diambil pada kondisi tertentu

	*Object Flow*	Menunjukan aliran object dari sebuah action atau *Activity* ke action
	*Join*/Penggabungan	Menyatakan untuk menggabungkan kembali *Activity* atau action yyang aparallel
	*Fork*	Menyatakan untuk memecah behavior menjadi *Activity* atau action yang parallel

Sumber: Sandy et al. (2024:6)

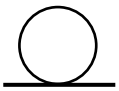
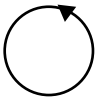
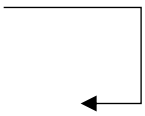
Tabel 2.3 menunjukkan unsur utama *Activity* Diagram. Start Point menandai awal proses dan End Point menandai akhir proses. *Activity* menunjukkan pekerjaan yang dilakukan, sedangkan Control Flow memperlihatkan urutan perpindahan antaraktivitas. Decision digunakan ketika proses memiliki lebih dari satu kemungkinan berdasarkan kondisi. Fork memecah alur menjadi beberapa aktivitas paralel, sedangkan Join menggabungkan kembali aktivitas tersebut. Dengan simbol ini, urutan kerja setiap menu dapat dijelaskan dari tindakan *Admin* sampai respons yang diberikan sistem.

c. *Sequence* Diagram

Sequence Diagram menggambarkan interaksi antarobjek berdasarkan urutan waktu. Sandy et al. (2024:4) menjelaskan bahwa diagram ini menunjukkan pesan yang dikirim dan diterima oleh objek di dalam maupun di sekitar sistem.

Dalam penelitian ini, *Sequence* Diagram digunakan untuk menjelaskan komunikasi antara *Admin*, *Frontend* Vue.js, *backend* FastAPI, model IndoBERT, dan PostgreSQL pada setiap proses utama sistem. Diagram tersebut membantu menunjukkan urutan permintaan, pemrosesan, dan respons secara lebih jelas. Simbol-simbol yang digunakan untuk menggambarkan alur aktivitas ditunjukkan pada Tabel 2.4.

Tabel 2. 4 Simbol Sequence Diagram

Simbol	Elemen	Keterangan
	Entity Class	merupakan bagian dari system yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	Boundary Class	berisi kumpulan kelas yang menjadi interfaces atau interaksi antar satu atau lebih actor dengan sistem, seperti tampilan form entry dan form cetak.
	Control Class	Suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	Recursive	menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Messege	Simbol mengirim pesan antar class
	Activation	mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus

		dengan durasi aktivitas sebuah operasi.
	Lifeline	garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

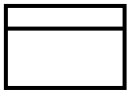
Sumber: (Muhammad et al., 2021)

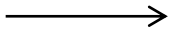
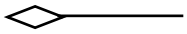
d. *Class* Diagram

Class Diagram menggambarkan struktur kelas, atribut, metode, dan hubungan antarkelas. Sandy et al. (2024:7) menjelaskan bahwa diagram ini memperlihatkan hubungan antar kelas serta rincian setiap kelas dalam rancangan sistem. Syahrizal Syahrizal (2025:159) menambahkan bahwa *Class* diagram digunakan untuk mendefinisikan struktur kelas yang akan dibuat.

Dalam penelitian ini, *Class* Diagram digunakan untuk menggambarkan struktur utama seperti *User*, Review, Prediction, Topic, dan komponen lain yang berkaitan dengan penyimpanan serta pengolahan data. Simbol hubungan antarkelas yang digunakan dalam *Class* Diagram ditunjukkan pada Tabel 2.5.

Tabel 2. 5 Simbol *Class* Diagram

Simbol	Elemen	Keterangan
	*Class*	Hempunan objek-objek dari berbagai atribut yang memiliki operasi yang sama
	*Nary Association*	Suatu upaya untuk menghindari asosiasi yang melebihi 2 objek.
	Assosiation	Relasi antar kelas dengan makna umum dan biasanya disertai multiplicity.

