



Integrasi Qdrant Vector Database dan DeepSeek AI Untuk Chatbot Otomatis pada Aplikasi E-Commerce

Azhar Abdurrafiq Sujana^{*1}, Indra Yustiana², Alun Sujjada³

Email: ¹azhar.abdurrafiq_ti20@nusaputra.ac.id, ²indra.yustiana@nusaputra.ac.id, ³alun.sujjada@nusaputra.ac.id

¹Teknik Informatika, Teknik Komputer dan Desain, Universitas Nusa Putra Sukabumi

²Teknik Informatika, Teknik Komputer dan Desain, Universitas Nusa Putra Sukabumi

³Teknik Informatika, Teknik Komputer dan Desain, Universitas Nusa Putra Sukabumi

Diterima: 09 Juli 2025 | Direvisi: - | Disetujui: 31 Agustus 2025

©2020 Program Studi Teknik Informatika Fakultas Ilmu Komputer,
Universitas Muhammadiyah Riau, Indonesia

Abstrak

Transformasi digital telah mendorong *e-commerce* untuk meningkatkan kualitas layanan pelanggan (*customer service*). Salah satu solusi yang ditawarkan yaitu munculnya teknologi chatbot sebagai program berbasis kecerdasan buatan yang mampu berinteraksi secara otomatis dengan pengguna. Penelitian ini akan mengembangkan chatbot dengan memanfaatkan Qdrant Vector Database sebagai penyimpanan yang berfungsi untuk menyimpan dan mencari informasi berbasis vektor, dan OpenRouter API Key model DeepSeekAI sebagai akses chatbot. Sistem ini dirancang untuk menjawab pertanyaan umum terkait informasi produk, ketersediaan stok, harga barang, status pengiriman dan pertanyaan lainnya yang ada sangkut pautnya dengan *e-commerce* dalam 24 jam. Hasil penelitian menunjukkan bahwa sistem dapat membantu meningkatkan layanan otomatis kepada pengguna untuk mencari informasi produk, khususnya diluar jam kerja.

Kata kunci: *chatbot, deepseek AI, qdrant vector database, e-commerce*

Integration of Qdrant Vector Database and DeepSeek AI for Automated Chatbots on E-Commerce Applications

Abstract

Digital transformation has encouraged e-commerce to improve customer service quality. One solution offered is the emergence of chatbot technology as an artificial intelligence-based program capable of interacting automatically with users. This research will develop a chatbot by utilizing the Qdrant Vector Database as storage for storing and searching vector-based information, and the OpenRouter API Key model DeepSeekAI as chatbot access. This system is designed to answer general questions related to product information, stock availability, product prices, delivery status, and other e-commerce-related inquiries within 24 hours. The research results indicate that the system can help improve automated services for users to search for product information, particularly outside of regular business hours.

Keywords: *chatbot, deepseek AI, qdrant vector database, e-commerce*

1. PENDAHULUAN

Transaksi digital merupakan bukti dari berkembangnya teknologi yang memungkinkan aktivitas jual beli tidak lagi dilakukan dengan cara yang tradisional. Menurut We Are Social (2024) [1] terdapat lebih dari 55% pengguna internet melakukan pembelian

secara daring. Data ini menunjukkan bahwa transaksi digital telah menjadi bagian dari kehidupan sehari – hari masyarakat modern dan mendorong pelaku bisnis untuk berinovasi dalam meningkatkan kualitas layanan.

E-commerce menjadi salah satu layanan jual beli daring telah mengalami perkembangan pesat dalam beberapa tahun terakhir. Studi lain mengatakan, bahwa sistem *e-commerce* [2] tidak hanya sekedar cakupan perniagaan saja, tetapi mencakup pengkolaborasi bisnis, pelayanan nasabah, lowongan pekerjaan [3]. Seiring dengan meningkatnya jumlah transaksi dan kebutuhan informasi secara *real-time*, tuntutan akan layanan pelanggan yang otomatis dan dapat diakses dalam 24/7 juga semakin tinggi. Solusi yang dapat ditawarkan yaitu menjadikan sistem chatbot [4] sebagai media sumber informasi yang dapat diakses pengguna untuk otomatisasi layanan dalam 24/7.

