

# Analisis Arsitektur Sistem Terdistribusi pada Chatbot Customer Service E-Commerce Menggunakan Load Balancing dan Message Queue: Studi Kasus Shopee

Osama Habib Candranata<sup>1\*</sup>, Lintang Putra Ramadhan<sup>2</sup>, Djenar Priya Asoka<sup>3</sup>, Aminuddin Fadli<sup>4</sup>

<sup>1</sup>Teknik Informatika  
Universitas Duta Bangsa  
<sup>1\*</sup>240103199@mhs.udb.ac.id

<sup>2</sup>Teknik Informatika  
Universitas Duta Bangsa  
<sup>2</sup>240103220@mhs.udb.ac.id

<sup>3</sup>Teknik Infomatika  
Universitas Duta Bangsa  
<sup>3</sup>240103188@mhs.udb.ac.id

<sup>4</sup>Teknik Informatika  
Universitas Duta bangsa  
<sup>4</sup>240103183@mhs.udb.ac.id

*Abstrak— Perkembangan e-commerce menyebabkan meningkatnya interaksi antara pengguna dan platform, termasuk dalam layanan customer service. Chatbot menjadi salah satu teknologi yang digunakan untuk memberikan respons otomatis kepada pengguna. Namun, pada platform e-commerce berskala besar seperti Shopee, jumlah permintaan pengguna dapat meningkat secara signifikan, terutama saat promo besar, flash sale, atau festival belanja. Kondisi tersebut dapat menimbulkan masalah pada arsitektur chatbot yang masih terpusat, seperti bottleneck, keterlambatan respons, single point of failure, dan keterbatasan skalabilitas. Penelitian ini bertujuan untuk menganalisis arsitektur sistem terdistribusi pada chatbot customer service e-commerce dengan menggunakan load balancing dan message queue. Metode penelitian yang digunakan adalah studi literatur dan analisis konseptual terhadap arsitektur awal, identifikasi masalah, perancangan arsitektur usulan, serta perbandingan arsitektur. Hasil analisis menunjukkan bahwa load balancing dapat membantu mendistribusikan permintaan pengguna ke beberapa chatbot server, sedangkan message queue dapat mengelola antrian pesan secara asinkron sehingga beban terhadap backend service dan database lebih terkendali. Arsitektur usulan dinilai mampu meningkatkan scalability, response time, reliability, availability, fault tolerance, dan maintainability. Dengan demikian, kombinasi load balancing dan message queue dapat menjadi pendekatan konseptual yang relevan untuk meningkatkan kualitas layanan chatbot customer service pada platform e-commerce berskala besar.*

*Kata kunci— chatbot, e-commerce, sistem terdistribusi, load balancing, message queue.*

*Abstract— The growth of e-commerce has increased user interaction with digital platforms, including customer service activities. Chatbots are widely used to provide automated responses to users. However, in large-scale e-commerce platforms such as Shopee, user requests may increase significantly, especially during major promotions, flash sales, or shopping festivals. This condition may create problems in centralized chatbot architecture, such as bottlenecks, delayed response time, single point of failure, and limited scalability. This study aims to analyze a distributed system architecture for e-commerce customer service chatbots using load balancing and message queue mechanisms. The research method used is literature study and conceptual analysis, including initial architecture analysis, problem identification, proposed architecture design, and architecture comparison. The results show that load balancing can distribute user requests to multiple chatbot servers, while message queues can manage message processing asynchronously, reducing direct pressure on backend services and databases. The proposed architecture is considered capable of improving scalability, response time, reliability, availability, fault tolerance, and maintainability. Therefore, the combination of load balancing and message queue can be a relevant conceptual approach to improving the quality of customer service chatbot systems in large-scale e-commerce platforms.*

*Keywords— chatbot, e-commerce, distributed system, load balancing, message queue.*

## I. PENDAHULUAN

Perkembangan e-commerce membuat interaksi antara pengguna dan platform semakin meningkat. Pengguna tidak hanya melakukan transaksi, tetapi juga sering mengajukan pertanyaan, keluhan, atau permintaan bantuan melalui layanan customer service. Karena itu, platform e-commerce membutuhkan sistem layanan pelanggan yang cepat, stabil, dan dapat digunakan kapan saja[1].