	*Directed Assosiation*	Relasi antara kelas dengan makna kelas yang satu digunakan oleh kelas lain.
	*Agregation*	Mengindikasikan keseluruhan bagian relationship disebut sebagai relasi
	*Composition*	Relasi composition terhadap *Class* tempat dia bergantung.
	Dependency	Menunjukan operasi pada suatu *Class* yang menggunakan *Class* yang lain

Sumber: Suharni et al. (2023:4)

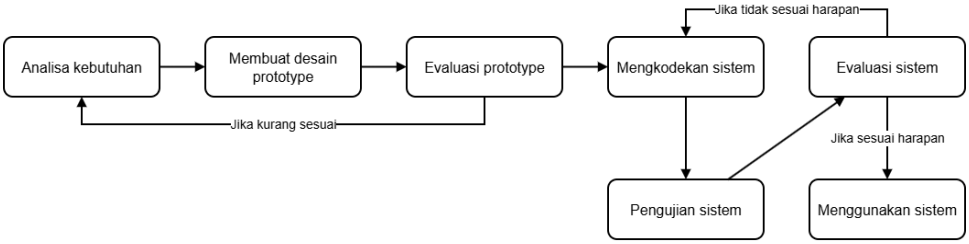
Tabel 2.5 menunjukkan simbol yang digunakan untuk menggambarkan struktur kelas. *Class* mewakili objek yang memiliki atribut dan operasi. Association menunjukkan hubungan umum, sedangkan Directed Association menunjukkan arah penggunaan satu kelas terhadap kelas lain. Aggregation menggambarkan hubungan bagian yang masih dapat berdiri sendiri, sementara Composition menunjukkan bagian yang sangat bergantung pada kelas utama. Dependency menunjukkan ketergantungan sementara dan N-ary Association digunakan apabila hubungan melibatkan lebih dari dua kelas. Simbol-simbol ini membantu menjelaskan hubungan data *User*, Review, Prediction, Topic, dan komponen lain pada sistem.

13. Model Prototype

Model Prototype merupakan metode pengembangan perangkat lunak yang dilakukan secara bertahap dan berulang melalui pembuatan bentuk awal sistem sebelum sistem dikembangkan secara keseluruhan. Prio et al. (2024) menjelaskan bahwa pendekatan prototipe membantu pengguna dan pengembang memahami kebutuhan sistem melalui bentuk awal yang dapat diamati secara langsung. Bentuk awal tersebut memungkinkan pengguna memberikan masukan, sedangkan pengembang dapat melakukan penyesuaian sebelum melanjutkan proses pengembangan sistem.

Menurut Andini et al. (2023) menjelaskan bahwa prototipe merupakan model awal sistem yang digunakan untuk menguji fungsi, mengevaluasi rancangan, dan memperoleh masukan sebelum versi akhir dikembangkan. Pendekatan ini memungkinkan kekurangan atau ketidaksesuaian sistem diketahui lebih awal sehingga perbaikan dapat dilakukan sebelum sistem diterapkan secara penuh.

Menurut Prio et al. (2024) tahapan Model Prototype meliputi analisis kebutuhan, pembuatan desain prototipe, evaluasi prototipe, pengkodean sistem, pengujian sistem, evaluasi sistem, dan penggunaan sistem. Proses tersebut dapat dilakukan secara berulang apabila hasil evaluasi menunjukkan bahwa prototipe atau sistem belum sesuai dengan kebutuhan. Alur Model Prototype ditunjukkan pada Gambar 2.6.



Gambar 2. 6 Metode Protoype
Sumber: Prio et al. (2024)

Gambar 2.6 menunjukkan bahwa proses pengembangan sistem dengan Model Prototype dimulai dari tahap analisis kebutuhan. Pada tahap ini dilakukan identifikasi terhadap kebutuhan pengguna, tujuan sistem, fungsi yang diperlukan, serta keluaran yang diharapkan. Hasil analisis tersebut kemudian digunakan sebagai dasar dalam membuat desain prototipe. Tahap selanjutnya adalah membuat desain prototipe yang menggambarkan bentuk awal sistem, seperti rancangan antarmuka, struktur menu, alur proses, dan fungsi utama. Desain prototipe kemudian dievaluasi untuk mengetahui kesesuaiannya dengan kebutuhan pengguna. Apabila prototipe masih kurang sesuai, proses dikembalikan ke tahap analisis kebutuhan agar kebutuhan dapat ditinjau kembali dan rancangan prototipe dapat diperbaiki.

Setelah prototipe dinilai sesuai, proses dilanjutkan ke tahap pengkodean sistem. Pada tahap ini, rancangan prototipe diterjemahkan menjadi perangkat lunak yang dapat dijalankan. Sistem yang telah dikembangkan kemudian memasuki tahap pengujian untuk memastikan bahwa setiap fungsi dapat berjalan sesuai dengan rancangan dan kebutuhan yang telah ditentukan.

Hasil pengujian selanjutnya diperiksa pada tahap evaluasi sistem. Apabila hasil evaluasi menunjukkan bahwa sistem belum sesuai dengan harapan, proses dikembalikan ke tahap pengkodean untuk dilakukan perbaikan. Sistem yang telah diperbaiki kemudian diuji dan dievaluasi kembali. Apabila hasil evaluasi telah sesuai dengan harapan, sistem dapat dilanjutkan ke tahap penggunaan oleh pengguna sesuai dengan tujuan pengembangannya.