Chatbot merupakan program komputer yang dirancang untuk mensimulasikan percakapan dengan pengguna secara otomatis layaknya percakapan manusia [5]. Dalam penelitian ini, sistem chatbot dikembangkan dengan memanfaatkan Qdrant sebagai *vector database* untuk menyimpan dan melakukan pencarian informasi berbasis konteks. Selain itu, digunakan OpenRouter API yang terintegrasi dengan model DeepSeek AI sebagai inti dari sistem chatbot, guna menghasilkan respons yang lebih natural dan relevan terhadap pertanyaan pengguna.

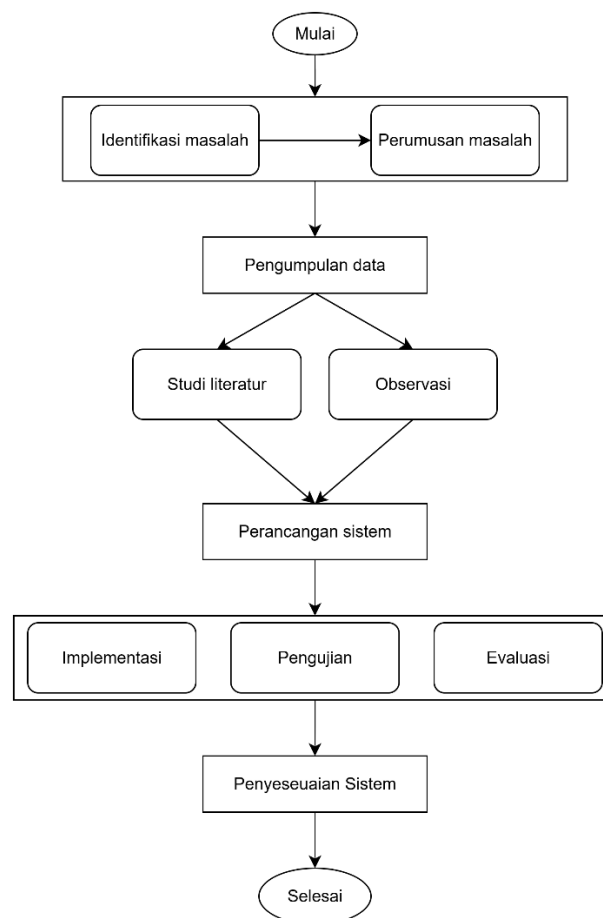
Penelitian yang dilakukan oleh [5], [6], [7]. Mereka menunjukkan keberhasilan sistem chatbot yang dikembangkan memanfaatkan kecerdasan buatan dapat mengurangi beban pekerjaan manusia hingga 50%.

Berdasarkan beberapa penelitian sebelumnya. Solusi yang diusulkan pada penelitian ini yaitu dengan menerapkannya Qdrant *vector database* dan *openrouter* DeepSeek AI sebagai sistem chatbot yang dapat membantu pengguna dalam mencari informasi secara otomatis dengan tingkat akurasi jawaban yang tinggi.

2. METODE PENELITIAN

2.1. Tahapan Penelitian

Terdapat beberapa tahapan yang akan dilakukan dalam proses penelitian ini dari awal hingga akhir. Proses penelitian akan dijabarkan dalam gambar berikut:



Gambar 1. Tahapan Penelitian

2.2. Identifikasi Masalah

Menurut [8] salah satu permasalahan yang ada pada sistem *e-commerce* yaitu keterbatasannya sumber data manusia untuk menangani masalah *customer service* yang masih konvensional. Ketika terdapat banyak pengguna yang mengakses secara bersamaan, akan mengakibatkan keterlambatan memberikan jawaban.

