

BAB II

KAJIAN LITERATUR

2.1 *Artificial Intelligence (AI)*

Artificial Intelligence (AI) merupakan bidang ilmu komputer yang berfokus pada pengembangan sistem yang mampu melakukan tugas yang umumnya membutuhkan kemampuan kognitif manusia, seperti memahami informasi, melakukan penalaran, mengenali pola, dan menghasilkan keputusan. AI memanfaatkan berbagai metode komputasi untuk memproses data serta menghasilkan keluaran berdasarkan pola atau informasi yang tersedia. Perkembangan AI memungkinkan penerapan teknologi tersebut pada berbagai bidang, termasuk pengolahan bahasa alami, sistem rekomendasi, dan aplikasi berbasis interaksi pengguna.

AI memiliki kemampuan untuk membantu proses pengolahan data dalam jumlah besar dan menjalankan tugas secara otomatis. Penerapan AI dapat meningkatkan efisiensi proses karena sistem mampu melakukan pemrosesan informasi tanpa harus menjalankan seluruh proses secara manual. Selain itu, kemampuan AI dalam mengenali pola dan memahami informasi dapat dimanfaatkan untuk membangun sistem yang memberikan keluaran sesuai dengan kebutuhan pengguna [9].

Dalam penelitian ini, AI diterapkan pada fitur pencarian kendaraan berbasis Chatbot AI. Pengguna dapat menyampaikan kebutuhan kendaraan menggunakan bahasa alami, kemudian sistem memproses permintaan tersebut untuk menghasilkan parameter pencarian kendaraan. Hasil pemrosesan selanjutnya digunakan untuk memperoleh kendaraan yang sesuai dengan kriteria pengguna dari database sistem.

2.1.1 Chatbot

Chatbot merupakan program komputer yang dirancang untuk melakukan interaksi dengan pengguna melalui percakapan menggunakan bahasa alami. Chatbot dapat menerima masukan pengguna dalam bentuk teks dan menghasilkan respons berdasarkan pemrosesan terhadap masukan tersebut. Penerapan teknologi Artificial Intelligence dan Natural Language Processing memungkinkan chatbot memahami masukan pengguna dengan cara yang lebih fleksibel dibandingkan sistem berbasis aturan sederhana.

Perkembangan chatbot dapat dibedakan berdasarkan teknologi yang digunakan. Chatbot berbasis aturan (*rule-based chatbot*) menggunakan aturan dan pola yang telah ditentukan untuk memberikan respons terhadap masukan tertentu, sedangkan chatbot berbasis AI memanfaatkan model pembelajaran mesin atau model bahasa untuk memahami masukan yang lebih beragam [13]. Perbedaan tersebut menyebabkan chatbot berbasis AI lebih sesuai untuk aplikasi yang membutuhkan interaksi menggunakan bahasa alami.

Dalam aplikasi jual beli kendaraan bekas, chatbot dapat digunakan sebagai media pencarian kendaraan secara konversasional. Pengguna dapat memasukkan kebutuhan seperti merek kendaraan, tahun, harga, jenis kendaraan, atau kriteria lainnya dalam satu kalimat. Sistem kemudian memproses informasi tersebut untuk menghasilkan hasil pencarian kendaraan yang sesuai dengan parameter yang berhasil dikenali.

2.1.2 Large Language Model / LLM

Large Language Model (LLM) merupakan model kecerdasan buatan yang dirancang untuk memahami dan menghasilkan bahasa manusia berdasarkan pola yang dipelajari dari data teks dalam jumlah besar. LLM menggunakan arsitektur berbasis *transformer* untuk memproses hubungan antarkata dan konteks dalam suatu masukan. Kemampuan tersebut memungkinkan LLM digunakan untuk berbagai tugas pengolahan bahasa alami, termasuk pemahaman teks, klasifikasi, ekstraksi informasi, dan generasi respons.

Pada penelitian ini, LLM digunakan sebagai komponen utama pada fitur Chatbot AI. LLM menerima masukan dalam bentuk bahasa alami dari pengguna dan membantu mengidentifikasi maksud serta parameter kendaraan yang terdapat pada masukan tersebut. Parameter yang diperoleh kemudian digunakan oleh sistem untuk melakukan pencarian kendaraan pada database. Model LLM yang digunakan dalam penelitian ini adalah [qwen/qwen3.6-27b, llama-3.3-70b-versatile, llama-3.1-8b-instant] yang diakses melalui **Groq API**. Model tersebut digunakan untuk memahami masukan pengguna, mengidentifikasi *intent*, dan mengekstraksi parameter pencarian kendaraan.