Salah satu teknologi yang banyak digunakan untuk membantu layanan customer service adalah chatbot. Chatbot dapat menjawab pertanyaan pengguna secara otomatis, melayani banyak percakapan secara bersamaan, dan bekerja selama 24 jam. Dengan adanya chatbot, pengguna dapat memperoleh bantuan lebih cepat tanpa harus selalu menunggu agen manusia[2][3].

Namun, pada platform e-commerce berskala besar seperti Shopee, jumlah pengguna yang mengakses layanan dapat meningkat sangat tinggi, terutama saat promo besar atau festival belanja. Jika sistem chatbot hanya bergantung pada satu server atau arsitektur terpusat, maka sistem dapat mengalami keterlambatan respons, penumpukan pesan, bahkan kegagalan layanan ketika trafik meningkat.

Untuk mengatasi masalah tersebut, arsitektur sistem terdistribusi dapat digunakan. Sistem terdistribusi membagi proses kerja ke beberapa server yang saling terhubung sehingga beban tidak hanya ditangani oleh satu server saja. Dengan cara ini, sistem dapat bekerja lebih stabil dan tetap berjalan meskipun salah satu server mengalami gangguan[4][5].

Dalam arsitektur ini, load balancing digunakan untuk membagi permintaan pengguna ke beberapa server secara merata. Sementara itu, message queue digunakan untuk mengatur *antrean* pesan agar tidak langsung membebani sistem secara bersamaan. Kombinasi kedua mekanisme ini dapat membantu chatbot bekerja lebih cepat,

stabil, dan mampu menangani lonjakan permintaan[6][7].

Beberapa penelitian terdahulu telah membahas sistem terdistribusi secara umum, seperti prinsip skalabilitas dan fault tolerance pada arsitektur enterprise[4][5], maupun penerapan load balancing dan message queue secara terpisah pada konteks jaringan atau sistem enterprise lainnya[6][7]. Namun, penelitian-penelitian tersebut belum secara khusus membahas penerapan kedua mekanisme ini pada arsitektur chatbot customer service di platform e-commerce berskala besar, khususnya dalam menghadapi lonjakan trafik musiman seperti flash sale dan festival belanja. Kesenjangan inilah yang menjadi alasan penelitian ini dilakukan, yaitu untuk menganalisis secara konseptual bagaimana kombinasi load balancing dan message queue dapat diterapkan secara spesifik pada arsitektur chatbot customer service e-commerce.

Berdasarkan permasalahan tersebut, penelitian ini berfokus pada analisis penerapan arsitektur sistem terdistribusi pada chatbot customer service e-commerce dengan studi kasus konseptual pada Shopee. Permasalahan yang dibahas adalah bagaimana load balancing dan message queue dapat digunakan untuk meningkatkan kemampuan chatbot dalam menangani banyak permintaan pengguna sekaligus serta menjaga kestabilan layanan ketika terjadi lonjakan trafik. Oleh karena itu, penelitian ini bertujuan untuk menganalisis peran kedua mekanisme tersebut dalam meningkatkan skalabilitas, keandalan, dan responsivitas sistem chatbot.

Agar pembahasan lebih terarah, penelitian ini dibatasi pada analisis konseptual berdasarkan studi literatur tanpa melakukan implementasi sistem secara langsung. Penelitian ini juga tidak menggunakan data operasional maupun data internal Shopee dan tidak membahas aspek keamanan sistem secara mendalam. Shopee digunakan sebagai studi kasus konseptual karena memiliki karakteristik yang mewakili platform e-

commerce berskala besar dengan jumlah pengguna dan trafik yang tinggi[8].

## II. METODOLOGI PENELITIAN

### A. Studi Literatur

Tahap pertama dalam penelitian ini adalah studi literatur yang bertujuan untuk memahami konsep-konsep yang berkaitan dengan sistem terdistribusi, load balancing, message queue, chatbot, dan layanan e-commerce berskala besar. Referensi diperoleh dari jurnal, prosiding, buku, serta dokumentasi teknis resmi yang relevan dengan topik penelitian.