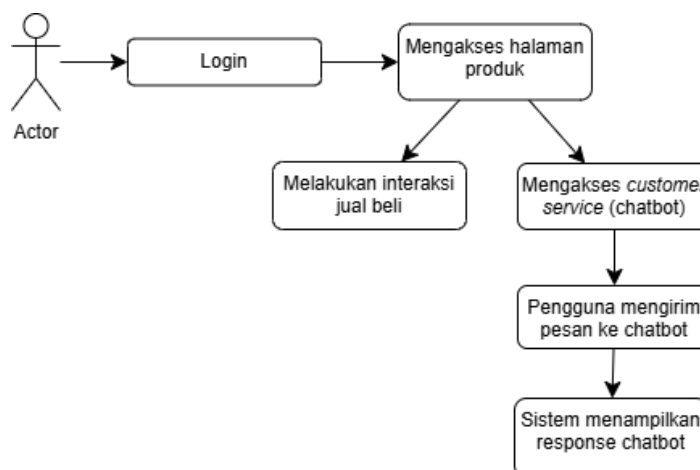
Berdasarkan permasalahan tersebut, penelitian ini akan merancang sebuah sistem chatbot yang dikaitkan dengan *platform e-commerce* guna membantu mengatasi permasalahan tersebut dan dapat memberikan layanan informasi kepada pengguna dalam 24/7 secara penuh.

2.3. Teknik Pengumpulan Data

1. Studi Literatur dilakukan untuk mengkaji terhadap referensi – referensi yang relevan dengan penelitian yang akan dilakukan. Referensi ini bisa bersumber dari artikel, jurnal, buku ataupun halaman website [9] yang memberikan informasi mengenai teknologi chatbot, seputar *e-commerce* serta tantangannya, gambaran mengenai NLP dan LLM serta tidak lupa mencari – cari informasi mengenai cara penggunaan *Qdrant Vector Database* dan *DeepSeek AI* sebagai *handle* chatbot.
2. Proses observasi dilakukan peneliti dengan cara mengamati pada sistem *e-commerce* mengenai pola layanan pelanggan yang ada untuk mengidentifikasi kebutuhan dan permasalahan yang bisa diselesaikan dengan sistem chatbot.
3. Data yang digunakan untuk melakukan pencarian produk dalam pemanfaatan *Qdrant* dan *DeepSeek AI* pada sistem chatbot merupakan data *dummy* yang diambil melalui *generate image*. Proses dilakukan pada saat menambahkan produk pada sistem *e-commerce*. Data akan tersimpan di database (*MongoDB*) dan *Qdrant* pada koleksi.