Secara algoritmis, proses utama yang dilakukan *chatbot* adalah:

1. *Intent recognition*: mengidentifikasi maksud pengguna (misalnya: pencarian mobil, *filter* harga, *filter* tahun, *filter* lokasi) berdasarkan teks input.
2. *Entity extraction*: mengekstrak parameter pencarian (misalnya merk, harga maksimal, tahun, lokasi) dari kalimat yang diketik pengguna.

3. *Query mapping*: mengonversi intent dan entity menjadi *query* pencarian terstruktur (misalnya SQL atau API *query*) ke *database* mobil dan motor bekas.
4. *Response generation*: mengambil hasil pencarian dari *database*, lalu mengolahnya menjadi kalimat yang mudah dipahami oleh pengguna dalam format percakapan.

2.1.3 Machine Learning

Machine Learning (ML) merupakan salah satu cabang dari kecerdasan buatan (Artificial Intelligence) yang memungkinkan sistem untuk belajar dari data dan meningkatkan kinerjanya tanpa harus diprogram secara eksplisit. *Machine Learning* digunakan dalam berbagai bidang, termasuk sistem rekomendasi, pengolahan bahasa alami (Natural Language Processing/NLP), analisis data, dan chatbot cerdas.

Dalam pengembangan aplikasi jual beli mobil dan motor bekas berbasis website, *Machine Learning* berperan sebagai teknologi pendukung yang memungkinkan chatbot memahami kebutuhan pengguna melalui bahasa alami. Dengan memanfaatkan data interaksi pengguna, sistem dapat memberikan hasil pencarian kendaraan yang lebih relevan sesuai dengan kata kunci, preferensi, atau pertanyaan yang diberikan pengguna.

Machine Learning memiliki beberapa pendekatan utama, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*. Pada sistem chatbot, pendekatan *supervised learning* banyak digunakan untuk melatih model dalam mengenali pola bahasa dan memahami maksud pengguna berdasarkan data yang telah diberi label. Sementara itu, *unsupervised learning* dapat digunakan untuk mengelompokkan data pengguna atau kendaraan yang memiliki karakteristik serupa sehingga mendukung proses rekomendasi.

Perkembangan *Machine Learning* juga mendorong munculnya teknologi *Large Language Model (LLM)* yang mampu memahami konteks percakapan secara lebih baik. Model ini memungkinkan chatbot memberikan respons yang lebih natural, relevan, dan informatif dibandingkan sistem pencarian berbasis kata kunci tradisional. Selain itu, *Machine Learning* juga menjadi dasar dalam pengembangan recommendation system yang membantu pengguna menemukan kendaraan sesuai kebutuhan dan preferensinya.

Dengan demikian, *Machine Learning* pada penelitian ini tidak digunakan untuk melakukan prediksi harga kendaraan, melainkan sebagai landasan teknologi yang mendukung fitur chatbot AI dan pencarian kendaraan berbasis bahasa alami sehingga dapat meningkatkan pengalaman pengguna dalam mencari mobil dan motor bekas secara lebih efektif.

2.1.4 Natural Language Processing (NLP)

Natural Language Processing (NLP) merupakan cabang dari AI yang berfokus pada interaksi antara komputer dan bahasa manusia. NLP memungkinkan sistem untuk memahami, memproses, dan menghasilkan bahasa alami. Dalam *chatbot AI* yang diterapkan dalam aplikasi jual beli kendaraan bekas, NLP digunakan untuk memahami maksud pengguna, mengekstraksi informasi penting, serta memberikan respons yang sesuai dengan konteks percakapan. NLP sangat penting dalam aplikasi ini untuk memproses *query* pengguna seperti “cari mobil Ford Tahun 2004 di bawah harga 180 juta dengan jarak tempuh dibawah 100000 km” untuk memberikan hasil pencarian yang relevan dan kontekstual [20].

Chatbot pencarian kendaraan bekas memanfaatkan *pipeline NLP* lengkap mulai dari tokenisasi (pemecahan teks menjadi token), *stemming* untuk normalisasi kata, hingga *Named Entity Recognition (NER)* guna mengidentifikasi entitas spesifik seperti merek "Ford", model "Everest", tahun "2004", atau jarak tempuh "100000". Intent recognition mendeteksi tujuan pengguna, misalnya "cari mobil" atau "bandingkan harga", sementara *entity extraction* menangkap parameter pencarian untuk *query database*. Komponen ini memastikan *chatbot* memahami variasi bahasa alami pengguna Indonesia.