14. Pengujian

Pengujian merupakan tahap untuk mengetahui apakah sistem dan model yang dikembangkan telah berjalan sesuai dengan tujuan penelitian. Pada penelitian ini, pengujian dilakukan menggunakan Black Box Testing untuk menilai fungsionalitas sistem serta Confusion Matrix untuk mengevaluasi kinerja model IndoBERT.

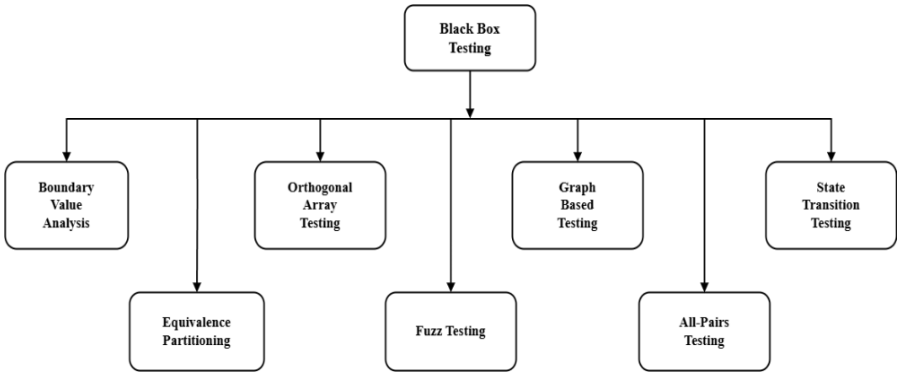
a. *Black Box Testing*

Black Box Testing merupakan pengujian yang berfokus pada fungsi sistem tanpa memeriksa struktur kode di dalamnya. Putu et al. (2022) menjelaskan bahwa pengujian dilakukan dengan memberikan masukan dan membandingkan keluaran sistem dengan hasil yang diharapkan.

Menurut Ayuningtyas & Rachmadi (2023) teknik Black Box Testing dapat menggunakan *Equivalence Partitioning* dan *Boundary Value Analysis* untuk mewakili kondisi masukan yang berbeda. Selviana

et al. (2023) menunjukkan bahwa pendekatan tersebut efektif untuk memeriksa fungsi aplikasi web berdasarkan kebutuhan pengguna.

Dalam penelitian ini, Black Box Testing digunakan untuk menguji menu *Login*, *Dashboard*, Live Scraping, AI Pipeline, Masukan Ulasan, dan *Logout*. Pengujian dilakukan melalui beberapa skenario untuk memastikan sistem menerima masukan, menjalankan proses, menampilkan hasil, dan memberikan validasi dengan benar. Contoh bentuk skenario pengujian Black Box yang digunakan sebagai acuan ditunjukkan pada Tabel 2.7.



Gambar 2. 7 Black Box Testing
Sumber: (Samuel, 2024)

Gambar 2.7 menunjukkan beberapa teknik dalam Black Box Testing, yaitu Boundary Value Analysis, Equivalence Partitioning, Orthogonal Array Testing, Fuzz Testing, Graph-Based Testing, All-Pairs Testing, dan State Transition Testing. Dari beberapa teknik tersebut, penelitian ini menggunakan *Equivalence Partitioning* dan *State Transition Testing* karena keduanya sesuai dengan skenario pengujian pada sistem yang dikembangkan.

Equivalence Partitioning digunakan untuk membagi masukan menjadi kelompok valid, tidak valid, dan kosong. Contohnya pada menu *Login*, pengujian dilakukan menggunakan akun yang benar, akun yang salah, akun tidak terdaftar, serta kolom email atau kata sandi yang tidak diisi. Teknik yang sama digunakan pada menu AI Pipeline dengan menguji teks ulasan yang diisi dan kolom ulasan yang dikosongkan.

State Transition Testing digunakan untuk menguji perubahan kondisi sistem setelah suatu tindakan dilakukan. Pada proses *Login*, kondisi berubah dari belum masuk menjadi berhasil masuk ke *Dashboard*. Pada proses *Logout*, kondisi berubah dari sesi aktif menjadi sesi berakhir. Teknik ini juga digunakan pada menu Live Scraping ketika status ulasan berubah menjadi selesai atau ditolak.