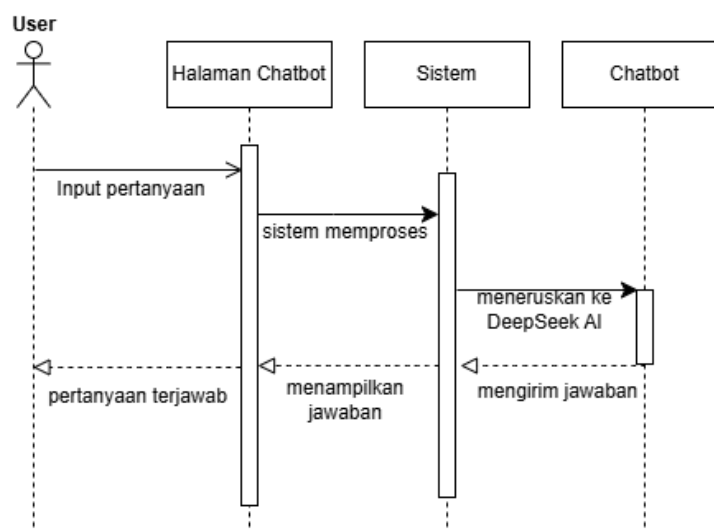
2.4. Perancangan Sistem

1. Diagram Use Case



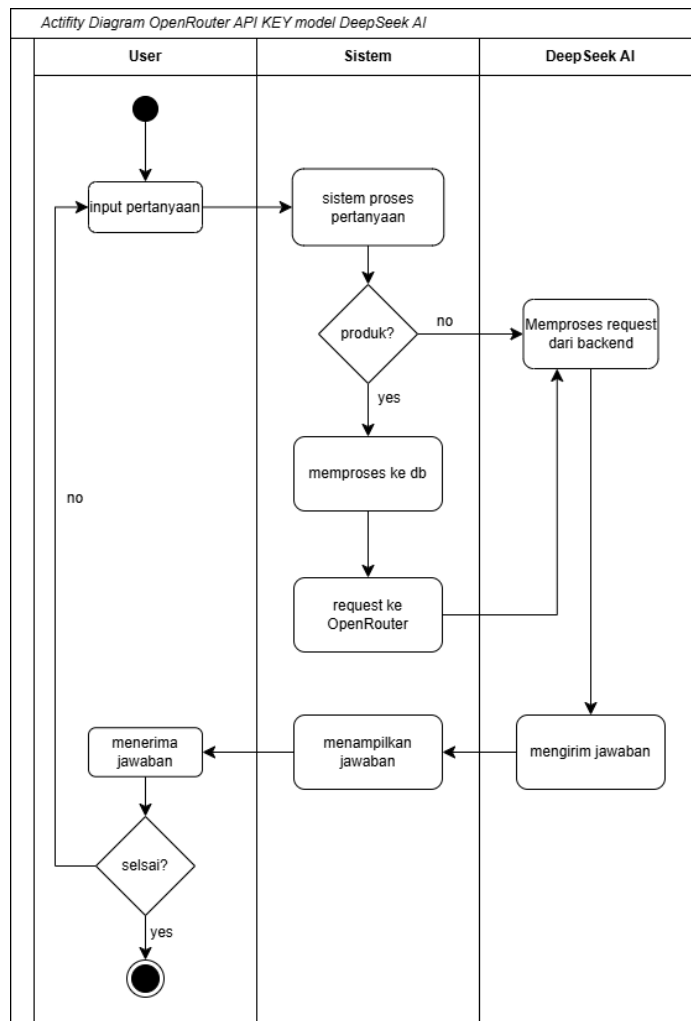
Gambar 2. Diagram Use Case

2. Diagram Sequence



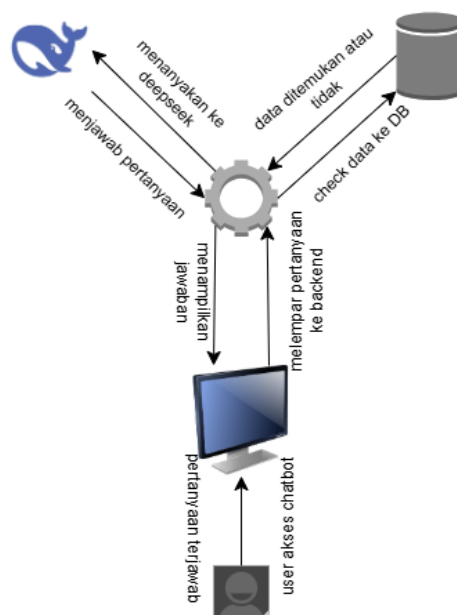
Gambar 3. Diagram Sequence

3. Diagram Aktifiti



Gambar 4. Diagram Aktifiti

4. Diagram Arsitektur



Gambar 5. Diagram Arsitektur

3. HASIL DAN PEMBAHASAN

Implementasi sistem chatbot ini dilakukan pada sisi *backend*. Semua proses *request* pada DeepSeek AI dan juga *search* data pada *qdrant* dilakukan didalamnya. Proses ini berlangsung ketika pengguna mengetikkan atau mengirimkan pertanyaan melalui halaman *customer service* (chatbot).

3.1. Implementasi Qdrant Vector Database dan DeepSeek AI

Proses penambahan data pada Qdrant dilakukan pada saat produk ditambahkan oleh admin, data yang dikonversi menjadi *vector* berasal dari data nama produk dan juga deskripsi produk.