Dalam perancangan aplikasi *website*, NLP diintegrasikan melalui *Natural Language Understanding (NLU)* untuk *parsing query* pencarian bebas teks, menghasilkan *filter* otomatis seperti harga, kondisi (bekas/baru), dan spesifikasi (manual/otomatis). *Chatbot* dapat menangani konteks *multi-turn*, seperti *follow-up* "tampilkan yang irit BBM" setelah pencarian awal, serta *Natural Language Generation (NLG)* untuk respons personal seperti "Mobil Ford Everest 2004 ditemukan 3 unit di Medan, harga mulai Rp 180 juta". Fitur ini meningkatkan *user experience* dibanding pencarian *keyword* tradisional.

NLP memungkinkan pencarian semantik yang akurat, mengurangi kesalahan interpretasi dan meningkatkan konversi penjualan hingga 30-40% pada platform *e-commerce* otomotif. Pengguna dapat berinteraksi secara *conversational*, mendapatkan rekomendasi berbasis preferensi ("mobil keluarga hemat bahan bakar"), dan integrasi dengan *database* iklan untuk *real-time matching*. Keamanan data terjaga melalui pemrosesan lokal, sesuai regulasi privasi di Indonesia [21].

2.2 Framework Pengembangan Website

Pengembangan aplikasi *website* untuk *marketplace* seperti jual beli mobil-motor bekas sering menggunakan pendekatan *cross-platform* untuk efisiensi biaya dan waktu. *Framework cross-platform* seperti *Flutter* dan *React Native* memungkinkan pengembangan

satu kode sumber untuk *multi-platform*. Hal ini bertujuan untuk mengurangi biaya hingga 40-50% dibanding *native development*. Flutter menggunakan Dart untuk UI kustom *pixel-perfect* dengan *rendering engine* sendiri, sementara React Native pakai JavaScript untuk akses komponen *native via bridge*. Keduanya *support hot reload* untuk percepatan *debugging*, ideal untuk *marketplace* dengan fitur pencarian *real-time* [22].

Tabel 2.1 Perbandingan Framework Utama

Framework	Bahasa	Performa	Kustomisasi UI
Flutter	Dart	Tinggi	Sangat Tinggi (<i>widget custom</i>)
React Native	JavaScript	Sedang	Sedang (komponen <i>native</i>)
Ionic	TypeScript	Tinggi	Tinggi (pendekatan <i>web-native</i>)
Xamarin	C#	Tinggi (mendekati asli)	Cukup Rendah
SwiftUI	Swift	Sedang (efisien dan responsif)	Sedang (deklaratif melalui <i>View Modifier</i>)

Flutter menawarkan performa superior dan konsistensi UI melalui *widget custom* tingkat tinggi. Flutter juga ideal untuk aplikasi *mobile trading* kendaraan bekas dengan tampilan dinamis. React Native unggul dalam *fleksibilitas integrasi native* dan pengembangan cepat berbasis JavaScript, sementara Ionic cocok untuk proyek *web-hybrid* dengan binding tinggi. Xamarin dengan bahasa pemrograman C# mendukung deklaratif efisien via XAML, dan SwiftUI memberikan modifikasi responsif khas iOS untuk pengalaman *native premium*.

2.3 Framework Scrum

Scrum merupakan *framework* Agile yang digunakan untuk mengembangkan produk secara iteratif dan bertahap. Scrum membagi proses pengembangan menjadi beberapa *Sprint* dengan durasi tertentu sehingga hasil pengembangan dapat dievaluasi secara berkala. Pendekatan tersebut memungkinkan perubahan kebutuhan dan masukan dari pengguna atau *stakeholder* dipertimbangkan selama proses pengembangan [23].

Scrum memiliki beberapa komponen utama yang terdiri atas *roles*, *events*, dan *artifacts*. *Roles* dalam Scrum meliputi *Product Owner*, *Scrum Master*, dan *Developers*, sedangkan *events* mencakup *Sprint*, *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*. *Artifacts* Scrum meliputi *Product Backlog*, *Sprint Backlog*, dan *Increment* [24].

Dalam penelitian ini, Scrum digunakan untuk mengatur tahapan pengembangan website AUTOMARKET. Pengembangan dilakukan melalui beberapa *Sprint* dengan durasi dua minggu dan setiap *Sprint* menghasilkan bagian sistem yang dapat diuji. Penggunaan Scrum membantu tim mengatur prioritas fitur, melakukan evaluasi hasil pengembangan, dan melakukan perbaikan berdasarkan hasil evaluasi.

