

Pengembangan Chatbot Berbasis Swarm untuk Kebutuhan Customer Service = Swarm Based Chatbot Development for Customer Service Needs

Muhammad Rifqi Praditya, author

Deskripsi Lengkap: <https://lib.ui.ac.id/detail?id=9999920552770&lokasi=lokal>

Abstrak

Agen customer service tidak dapat beroperasi 24 jam sehari, sehingga diperlukan sebuah program untuk mengatasi masalah ini, salah satunya adalah chatbot. Sejak munculnya makalah "Attention is All You Need", kualitas chatbot telah meningkat secara signifikan. Salah satu kegunaan utama chatbot adalah untuk menyediakan customer service 24 jam, sehingga memungkinkan agen customer service untuk fokus pada pertanyaan yang lebih kompleks. Untuk mengembangkan chatbot yang efektif yang dapat menjawab pertanyaan pelanggan, diperlukan data percakapan antara pelanggan dan agen, serta model pembelajaran mesin yang dilatih pada data ini. Dalam penelitian ini, eksperimen dilakukan menggunakan empat model yang berbeda: T5, BART, DistilGPT2, dan GPT2, untuk menentukan model terbaik yang dapat memberikan jawaban yang relevan kepada pelanggan. Model-model ini dievaluasi menggunakan metrik BLEU dan ROUGE untuk menentukan kualitas terbaik. Hasil eksperimen menunjukkan bahwa GPT2 adalah model terbaik untuk keperluan customer service. Untuk membuat model ini dapat diakses oleh pelanggan, diperlukan sebuah aplikasi web. Aplikasi web ini dikembangkan dengan menguji tiga frontend framework: Next.js, Vue.js, dan Angular, serta tiga backend framework: FastAPI, Flask, dan Django, untuk menentukan kombinasi yang memberikan respons tercepat. Frontend framework dievaluasi menggunakan Google Lighthouse. Backend framework dievaluasi menggunakan metrik Average Response Time. Hasil eksperimen menunjukkan bahwa Next.js adalah frontend framework terbaik dan FastAPI adalah backend framework terbaik. Untuk menangani sejumlah besar pelanggan secara bersamaan, chatbot ini memerlukan infrastruktur yang memungkinkan penggunaan paralel. Infrastruktur disediakan menggunakan Terraform untuk mengurangi langkah-langkah pengembangan dan memfasilitasi replikasi infrastruktur yang dikembangkan. Infrastruktur yang diuji adalah mikroservis dan monolitik, keduanya dapat diskalakan. Infrastruktur ini dievaluasi menggunakan metrik waktu respons dan tingkat kegagalan. Hasil eksperimen menunjukkan bahwa arsitektur monolitik cukup untuk menangani chatbot.

.....Customer service agents cannot operate 24 hours a day, thus a program is needed to address this issue, one of which is a chatbot. Since the emergence of the paper "Attention is All You Need", the quality of chatbots has improved significantly. One of the main uses of chatbots is to provide 24-hour customer service, thus allowing customer service agents to focus on more complex inquiries. To develop an effective chatbot that can answer customer questions, conversation data between customers and agents, as well as a machine learning model trained on this data, are required. In this research, experiments were conducted using four different models: T5, BART, DistilGPT2, and GPT2, to determine the best model that can provide relevant answers to customers. These models were evaluated using BLEU and ROUGE metrics to determine the best quality. The experimental results showed that GPT2 is the best model for customer service purposes. To make this model accessible to customers, a website is required. This website was developed by testing three frontend frameworks: Next.js, Vue.js, and Angular, and three backend frameworks: FastAPI, Flask, and Django, to determine the combination that provides the fastest response.

The frontend frameworks were evaluated using Google Lighthouse. The backend frameworks were evaluated using the Average Response Time metric. The experimental results showed that Next.js is the best frontend framework and FastAPI is the best backend framework. To handle a large number of customers simultaneously, this chatbot requires infrastructure that allows parallel usage. Infrastructure is provided using Terraform to reduce development steps and facilitate the replication of developed infrastructure. The tested infrastructures are microservices and monolithic, both of which are scalable. This infrastructure is evaluated using response time and failure rate metrics. The experimental results indicate that the monolithic architecture is sufficient for handling a chatbot.