Kode Program Connection Qdrant

```
const {QdrantClient} = require('@qdrant/js-client-rest');

const qdrant = new QdrantClient({
  url: process.env.QDRANT_URL,
  apiKey: process.env.QDRANT_API_KEY
})

Module.export = qdrant;
```

Ketika data berhasil di konversi kedalam bentuk *vector* data yang lainnya akan disimpan sebagai *payload* yang dijadikan sebagai acuan data. Data *payload* ini berisi, nama produk, deksripsi, harga, stok, dan info yang mungkin akan ditanyakan oleh pengguna.

Ada 2 variabel yang digunakan sebagai acuan data yang akan dicari berdasarkan pertanyaan user. Qdrant melakukan pencarian berbasis *similarity* yang hanya kemungkinan kebenarannya berdasarkan kesamaan *vector*, maka dari itu diperlukanya filtering ini agar dapat meningkatkan kemungkinan memberikan jawaban dengan tingkat akurasi tinggi. Kemudian pertanyaan user difilter untuk melihat kategori dan gender apa yang sedang dicari. Ini bermanfaat agar memberikan ketegasan kepada *qdrant* untuk data berdasarkan filter tersebut.

Kode Program Search Qdrant

```
const mustFilter = [];
if (gender) mustFilter.push({ key: "gender", match: { value: gender } });
if (kategori) mustFilter.push({ key: "kategori", match: { value: kategori } });

const searchResult = await qdrant.search('products', {
  vector: queryVector,
  limit: 10,
  payload: true,
  with_vector: false,
  filter: mustFilter.length ? { must: mustFilter } : undefined
});

let contextText = 'Tidak ada produk relevan ditemukan.';
if (searchResult.length) {
  const context = searchResult.map((hit, i) => {
    const p = hit.payload;
    return `${i + 1}. ${p.name} - Rp${p.price} (stok ${p.stock}) - ${p.description}`;
  }).join('\n');

  contextText = `Berikut daftar produk relevan:\n${context}`;
}

const messageWithContext = [
  {
    role: 'system',
    content: `Kamu adalah asisten e-commerce yang menjawab hanya berdasarkan informasi produk berikut:\n${contextText}\n` +
      `Jawab dengan sopan jika produk tidak tersedia.\n` +
      `Pengiriman hanya JNE dan J&T. Pembayaran via BCA, BRI, BSI. Lacak pesanan di menu Pesanan Saya`
  },
  message
];
```

Pada variabel *searchResult* akan dilakukan pencarian data produk berdasarkan kesamaan *vector* antara pertanyaan user dengan *vector* produk yang ada di database dengan menambahkan filter agar hasil semakin sesuai dengan pertanyaan yang di ajukan. Variabel *messageWithContext* merupakan pesan yang akan dilempar pada DeepSeek untuk di proses. Pesan yang dikirim memuat pertanyaan user dengan data yang sudah di dapatkan dari proses *search vector* pada *qdrant*.

Kode Program Request pada DeepSeek AI

```
const response = await fetch('https://openrouter.ai/api/v1/chat/completions', {
  method: 'POST',
```

```
headers: {
  'Content-Type': 'application/json',
  'Authorization': `Bearer ${process.env.DEEPSEEK_API_KEY}`
},
body: JSON.stringify({
  model: 'deepseek/deepseek-r1:free',
  messages: messagesWithContext
})
});

if (!response.ok) {
  const errText = await response.text();
  console.error('DeepSeek API error:', response.status, errText);
  return res.status(500).json({ error: 'DeepSeek API gagal.' });
}

const data = await response.json();
const reply = data.choices[0].message;

const chatDoc = new ChatHistory({
  userId,
  messages: [...message, reply],
  timestamp: new Date()
});
await chatDoc.save();

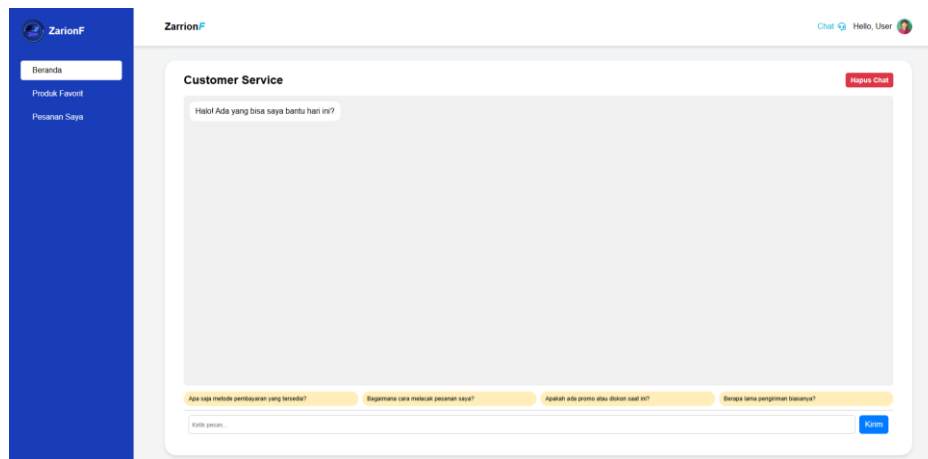
res.json({ reply });
```