Melalui tahap ini diperoleh pemahaman mengenai cara kerja sistem terdistribusi, mekanisme pembagian beban, pengelolaan *antrean* pesan, serta pola arsitektur yang umum digunakan pada platform e-commerce. Hasil studi literatur kemudian digunakan sebagai dasar dalam melakukan analisis dan perancangan arsitektur yang diusulkan.

### B. Analisis Arsitektur Awal

Setelah melakukan studi literatur, tahap berikutnya adalah menganalisis arsitektur chatbot customer service yang umum digunakan pada platform e-commerce[9]. Analisis ini dilakukan berdasarkan referensi dan studi kasus yang tersedia secara publik, tanpa menggunakan data atau sistem internal Shopee. Pada arsitektur dasar, pesan dari pengguna akan langsung dikirim ke satu server chatbot untuk diproses, kemudian server mengakses database dan mengirimkan respons kembali kepada pengguna.

Arsitektur ini cukup sederhana dan mudah diterapkan, tetapi memiliki beberapa keterbatasan ketika digunakan pada skala besar. Seluruh beban pemrosesan berada pada satu server sehingga berisiko mengalami overload saat trafik meningkat. Selain itu, jika server mengalami gangguan, layanan chatbot dapat berhenti beroperasi. Oleh karena itu, analisis terhadap arsitektur awal diperlukan untuk mengidentifikasi kelemahan yang ada dan menjadi dasar dalam merancang arsitektur yang lebih andal dan skalabel.

### C. Identifikasi Masalah

Beberapa permasalahan yang sering muncul pada arsitektur chatbot customer service yang masih menggunakan server terpusat adalah sebagai berikut.

#### 1. Bottleneck pada Server Tunggal

Seluruh permintaan pengguna diproses oleh satu server sehingga berpotensi menimbulkan kemacetan ketika jumlah pengguna meningkat, terutama pada saat promo besar atau lonjakan trafik.

#### 2. Keterlambatan Respons

Penumpukan permintaan dapat menyebabkan waktu respons chatbot menjadi lebih lambat, sehingga memengaruhi kualitas layanan yang diterima pengguna.

#### 3. Single Point of Failure

Ketergantungan pada satu server membuat layanan rentan mengalami gangguan. Jika server utama gagal beroperasi, seluruh layanan chatbot dapat berhenti.

#### 4. Skalabilitas Terbatas

Penambahan kapasitas sistem umumnya hanya dilakukan dengan meningkatkan spesifikasi server yang ada, sehingga kurang fleksibel untuk menghadapi pertumbuhan jumlah pengguna.

#### 5. Ketersediaan Layanan Rendah

Ketersediaan layanan sangat bergantung pada kondisi satu server. Ketika terjadi gangguan, layanan tidak dapat berjalan secara optimal dan berpotensi mengurangi pengalaman pengguna.

Permasalahan-permasalahan tersebut menunjukkan bahwa arsitektur terpusat memiliki keterbatasan dalam menangani kebutuhan layanan chatbot pada platform e-commerce berskala besar. Oleh karena itu, diperlukan rancangan arsitektur yang lebih skalabel, andal, dan mampu menjaga ketersediaan layanan[10], sehingga dapat mendukung peningkatan jumlah pengguna dan lonjakan trafik secara lebih efektif.