Komponen inti Scrum mencakup:

1. *Roles*:

- a. ***Product Owner***, orang yang bertanggung jawab dalam mengelola *Product Backlog* dengan memprioritaskan item berdasarkan nilai bisnis, memastikan tim memahami kebutuhan pengguna, dan memaksimalkan nilai produk; peran ini tidak mengelola tim secara langsung melainkan fokus pada visi produk.
- b. ***Scrum Master***, orang yang bertindak sebagai fasilitator yang melindungi proses Scrum, menghilangkan hambatan tim, melatih anggota tentang praktik Scrum, dan memastikan penerapan nilai-nilai *Agile* tanpa menjadi otoritas hierarkis.
- c. ***Development Team***, tim lintas fungsi yang beranggotakan 3 sampai 9 orang yang *self-organizing* dan *self-managing*, bertanggung jawab menyelesaikan pekerjaan untuk menghasilkan *Increment* setiap *Sprint*, termasuk *developer*, *designer*, dan *tester* tanpa sub-tim [24].

2. *Events*:

- a. ***Sprint***, periode waktu tetap umunya 1 sampai 4 minggu di mana tim menghasilkan *Increment* yang *usable*, tidak boleh dibagi atau digabung, dan fokus pada satu hasil.
- b. ***Sprint Planning***, membahas apa yang akan dikerjakan dan bagaimana cara menyelesaikannya, melibatkan seluruh tim.
- c. ***Daily Scrum***, pertemuan singkat harian untuk sinkronisasi, menjawab tiga pertanyaan: “Apa yang dilakukan kemarin?”, “Rencana hari ini?”, dan “Hambatan apa?”. Bukan merupakan *status report* tapi komitmen ulang *Sprint Goal*.
- d. ***Sprint Review***, mendemonstrasikan *Increment* kepada *stakeholder* dengan maksimal durasi 4 jam, menyesuaikan *Product Backlog* berdasarkan *feedback*, dan berkolaborasi tentang apa yang akan dilakukan selanjutnya.
- e. ***Sprint Retrospective***, merefleksikan *sprint* yang berakhir untuk meningkatkan proses, orang, dan *tools* di *sprint* berikutnya, fokus pada “Apa yang baik?”, “Apa yang bisa diperbaiki”, dan “Apa yang akan dilakukan selanjutnya?”, Berlangsung sekitar 1-3 jam [25].

3. *Artifacts*:

- a. **Product Backlog**, daftar terurut, hidup, dan dinamis dari semua fitur, *enhancements*, *bug fixes*, dll. Dikelola oleh *Product Owner*. Item yang berada diatas lebih detail dan memiliki prioritas tinggi, berevolusi sepanjang proyek.
- b. **Sprint Backlog**, sub-set *Product Backlog* yang dipilih untuk *sprint* tertentu, plus rencana *actionable* dan *Sprint Goal*. Dimiliki oleh *Development Team*, diupdate harian dan memiliki fokus pada "*Definition of Done*".
- c. **Increment**, jumlah semua *Product Backlog Item* selesai selama *Sprint* plus *Increment* sebelumnya, harus "*Done*" dan *potentially shippable*, memberikan transparansi kemajuan menuju *Product Goal* [26].

2.4 *Application Programming Interface (API)*

Application Programming Interface (API) merupakan mekanisme yang memungkinkan suatu aplikasi berkomunikasi dengan aplikasi atau layanan lainnya. API menyediakan aturan mengenai bagaimana permintaan dikirim dan bagaimana respons diberikan oleh suatu layanan. Penggunaan API memungkinkan setiap komponen aplikasi dikembangkan secara terpisah tetapi tetap dapat berkomunikasi satu sama lain.

Pada website AUTOMARKET, API digunakan sebagai penghubung antara frontend dengan backend. Frontend mengirimkan permintaan kepada backend, kemudian backend melakukan proses bisnis, mengakses database, atau menghubungi layanan eksternal sesuai kebutuhan. Hasil pemrosesan kemudian dikirim kembali kepada frontend dalam bentuk data yang dapat ditampilkan kepada pengguna.

API juga digunakan untuk menghubungkan backend dengan layanan LLM yang digunakan pada fitur Chatbot AI. Ketika pengguna mengirimkan pertanyaan melalui chatbot, backend mengirimkan permintaan kepada layanan LLM melalui API. Respons dari LLM kemudian diproses oleh backend sebelum digunakan untuk menghasilkan hasil pencarian kendaraan.