Pada kode inilah proses utama dalam sistem chatbot ini, *backend* akan *request* dengan message yang sudah termasuk *systemPrompt* dan juga input pada halaman chatbot dari user. *Variable chatDoc* akan menyimpan *history chat* dari setiap user dengan *foreign key* untuk dasar mengambil data *history chat* berdasarkan *userId* dari user.

Ketika tidak adanya data yang dihasilkan dari proses *serach* data *qdrant vector*, maka pertanyaan akan langsung dikirimkan pada DeepSeek AI untuk diproses tanpa adanya data local *e-commerce*. Dalam scenario ini, DeepSeek AI bertindak sebagai *fallback* yang memberikan jawaban diluar cakupan data pada *qdrant*.

3.2. Hasil Pengujian Chatbot

Pengujian ini menggunakan metode *Black Box Testing* dimana sistem diuji langsung tanpa melihat kode program [10]. Beberapa tes yang sudah di lakukan pada sistem chatbot ini dengan harapan jawaban dari chatbot itu sendiri.



Gambar 5. Halaman Chatbot

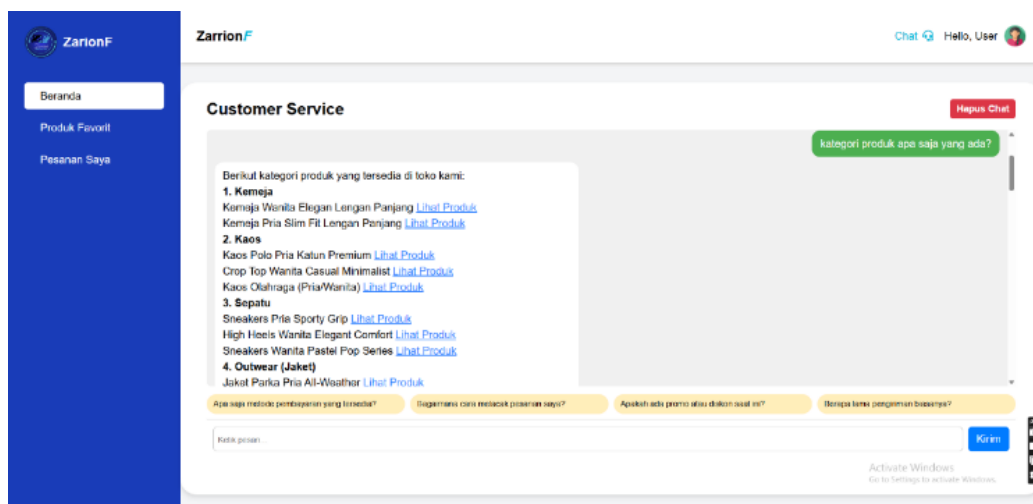
Pada halaman ini, user dapat melakukan input pertanyaan lalu klik tombol kirim, sistem akan otomatis melakukan *request* ke backend dan diteruskan ke DeepSeek AI, setelah backend mendapatkan jawaban maka backend akan mengirim jawaban ke frontend untuk ditampilkan ke user.