#### 6. Perancangan Arsitektur Usulan

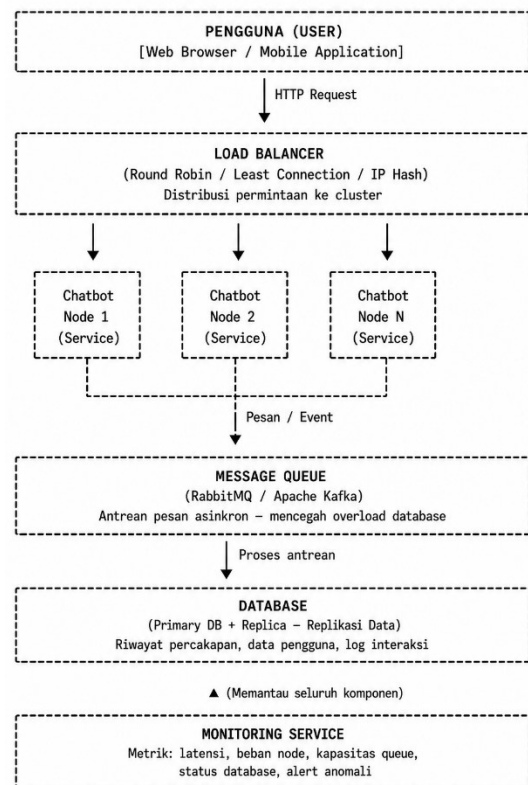
Berdasarkan permasalahan yang telah diidentifikasi, penelitian ini mengusulkan

arsitektur sistem terdistribusi yang terdiri dari beberapa komponen utama, yaitu User, Load Balancer, Chatbot Service Cluster, Message Queue, Database, dan Monitoring Service. Pengguna mengakses layanan chatbot melalui aplikasi, kemudian permintaan akan diterima oleh load balancer yang bertugas membagi beban secara merata ke beberapa server chatbot. Dengan adanya beberapa server yang bekerja secara bersamaan, sistem dapat menangani lebih banyak permintaan dan tetap beroperasi meskipun salah satu server mengalami gangguan.

Untuk mendukung proses layanan, message queue digunakan sebagai penghubung antara chatbot dan database sehingga pemrosesan data dapat dilakukan secara asinkron tanpa membebani sistem secara langsung. Database berfungsi menyimpan data percakapan dan informasi pendukung lainnya, sedangkan monitoring service digunakan untuk memantau kondisi seluruh komponen sistem secara real-time. Kombinasi komponen-komponen tersebut memungkinkan sistem chatbot bekerja lebih stabil, mudah dikembangkan, dan mampu menghadapi lonjakan trafik yang tinggi.

#### D. Diagram Arsitektur Penelitian

Untuk memberikan gambaran mengenai arsitektur sistem yang dianalisis dalam penelitian ini, disusun sebuah diagram arsitektur penelitian sebagaimana ditunjukkan pada Gambar 1. Diagram tersebut memperlihatkan hubungan antar komponen utama dalam arsitektur chatbot customer service berbasis sistem terdistribusi, mulai dari pengguna, load balancer, cluster chatbot, message queue, database, hingga monitoring service. Penyajian diagram ini bertujuan untuk mempermudah pemahaman mengenai alur pemrosesan permintaan pengguna dan peran masing-masing komponen dalam meningkatkan kinerja sistem.



(gambar 1. Diagram Arsitektur Penelitian)

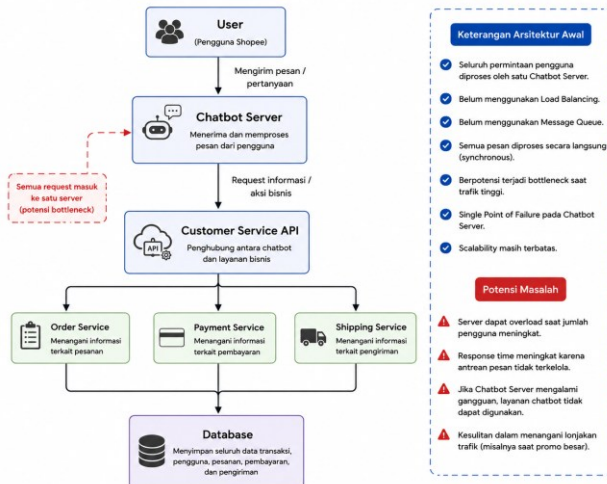
Berdasarkan Gambar 1, pengguna mengirimkan permintaan ke load balancer yang mendistribusikannya ke beberapa chatbot node. Selanjutnya, pesan diproses melalui message queue sebelum disimpan pada database. Seluruh komponen dipantau oleh monitoring service untuk menjaga kinerja dan ketersediaan sistem.