Tabel 2.2 Contoh Endpoint API

Endpoint	Method	Deskripsi	Contoh Response
/api/v1/vehicles	GET	Daftar kendaraan bekas	{id:1, merk:"Toyota", harga:150000000}

/api/v1/vehicles/search	POST	Pencarian dengan filter chatbot	{results: [...kendaraan matches...]}
/api/v1/chatbot/query	POST	Proses query AI untuk rekomendasi	{recommendations: [mobil1, mobil2]}
/api/v1/users/profile	GET	Profil penjual/pembeli	{nama: "User", favorit: [id_kendaraan]}

API memungkinkan akses *cross-platform (website)* untuk fitur seperti upload foto kendaraan dan integrasi pembayaran. Dengan *chatbot*, API menyediakan rekomendasi personal berdasarkan preferensi pengguna, meningkatkan konversi penjualan hingga 24/7. Keamanan dijamin melalui HTTPS, rate limiting, dan validasi input, sesuai *best practices e-commerce* [27].

2.5 Sistem Jual Beli Kendaraan Bekas Berbasis Digital

Sistem jual beli kendaraan bekas berbasis digital merupakan sistem yang memanfaatkan teknologi informasi untuk membantu proses pencarian, penyajian informasi, dan pemasaran kendaraan bekas melalui platform digital. Sistem tersebut memungkinkan informasi kendaraan seperti merek, tipe, tahun, harga, dan kondisi kendaraan disajikan kepada calon pembeli. Digitalisasi proses tersebut dapat memberikan alternatif bagi pengguna untuk memperoleh informasi kendaraan tanpa harus melakukan pencarian secara konvensional.

Pada website jual beli kendaraan bekas, fitur pencarian menjadi salah satu komponen penting karena jumlah kendaraan yang tersedia dapat menyulitkan pengguna dalam menemukan kendaraan yang sesuai. Fitur pencarian dapat menggunakan parameter seperti merek, jenis kendaraan, tahun, harga, dan lokasi. Penerapan Chatbot AI pada penelitian ini digunakan untuk membantu pengguna menyampaikan kriteria pencarian menggunakan bahasa alami.

Website AUTOMARKET menyediakan informasi kendaraan yang dapat digunakan pengguna untuk menentukan pilihan kendaraan. Pengguna dapat melakukan pencarian kendaraan melalui parameter yang tersedia maupun menggunakan Chatbot AI. Setelah kendaraan ditemukan, pengguna dapat melihat informasi detail kendaraan dan menghubungi

penjual melalui WhatsApp. Platform ini akan dilengkapi dengan fitur-fitur canggih untuk mendukung transaksi yang aman dan lancar, diantaranya:

1. **Pencarian dan *Filter Kendaraan*:** Pengguna dapat mencari berdasarkan merek, model, tahun produksi, harga, kilometer tempuh, atau lokasi geografis, dengan algoritma rekomendasi berbasis AI.
2. **Detail dan Inspeksi Virtual:** Tampilan 360 derajat, foto *high-resolution*, serta video *walkthrough* untuk menjunjung meningkatkan detail fisik kendaraan.

Dengan diterapkannya fitur-fitur seperti yang dijelaskan di atas, sangat diharapkan dapat memberikan dampak yang signifikan bagi semua pihak. Para calon pembeli diharapkan dapat menerima kemudahan dalam mengakses informasi sehingga calon pembeli tidak perlu lagi menghabiskan waktu untuk membandingkan ribuan daftar kendaraan yang tersedia. Data riwayat kendaraan yang tersedia juga dapat dilacak melalui API, sehingga meminimalisir adanya tindakan penipuan [29].

2.6 Sistem Rekomendasi

Sistem rekomendasi merupakan sistem yang digunakan untuk memberikan saran terhadap item tertentu berdasarkan informasi atau kriteria yang tersedia. Pada sistem jual beli kendaraan, rekomendasi dapat digunakan untuk membantu pengguna menemukan kendaraan yang sesuai dengan kebutuhan. Informasi kendaraan seperti merek, tipe, tahun, harga, dan lokasi dapat digunakan sebagai dasar dalam menentukan hasil pencarian atau rekomendasi.

Sistem rekomendasi dapat menggunakan beberapa pendekatan, seperti *content-based filtering*, *collaborative filtering*, dan *hybrid filtering*. *Content-based filtering* menggunakan karakteristik item untuk menentukan item yang memiliki kesesuaian dengan kebutuhan pengguna, sedangkan *collaborative filtering* menggunakan pola interaksi dari pengguna. *Hybrid filtering* menggabungkan lebih dari satu pendekatan untuk memperoleh hasil yang lebih sesuai.