Pengujian sistem chatbot ini dilakukan dengan menerapkan beberapa skenario interaksi pengguna dengan sistem chatbot. Pengujian akan dijabarkan dalam tabel berikut:

Tabel 1. Pengujian Blackbox Testing Sistem Chatbot

No	Skenario	Hasil Yang Diharapkan	Hasil Pengujian	Simpulan	Waktu (detik)
1	“Saya bisa membayar dengan cara apa?”	Chatbot menjawab metode pembayaran transfer bank	Chatbot menjawab melalui transfer bank BRI, BCA, dan BSI.	Berhasil	11.32
2	“Berapa harga untuk sepatu sneakers?”	Chatbot menampilkan harga produk yang tersedia di database	Chatbot memberi info harga, dan detail produk	Berhasil	14.01
3	“Kategori apa saja yang tersedia?”	Chatbot menampilkan kategori yang ada pada took	Chatbot memberikan informasi mengenai kategori yang disediakan	Berhasil	10.20
4	“Berikan rekomendasi outfit untuk berolahraga?”	Chatbot merekomendasikan produk yang cocok dengan kondisi yang di minta	Chatbot memberikan informasi produk yang cocok	Berhasil	13.06
5	“Bagaimana cara membuat nasi goreng?”	Chatbot menjawab dengan mengarahkan Kembali arah obrolan keranah fashion	Chatbot mengarahkan kembali arah obrolan ke topik produk fashion	Berhasil	7.09

Berikut merupakan sample hasil tes dari skenario pengujian pada tabel di atas:



Gambar 6. Contoh Pengujian Sistem

3.3. Evaluasi

Tahap selanjutnya yaitu evaluasi hasil dari pengujian, tahapan ini diperlukan untuk dapat menentukan hasil apakah sistem berjalan sesuai dengan yang telah di tentukan sebelumnya atau belum. Jika sudah maka akan lanjut pada tahap development, dan jika masih terdapat ketidak sesuaian terhadap sistem maka akan dilakukan penyesuaian kembali untuk memastikan sistem berjalan sesuai tanpa adanya ketidaksesuaian.

4. KESIMPULAN

Implementasi DeepSeek AI dilakukan dengan mengintegrasikan *OpenRouter* model DeepSeek AI ke dalam aplikasi *e-commerce* berbasis web. Integrasi ini memanfaatkan arsitektur dari *frontend*, dimana permintaan dari pengguna akan dikirim melalui halaman chatbot ke *backend*. Backend akan memproses dan meneruskan pertanyaan ke DeepSeek AI dan menerima jawaban balikan lalu ditampilkan kembali ke pengguna sebagai jawaban akhir.

Waktu yang dibutuhkan untuk setiap pertanyaan rata rata kurang dari 10 – 15 detik (*free model*) dengan tingkat keberhasilan 100% dari 4 kasus pengujian.

Qdrant digunakan sebagai *database* untuk menyimpan *embedding (vector)* dari data yang dijadikan sebagai acuan, seperti nama produk dan deskripsi produk. Ketika pengguna mengajukan pertanyaan, sistem akan melakukan proses *semantic search* ke Qdrant untuk mencari *vector* yang paling mirip dengan pertanyaan perngguna.

Penambahan filter pada fungsi *search* akan menambah tingkat akurasi yang tinggi untuk memunculkan jawaban yang sesuai dengan pertanyaan. Dengan pendekatan ini, chatbot tidak akan menjawab pertanyaan diluar cakupan data produk yang ada.

Ketika pertanyaan pengguna tidak memiliki kecocokan *vector* atau tidak ditemukannya produk dalam Qdrant, sistem akan otomatis mengalihkan pertanyaan ke DeepSeek secara penuh tanpa adanya bantuan data yang ddi dapatkan. Skenario ini

memungkinkan DeepSeek menjadi *fallback reasoning engine* yang memberikan jawaban dari basis pengetahuan umum modelnya.