Kedua teknik tersebut tidak menggunakan rumus perhitungan khusus. Pengujian dilakukan dengan menjalankan setiap skenario dan membandingkan hasil sistem dengan hasil yang diharapkan. Selviana et al. (2023) menunjukkan bahwa pengujian Black Box dapat digunakan untuk memeriksa kesesuaian fungsi aplikasi web dengan kebutuhan pengguna. Dalam penelitian ini, pengujian diterapkan pada menu *Login*, *Dashboard*, Live Scraping, AI Pipeline, Masukan Ulasan, dan *Logout*. Hasil pengujian setiap fitur disajikan secara lengkap pada BAB IV. Hasil Black Box Testing dapat dihitung berdasarkan jumlah skenario yang memperoleh status valid dibandingkan dengan seluruh skenario yang

diuji. Persentase keberhasilan pengujian dapat dihitung menggunakan persamaan 2.4.

$$PK = \frac{SV}{TS} \times 100\% \quad (2.4)$$

b. *Confusion Matrix*

Confusion Matrix merupakan tabel yang membandingkan label aktual dengan hasil prediksi model. Sathyanarayanan & Tantri (2024) menjelaskan bahwa tabel ini membantu menunjukkan jumlah prediksi yang benar dan salah pada setiap kelas. Informasi tersebut membuat kesalahan model dapat diamati secara lebih rinci daripada hanya melihat nilai akurasi.

Penelitian ini menggunakan tiga kelas sentimen, yaitu positif, netral, dan negatif. Oleh karena itu, Confusion Matrix yang digunakan memiliki tiga baris untuk kelas aktual dan tiga kolom untuk kelas prediksi. Struktur matriks tersebut ditunjukkan pada Tabel 2.6.

Tabel 2. 6 Struktur Confusion Matrix

Aktual	Positif	Netral	Negatif
Positif	Benar sebagai Positif	Salah sebagai Netral	Salah sebagai Negatif
Netral	Salah sebagai Positif	Benar sebagai Netral	Salah sebagai Negatif
Negatif	Salah sebagai Positif	Salah sebagai Netral	Benar sebagai Negatif

Sumber: (Sathyanarayanan & Tantri, 2024)

Tabel 2.6 dibaca dengan mempertemukan baris kelas aktual dan kolom kelas prediksi. Sel diagonal utama berisi prediksi yang benar, yaitu Positif diprediksi Positif, Netral diprediksi Netral, dan Negatif diprediksi Negatif. Sel di luar diagonal menunjukkan kesalahan model.

Sebagai contoh, data aktual Positif yang berada pada kolom Netral berarti ulasan positif salah diprediksi sebagai netral. Melalui susunan ini, peneliti dapat melihat kelas yang paling sering diprediksi dengan benar dan pasangan kelas yang masih sering tertukar.

Pada klasifikasi multikelas, True Positive, False Positive, False Negatif, dan True Negatif dihitung secara terpisah untuk setiap kelas dengan pendekatan one-versus-rest. Arti setiap komponen serta metrik evaluasi yang digunakan ditunjukkan pada Tabel 2.7.

Tabel 2. 7 Metrik Evaluasi Kinerja Model

Komponen/Metrik	Keterangan
True Positive (TP_i)	Jumlah data kelas i yang diprediksi benar sebagai kelas i.
False Positive (FP_i)	Jumlah data kelas lain yang salah diprediksi sebagai kelas i.
False Negatif (FN_i)	Jumlah data kelas i yang salah diprediksi sebagai kelas lain.
True Negatif (TN_i)	Jumlah data selain kelas i yang diprediksi benar bukan sebagai kelas i.
Accuracy	Proporsi seluruh prediksi yang benar terhadap seluruh data uji.
Precision	Ketepatan model ketika memberikan prediksi pada suatu kelas.
Recall	Kemampuan model menemukan seluruh data aktual pada suatu kelas.
F1-Score	Rata-rata harmonik antara precision dan recall.
Macro Average	Rata-rata nilai metrik dari seluruh kelas tanpa mempertimbangkan jumlah data setiap kelas.

Sumber: (Kozak et al., 2022)

Tabel 2.7 menjelaskan bahwa TP_i adalah data kelas i yang diprediksi dengan benar. FP_i adalah data dari kelas lain yang keliru dimasukkan ke kelas i, sedangkan FN_i adalah data kelas i yang keliru dimasukkan ke kelas lain. TN_i merupakan data di luar kelas i yang juga diprediksi bukan sebagai kelas i. Nilai-nilai tersebut digunakan untuk

menghitung Accuracy, Precision, Recall, dan F1-Score. Macro Average kemudian digunakan untuk memperoleh rata-rata performa seluruh kelas tanpa memberikan bobot lebih besar kepada kelas yang memiliki data lebih banyak.