Pada penelitian ini, pencarian berbasis Chatbot AI menggunakan informasi yang diperoleh dari masukan pengguna untuk menentukan parameter pencarian kendaraan. Parameter tersebut kemudian digunakan untuk memperoleh kendaraan yang sesuai dari database. Dengan demikian, sistem tidak melakukan pelatihan model rekomendasi secara

mandiri, tetapi memanfaatkan LLM untuk memahami permintaan pengguna dan sistem database untuk memperoleh kendaraan berdasarkan parameter yang dihasilkan.

2.6.1 Jenis-Jenis Sistem Rekomendasi

Terdapat beberapa pendekatan utama dalam sistem rekomendasi, yang dapat dikombinasikan untuk mendapatkan hasil optimal:

- 1. **Content-Based Filtering:** Rekomendasi dibuat berdasarkan kesamaan fitur item dengan preferensi pengguna sebelumnya. Misalnya, jika pengguna mencari mobil SUV Toyota tahun 2018 dengan budget Rp150 juta, sistem akan merekomendasikan kendaraan serupa berdasarkan atribut seperti transmisi otomatis, kilometer rendah, dan warna favorit.
- 2. **Collaborative Filtering:** Mengandalkan perilaku pengguna lain yang mirip. Contoh: Pengguna A yang membeli Honda Civic didasarkan ke pengguna B dengan riwayat pencarian serupa, karena pengguna lain dengan profil mirip juga menyukai model tersebut. Pendekatan ini dibagi menjadi user-based (kesamaan berdasarkan pengguna) dan item-based (kesamaan berdasarkan benda).
- 3. **Hybrid Filtering:** Kombinasi *content-based* dan *collaborative* untuk mengatasi *cold-start problem* (kurangnya data pengguna baru). Dalam aplikasi ini, *Hybrid Filtering* dapat meningkatkan presisi hingga 20%-30% [31].

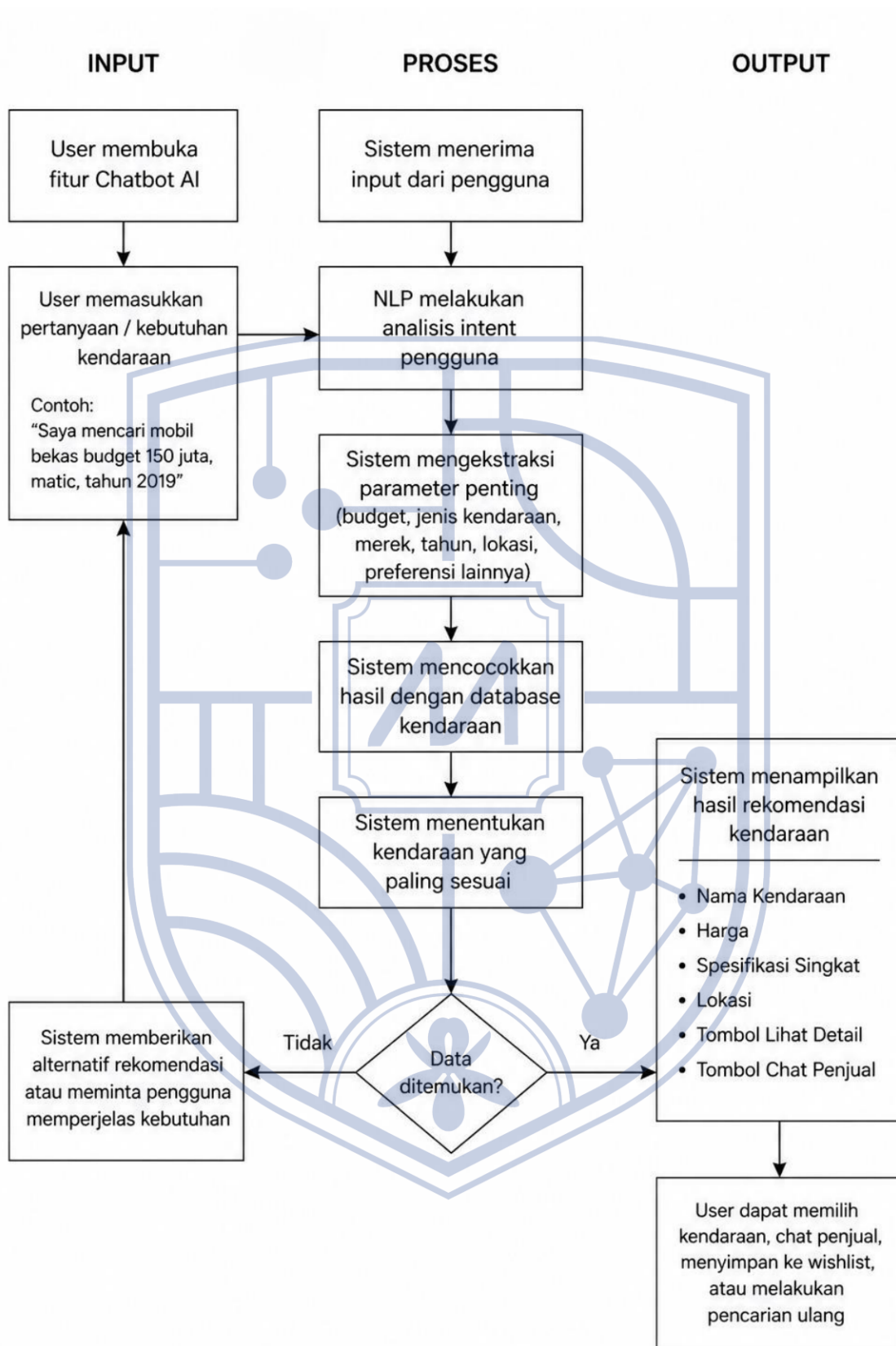
Tabel 2.3 Perbandingan Jenis Sistem Rekomendasi