Berdasarkan hasil penelitian dalam pengembangan sistem yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. *Qdrant Vector Database* dan DeepSeek AI berhasil diintegrasikan pada aplikasi *e-commerce* berbasis web yang dapat diakses pada halaman pengguna melalui *icon* chat untuk membantu menjawab pertanyaan – pertanyaan umum terkait produk, ketersediaan stok, metode pembayaran dan pertanyaan lainnya.
2. Chatbot dapat merespons pertanyaan pengguna secara real-time menggunakan OpenRouter API KEY model DeepSeek AI, dengan bantuan seleksi data menggunakan QDrant Database Vector memungkinkan DeepSeek menjawab pertanyaan berdasarkan data produk yang ada di database, sehingga jawaban yang diberikan relevan dan kontekstual.
3. Sistem yang berjalan 24/7 memungkinkan pengguna untuk tetap mendapatkan pengalaman dalam mengakses layanan pengguna pada aplikasi *e-commerce*.
4. Penggunaan chatbot ini terbukti membantu efisiensi dalam memberikan pelayanan pelanggan (*customer service*) secara cepat tanpa ketergantungan sumber daya manusia, sehingga ketika diluar jam kerja toko sistem masih dapat diakses oleh pengguna.

DAFTAR PUSTAKA

- [1] N. K. Putri, "We Are Social: 59 persen pengguna internet Indonesia hobi belanja online," TECHINASIA. Accessed: Jun. 24, 2025. [Online]. Available: <https://id.techinasia.com/indonesia-hobi-belanja-online>
- [2] E. Nababan, Filza Izzati, Rati Rahmadani, and Lusiana Efrizoni, "Sistem Informasi Promosi dan Penjualan Pupuk, Beras dan Benih Berbasis Web," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 3, pp. 487–494, 2022, doi: 10.37859/coscitech.v3i3.4433.
- [3] Y. L. . Rehatlanit, "Peran E-Commerce Dalam Pengembangan Bisnis," *J. Teknol. Ind.*, vol. 5, no. 0, pp. 62–69, 2021, [Online]. Available: <https://journal.universitassuryadarma.ac.id/index.php/jti/article/view/764>
- [4] B. Web, S. Kasus, T. Seni, M. Abdul, and M. Akbar, "PROTOTIPE CHATBOT PADA APLIKASI E-COMMERCE," vol. 6, no. 2, pp. 240–250, 2024.
- [5] Y. N. Sari, "Jurnal Serambi Ekonomi dan Bisnis Penggunaan Chatbots dalam Meningkatkan Pengalaman Pelanggan pada Situs E-commerce," *J. Serambi Ekon. dan Bisnis*, vol. 7, no. 1, pp. 327–333, 2024, [Online]. Available: <https://ojs.serambimekkah.ac.id/serambi-ekonomi-dan-bisnis/>
- [6] A. Fitriani, D. Sulaeman, R. Firmansyah, F. Ekonomi, and S. Stambi, "Efektivitas Chatbot Sebagai Media Komunikasi Bisnis Antara Penjual dan Pembeli Pada Marketplace," *J. Disrupsi Bisnis*, vol. 5, no. 3, pp. 189–196, 2022, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/DRB/index><http://openjournal.unpam.ac.id/index.php/DRB/index>
- [7] N. Rachmat and D. P. Kesuma, "Implementasi Large Language Models Gemini Pada Pengembangan Aplikasi Chatbot Berbasis Android," *J. Ilmu Komput.*, vol. 4, no. 1, p. 2024, 2024, doi: 10.31314/juik.v4i1.2831.
- [8] D. A. Ryano, "Chatbot Pelayanan E-Commerce Dengan Naive Bayes Classifier," *Univ. UIN Suska Riau*, 2022, [Online]. Available: <https://repository.uin-suska.ac.id/59921/>
- [9] R. Hayami, S. Mohnica, and Soni, "Klasifikasi multilabel komentar toxic pada sosial media twitter menggunakan convolutional neural network(CNN)," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 4, no. 1, pp. 1–6, 2023, doi: 10.37859/coscitech.v4i1.4365.
- [10] S. Kasus, P. Anomali, and D. Sensus, "Pengembangan Aplikasi Chatbot dengan Large Language Model untuk Text-to-SQL Generation," pp. 831–840.