Menurut Kozak et al. (2022) performa model klasifikasi dapat dinilai menggunakan Accuracy, Precision, Recall, dan F1-Score. Selain itu, penelitian ini menggunakan Macro Average untuk merangkum nilai metrik dari tiga kelas sentimen.

Proporsi seluruh prediksi model yang benar dihitung menggunakan rumus Accuracy 2.5.

$$Accuracy = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah seluruh data}} \quad (2.5)$$

Pada rumus Accuracy, jumlah prediksi benar adalah seluruh data yang label prediksinya sama dengan label aktual. Nilai tersebut dibagi dengan jumlah seluruh data uji. Hasil yang semakin mendekati 1 atau 100% menunjukkan bahwa semakin banyak ulasan yang berhasil diklasifikasikan dengan benar secara keseluruhan.

Ketepatan model ketika memberikan prediksi pada kelas ke-i dihitung menggunakan rumus Precision 2.6.

$$Precision_{\text{i}} = \frac{TP_{\text{i}}}{TP_{\text{i}} + FP_{\text{i}}} \quad (2.6)$$

Pada rumus tersebut $\text{Precision}_{\text{i}}$, TP_{i} adalah jumlah data kelas i yang diprediksi dengan benar, sedangkan FP_{i} adalah data dari kelas lain yang salah diprediksi sebagai kelas i. Nilai Precision yang tinggi

menunjukkan bahwa ketika model menyebut suatu ulasan sebagai kelas tertentu, prediksi tersebut biasanya benar.

Kemampuan model menemukan seluruh data aktual pada kelas ke-i dihitung menggunakan rumus Recall 2.7.

$$Recall_{\mathrm{i}} = \frac{TP_{\mathrm{i}}}{TP_{\mathrm{i}} + FN_{\mathrm{i}}} \quad (2.7)$$

Pada rumus $\mathrm{Recall_i}$, $\mathrm{TP_i}$ merupakan data kelas i yang berhasil ditemukan dengan benar, sedangkan $\mathrm{FN_i}$ merupakan data kelas i yang terlewat karena diprediksi sebagai kelas lain. Nilai Recall yang tinggi menunjukkan bahwa model mampu menemukan sebagian besar ulasan yang benar-benar termasuk dalam kelas tersebut.

Keseimbangan antara Precision dan Recall pada kelas ke-i dihitung menggunakan rumus F1-Score berikut.

$$F1 - Score_{\mathrm{i}} = \frac{2 \times Precision_{\mathrm{i}} \times Recall_{\mathrm{i}}}{Precision_{\mathrm{i}} + Recall_{\mathrm{i}}} \quad (2.8)$$

Rumus $\mathrm{F1\text{-}Score_i}$ menggunakan rata-rata harmonik antara $\mathrm{Precision_i}$ dan $\mathrm{Recall_i}$. Metrik ini memberikan nilai yang baik apabila Precision dan Recall sama-sama tinggi. Oleh karena itu, F1-Score berguna ketika peneliti ingin menilai ketepatan prediksi sekaligus kemampuan model menemukan data pada suatu kelas.

Rata-rata performa ketiga kelas sentimen dihitung menggunakan rumus Macro Average 2.9

$$Macro\ Average = \frac{Metrik\ Positif + Metrik\ Netral + Metrik\ Negatif}{3} \quad (2.9)$$

Pada rumus Macro Average, nilai metrik untuk kelas Positif, Netral, dan Negatif dijumlahkan, kemudian dibagi tiga. Setiap kelas

memiliki pengaruh yang sama meskipun jumlah datanya berbeda. Dengan demikian, hasil evaluasi tidak hanya menggambarkan kelas yang memiliki data paling banyak.

Keempat metrik tersebut memberikan sudut pandang yang berbeda. Accuracy menilai ketepatan keseluruhan, Precision menilai ketepatan setiap prediksi kelas, Recall menilai kemampuan menemukan seluruh data aktual, dan F1-Score menilai keseimbangan Precision dengan Recall. Nilai per kelas dan Macro Average akan digunakan pada BAB IV untuk menjelaskan performa model IndoBERT secara lebih lengkap.

Berdasarkan Penjelasan Zeng (2025) *macro-averaging* menghitung rata-rata metrik dari setiap kelas tanpa mempertimbangkan jumlah sampel. Pendekatan ini sesuai untuk penelitian yang melibatkan kelas Positif, Netral, dan Negatif karena performa setiap kelas tetap dinilai secara seimbang.