Jenis	Kelebihan	Kekurangan	Contoh Aplikasi
<i>Content-Based</i>	Personalisasi tinggi berdasarkan fitur	Membutuhkan profil item lengkap	Netflix (rekomendasi film genre)
<i>Collaborative</i>	Temuan tak terduga (<i>serendipity</i>)	<i>Cold-start</i> dan <i>sparsity data</i>	Amazon (produk populer pengguna lain)
<i>Hybrid</i>	Akurasi tinggi, skalabel	Kompleksitas komputasi	Spotify (musik campuran)

2.6.2 Arsitektur Sistem Rekomendasi pada Aplikasi Jual Beli Kendaraan Bekas

Arsitektur dirancang untuk aplikasi *website* dengan fitur pencarian berbasis *Chatbot AI*, menggunakan backend seperti Node.js atau Python (Flask/Django) dan database PostgreSQL (Supabase) untuk data kendaraan dinamis. Proses utama meliputi:

1. **Pengumpulan Data:** Data pengguna (riwayat pencarian, *like/dislike*), data item (spesifikasi kendaraan: merek, harga, tahun, kilometer, lokasi), dan interaksi (*rating*, *chat log* dari *chatbot*).
2. **Preprocessing:** Normalisasi data harga (misal, *log* transformasi untuk distribusi skewed) dan Vektorisasi fitur menggunakan TF-IDF untuk deskripsi kendaraan [32].
3. **Model Training:**
 - a. **Content-based:** Persamaan *Cosinus* pada vector fitur,
$$(u, i) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$
 - b. **Collaborative:** Matrix faktorisasi dengan *Singular Value Decomposition* (SVD), memprediksi *rating*
$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T P_u$$
 - c. **Integrasi Chatbot:** Chat AI (*GPT*) mengekstrak inputan dan memicu rekomendasi secara real-time.
4. **Generasi Rekomendasi:** Top-N item disortir berdasarkan skor prediksi, ditampilkan via API RESTful ke *frontend* React Native (*mobile*) atau Vue.js (*web*) [33].
5. **Evaluasi:** Menggunakan metrik Precision@K, Recall@K, dan RMSE. Contoh: Precision@5 > 0.8 berarti 80% dari 5 rekomendasi pertama relevan [34].



Gambar 2.1 Flowchart Sistem Rekomendasi

2.7 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan metode yang digunakan untuk mengukur persepsi pengguna terhadap kemudahan penggunaan suatu sistem. SUS menggunakan sepuluh pernyataan dengan skala penilaian lima tingkat, yaitu dari sangat tidak setuju sampai sangat setuju. Metode tersebut dapat digunakan untuk memperoleh nilai kuantitatif yang menggambarkan tingkat usability suatu sistem.

Setiap pernyataan SUS memiliki aturan perhitungan yang berbeda berdasarkan nomor pernyataan. Pernyataan bernomor ganjil dihitung dengan mengurangi nilai jawaban dengan angka satu, sedangkan pernyataan bernomor genap dihitung dengan mengurangi nilai jawaban dari angka lima. Seluruh nilai hasil penyesuaian kemudian dijumlahkan dan dikalikan dengan 2,5 untuk menghasilkan skor SUS pada rentang 0 sampai 100.