B. Kajian Empiris

Penelitian mengenai analisis sentimen pada ulasan aplikasi telah banyak dilakukan untuk mengubah teks tidak terstruktur menjadi informasi yang lebih mudah dipahami. Salah satu model yang digunakan adalah IndoBERT karena mampu mempelajari hubungan antarkata berdasarkan konteks kalimat berbahasa Indonesia. Penerapan model tersebut umumnya dilakukan melalui proses *fine-tuning* menggunakan data ulasan yang telah memiliki label sentimen positif, netral, dan negatif. Salah satu penerapannya dilakukan pada ulasan aplikasi Tiket.com yang diperoleh dari Google Play Store dan menunjukkan

bahwa IndoBERT dapat digunakan untuk mengklasifikasikan sentimen pada ulasan aplikasi berbahasa Indonesia (Nuryadi et al., 2025).

Kajian dengan objek yang sama juga telah dilakukan terhadap ulasan pengguna aplikasi *Access by KAI* berbahasa Indonesia. Penelitian tersebut menggunakan pendekatan word *embedding* dan classical *Machine Learning* untuk melakukan analisis sentimen terhadap ulasan pengguna. Penelitian tersebut menunjukkan bahwa ulasan *Access by KAI* dapat dimanfaatkan sebagai sumber informasi untuk mengetahui tanggapan pengguna terhadap layanan aplikasi. Penelitian ini memiliki kesamaan pada objek yang dianalisis, tetapi menggunakan pendekatan yang berbeda karena model IndoBERT digunakan untuk memahami konteks kata dalam kalimat ulasan (Damanhuri & Husein, 2024)

Penggunaan IndoBERT pada ulasan aplikasi lain juga menunjukkan bahwa model tersebut dapat diterapkan pada teks yang diperoleh dari Google Play Store. Analisis terhadap ulasan aplikasi Instagram dilakukan menggunakan model IndoBERT pralatih untuk mengenali sentimen yang terkandung dalam teks berbahasa Indonesia Jansnio & Eka (2025). Penerapan yang serupa dilakukan pada ulasan aplikasi ChatGPT dengan melalui tahapan pembersihan teks, tokenisasi, normalisasi, penghapusan stopword, dan stemming. Tahapan pengolahan tersebut digunakan untuk mempersiapkan data sebelum diproses oleh model klasifikasi sentimen (Mursidah et al., 2025).

Kemampuan IndoBERT dalam melakukan klasifikasi dapat dikembangkan melalui penyesuaian model terhadap karakteristik dataset

tertentu. Proses *fine-tuning* digunakan untuk menyesuaikan pengetahuan bahasa yang telah dimiliki model dengan data ulasan yang menjadi objek penelitian. Penelitian sebelumnya juga menggabungkan IndoBERT dengan *Recurrent Convolutional Neural Network* atau R-CNN untuk mendukung proses klasifikasi sentimen ulasan berbahasa Indonesia. Kajian tersebut menunjukkan bahwa representasi kontekstual IndoBERT dapat digunakan secara mandiri maupun dikembangkan bersama arsitektur lain sesuai kebutuhan penelitian (Jayadianti et al., 2022).

Analisis sentimen tidak hanya dapat digunakan untuk menentukan sentimen secara umum, tetapi juga dapat dilengkapi dengan klasifikasi aspek. Pendekatan *Aspect-Based* Sentimen Analysis menggunakan IndoBERT digunakan untuk mengelompokkan isi ulasan berdasarkan aspek yang sedang dibahas oleh pengguna Dany et al. (2024). Pendekatan serupa juga diterapkan pada ulasan layanan hotel dengan menggabungkan hasil sentimen dan kategori aspek. Penggabungan tersebut dapat memberikan informasi yang lebih terperinci mengenai bagian layanan yang mendapatkan tanggapan dari pengguna dibandingkan hanya menentukan sentimen secara umum (Apriliani et al., 2025).

Berdasarkan penelitian terdahulu, analisis sentimen ulasan berbahasa Indonesia dapat dilakukan menggunakan metode classical *Machine Learning* maupun model bahasa IndoBERT. Penelitian sebelumnya telah membahas objek *Access by KAI*, penerapan IndoBERT pada ulasan aplikasi, proses *fine-tuning*, serta penggabungan klasifikasi sentimen dengan klasifikasi aspek. Penelitian ini menggabungkan bagian-bagian tersebut dalam satu sistem berbasis web.

IndoBERT digunakan untuk menentukan sentimen negatif, netral, dan positif, sedangkan *Rule Based Classification* digunakan untuk mengelompokkan ulasan ke dalam kategori *Login*, Sistem, Pembayaran, UI/UX, Performa, dan Lainnya. Hasil analisis kemudian dikelola melalui fitur Dashboard, Live Scraping, AI Pipeline, dan Masukan Ulasan agar *Admin* dapat memperoleh informasi ulasan pengguna *Access by KAI* secara lebih terstruktur.

C. Kerangka Berpikir

Ulasan pengguna aplikasi *Access by KAI* pada *Google Play Store* dapat digunakan sebagai sumber informasi untuk mengetahui pendapat pengguna terhadap kualitas aplikasi. Ulasan tersebut terdiri atas rating bintang dan teks komentar. Namun, jumlah ulasan yang terus bertambah, penggunaan bahasa yang beragam, serta bentuk teks yang tidak terstruktur menyebabkan proses analisis secara manual membutuhkan waktu yang lama dan berisiko menghasilkan penilaian yang tidak konsisten. Kondisi tersebut menunjukkan perlunya sistem yang mampu mengolah ulasan secara otomatis.

Permasalahan lain yang ditemukan adalah adanya kemungkinan ketidaksesuaian antara rating bintang dan isi teks ulasan. Sebagai contoh, pengguna dapat memberikan rating lima bintang, tetapi menuliskan komentar yang berisi keluhan atau penilaian negatif. Sebaliknya, pengguna dapat memberikan rating rendah, tetapi menuliskan komentar yang bernada positif. Apabila analisis hanya dilakukan berdasarkan rating, informasi sentimen yang dihasilkan dapat tidak sesuai dengan pendapat yang sebenarnya terdapat dalam

teks. Oleh karena itu, isi teks ulasan perlu dianalisis secara terpisah menggunakan model klasifikasi sentimen.

Berdasarkan kajian teoritis dan penelitian terdahulu, IndoBERT dipilih karena mampu memahami konteks teks berbahasa Indonesia, termasuk penggunaan singkatan, bahasa tidak baku, dan susunan kalimat yang beragam. Model IndoBERT digunakan untuk mengklasifikasikan teks ulasan ke dalam sentimen positif, netral, atau negatif. Nilai sentimen hasil model kemudian dapat dibandingkan dengan rating pengguna untuk membantu mengidentifikasi ulasan yang memiliki ketidaksesuaian antara rating bintang dan isi teks.

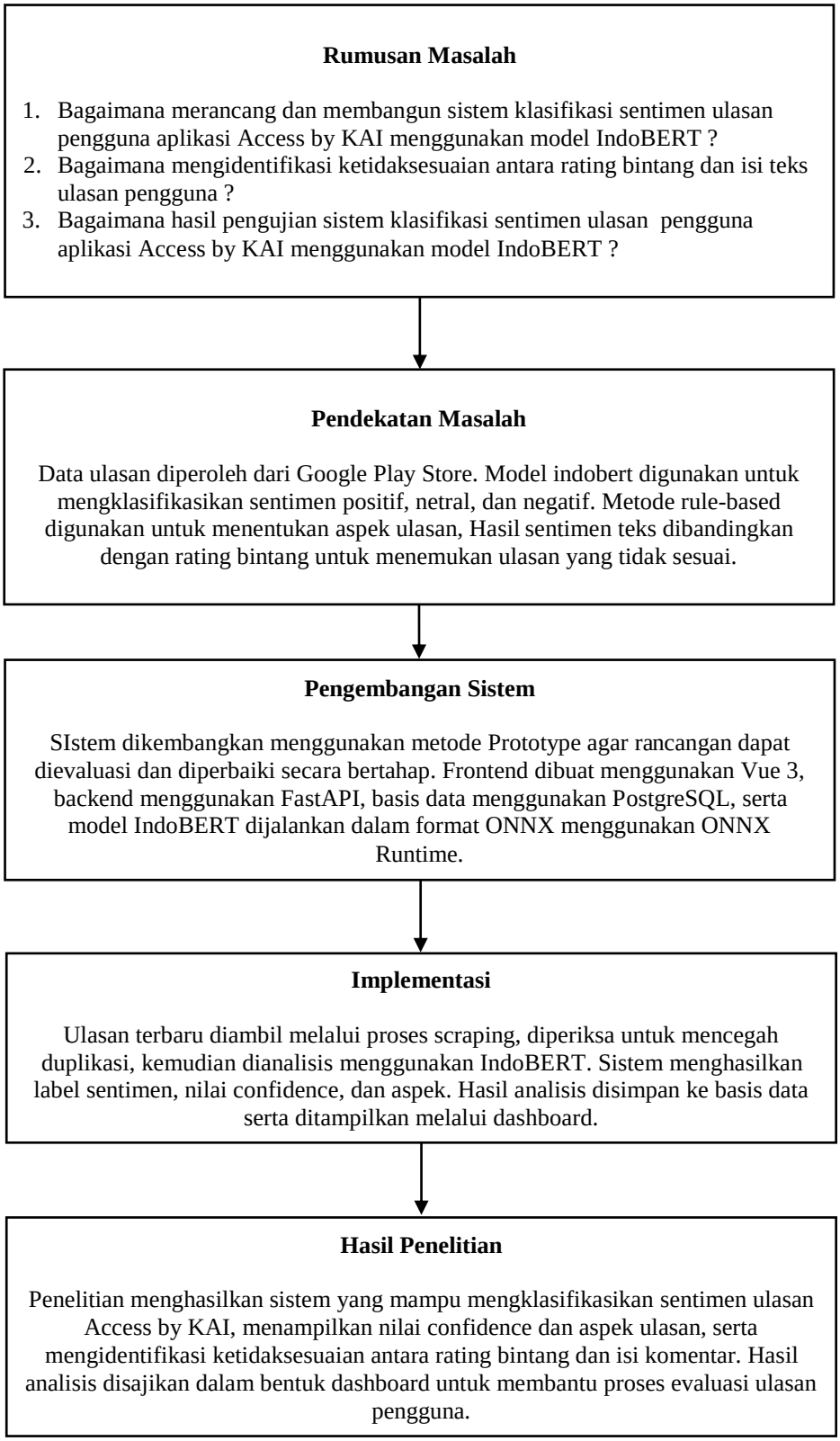
Selain klasifikasi sentimen, penelitian ini menggunakan pendekatan *Rule Based Classification* untuk mengelompokkan ulasan berdasarkan aspek yang dibahas. Kategori aspek tersebut meliputi *Login*, Sistem, Pembayaran, UI/UX, Performa, dan Lainnya. Pengelompokan aspek digunakan untuk memberikan informasi yang lebih terperinci mengenai bagian aplikasi yang sering memperoleh tanggapan, keluhan, atau apresiasi dari pengguna. Pendekatan penelitian ini sesuai dengan kajian sebelumnya pada laporan, yaitu menggabungkan IndoBERT untuk klasifikasi sentimen dan aturan kata kunci untuk klasifikasi aspek.