Rumus perhitungan

Untuk pernyataan bernomor ganjil:

$$X_i = R_i - 1 \quad (2.1)$$

Untuk pernyataan bernomor genap:

$$X_i = 5 - R_i \quad (2.2)$$

Nilai SUS responden dihitung menggunakan:

$$SUS = (i=1 \sum 10 X_i) \times 2,5 \quad (2.3)$$

Nilai SUS rata-rata seluruh responden dihitung dengan:

$$SUS_{rata-rata} = \sum SUS_i / n \quad (2.4)$$

Keterangan:

- R_i = nilai jawaban responden pada pernyataan ke- i
- X_i = nilai hasil penyesuaian
- n = jumlah responden

SUS digunakan dalam penelitian ini untuk mengevaluasi tingkat kemudahan penggunaan website AUTOMARKET berdasarkan persepsi pengguna. Hasil SUS kemudian digunakan sebagai salah satu indikator evaluasi usability sistem. Daftar sepuluh pernyataan SUS yang digunakan dalam penelitian harus dilampirkan pada bagian instrumen penelitian.

2.8 Black Box Testing

Black Box Testing merupakan metode pengujian perangkat lunak yang berfokus pada fungsi sistem berdasarkan masukan dan keluaran yang dihasilkan. Pengujian dilakukan tanpa memperhatikan struktur kode internal sistem. Setiap fungsi diuji berdasarkan kebutuhan yang telah ditentukan untuk memastikan sistem menghasilkan keluaran yang sesuai.

Pada penelitian ini, Black Box Testing digunakan untuk menguji fungsi utama website AUTOMARKET. Pengujian mencakup fungsi autentikasi, pencarian kendaraan, pengelolaan kendaraan, penggunaan chatbot, dan fungsi lain yang tersedia pada sistem. Hasil pengujian dicatat berdasarkan kesesuaian antara keluaran aktual dan keluaran yang diharapkan.

2.9 Pengujian Chatbot AI

Pengujian Chatbot AI dilakukan secara khusus untuk mengetahui kemampuan sistem dalam memahami masukan pengguna dan menghasilkan parameter pencarian yang sesuai. Pengujian tidak hanya melihat apakah fitur dapat dijalankan, tetapi juga mengevaluasi ketepatan *intent* dan parameter kendaraan yang diperoleh dari masukan pengguna. Pengujian tersebut diperlukan karena keberhasilan fungsi chatbot tidak sepenuhnya dapat direpresentasikan hanya melalui pengujian fungsional dan usability.

Pengujian dilakukan menggunakan beberapa skenario masukan yang mewakili kebutuhan pencarian kendaraan. Parameter yang diuji dapat meliputi *intent*, merek, tipe kendaraan, tahun, harga, lokasi, dan parameter lain yang memang didukung oleh sistem. Hasil aktual dari chatbot dibandingkan dengan hasil yang seharusnya diperoleh berdasarkan skenario pengujian.

Tabel 2.4 Skenario Pengujian Chatbot AI

No	Jenis Pengujian	Contoh Input	Parameter yang Diharapkan
1	Intent pencarian	“Saya mencari mobil keluarga”	Intent pencarian kendaraan
2	Merek	“Cari Toyota”	Merek = Toyota
3	Harga	“Mobil di bawah 200 juta”	Harga maksimum = Rp200.000.000
4	Tahun	“Mobil tahun 2020 ke atas”	Tahun minimum = 2020
5	Lokasi	“Cari mobil di Medan”	Lokasi = Medan

No	Jenis Pengujian	Contoh Input	Parameter yang Diharapkan
6	Multi-parameter	“Toyota Avanza 2020 di bawah 200 juta”	Merek, tipe, tahun, harga
7	Bahasa alami	“Saya butuh mobil keluarga yang murah”	Intent + parameter relevan
8	Input di luar konteks	Pertanyaan di luar pencarian kendaraan	Respons sesuai batasan chatbot

Jika data pengujian aktual sudah tersedia, hasilnya dapat dihitung menggunakan:

$$\text{Accuracy} = \text{Jumlah hasil yang benar} / \text{Jumlah seluruh pengujian} \times 100\% \quad (2.5)$$

Hasil pengujian chatbot digunakan untuk mengetahui kemampuan fitur AI dalam memahami kebutuhan pencarian kendaraan. Nilai tersebut kemudian dibahas bersama hasil Black Box Testing dan SUS untuk memberikan evaluasi yang lebih menyeluruh terhadap sistem.