Data ulasan diperoleh dari *Google Play Store* menggunakan *Google Play Scraper*. Data yang terkumpul melalui proses pengembangan model dibersihkan dan disiapkan menjadi dataset berlabel untuk melakukan *fine-tuning* IndoBERT. Dataset dibagi menjadi data pelatihan, validasi, dan pengujian. Model hasil pelatihan dievaluasi menggunakan accuracy, precision, recall, F1-

score, dan confusion matrix. Model terbaik kemudian diekspor ke format ONNX agar dapat diterapkan pada *backend* sistem menggunakan ONNX *Runtime*.

Pada proses operasional sistem, ulasan terbaru diambil melalui fitur scraping. Teks ulasan diproses oleh model IndoBERT untuk menghasilkan label sentimen dan nilai confidence, sedangkan kategori aspek ditentukan menggunakan aturan kata kunci. Rating asli tetap disimpan sebagai data dari pengguna dan tidak digunakan sebagai satu-satunya penentu sentimen. Perbandingan antara rating dan sentimen teks digunakan untuk menandai ulasan yang berpotensi tidak konsisten, misalnya rating tinggi dengan teks negatif atau rating rendah dengan teks positif.

Sistem dikembangkan menggunakan metode Prototype agar rancangan dan fungsi sistem dapat dievaluasi serta diperbaiki secara bertahap. *Backend* dikembangkan menggunakan FastAPI, antarmuka menggunakan Vue 3 dan TypeScript, sedangkan PostgreSQL digunakan untuk menyimpan data ulasan dan hasil klasifikasi. Model IndoBERT diintegrasikan ke *backend* melalui ONNX Runtime. Sistem dirancang untuk menyediakan fitur pengambilan ulasan, klasifikasi sentimen, klasifikasi aspek, identifikasi ketidaksesuaian rating dan teks, penyajian *Dashboard* analitik, serta pembuatan laporan hasil analisis. Dashboard juga menyediakan visualisasi kata dominan untuk menampilkan kata yang paling sering muncul pada keseluruhan ulasan maupun berdasarkan sentimen negatif, netral, dan positif. Hubungan antara permasalahan, pendekatan, pengembangan, implementasi, pengujian, dan luaran penelitian ditunjukkan pada Gambar 2.8



Gambar 2.8 Diagram Kerangka Berpiki