### III. HASIL DAN PEMBAHASAN

#### A. Arsitektur Awal Chatbot Customer Service

Arsitektur awal chatbot customer service pada platform e-commerce dalam penelitian ini digambarkan sebagai model arsitektur terpusat. Pada model ini, pengguna mengirimkan pesan atau pertanyaan melalui aplikasi e-commerce, kemudian pesan tersebut diteruskan ke satu chatbot server. Chatbot server berfungsi menerima, memproses, dan memahami pesan pengguna sebelum meneruskan permintaan ke customer service API. Selanjutnya, customer service API menghubungkan chatbot dengan layanan

bisnis, seperti order service, payment service, dan shipping service. Data yang dibutuhkan kemudian diambil dari database untuk menghasilkan respons kepada pengguna.



(gambar 2. Arsitektur Awal Chatbot Customer Service Berbasis Server Tunggal)

Berdasarkan Gambar 2, terlihat bahwa seluruh permintaan pengguna masih diproses melalui satu chatbot server. Arsitektur seperti ini memiliki alur yang sederhana dan mudah diterapkan, terutama pada sistem dengan jumlah pengguna yang belum terlalu besar. Namun, ketika digunakan pada platform e-commerce berskala besar, model ini memiliki beberapa kelemahan. Salah satu kelemahan utamanya adalah seluruh beban permintaan terpusat pada satu server, sehingga server tersebut berpotensi mengalami bottleneck ketika jumlah pesan pengguna meningkat.

Dalam konteks e-commerce seperti Shopee, peningkatan jumlah pengguna dapat terjadi pada momen tertentu, misalnya saat flash sale, festival belanja bulanan, promo besar, atau event khusus. Pada kondisi tersebut, pengguna tidak hanya melakukan transaksi, tetapi juga mengajukan pertanyaan tentang pesanan, pembayaran, pengiriman, voucher, pengembalian barang, dan keluhan transaksi. Apabila seluruh permintaan tersebut masuk ke satu server chatbot, maka sistem dapat mengalami peningkatan response time, antrian

pesan yang tidak terkendali, bahkan gangguan layanan.

Selain bottleneck, arsitektur awal juga memiliki risiko single point of failure. Artinya, apabila chatbot server mengalami gangguan, maka seluruh layanan chatbot dapat ikut berhenti karena tidak ada server lain yang dapat mengambil alih proses layanan. Kondisi ini dapat berdampak pada reliability dan availability sistem. Padahal, layanan customer service pada e-commerce harus mampu berjalan secara stabil dan tersedia secara terus-menerus agar pengguna tetap memperoleh bantuan secara cepat.

### B. Analisis Masalah pada Arsitektur Awal

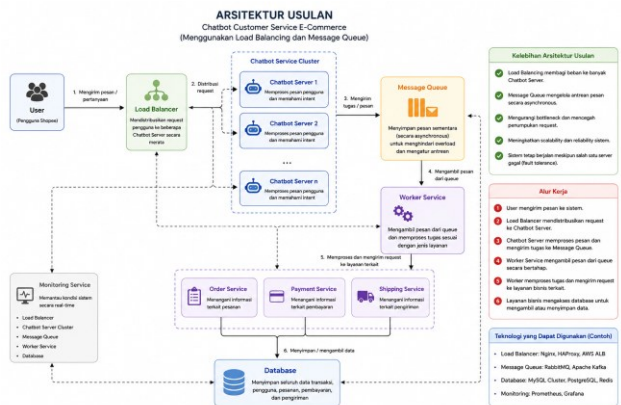
Berdasarkan analisis terhadap arsitektur awal, terdapat beberapa permasalahan yang dapat memengaruhi kinerja chatbot customer service. Masalah utama adalah bottleneck pada chatbot server karena seluruh permintaan pengguna diproses oleh satu server yang sama. Ketika jumlah permintaan meningkat, server harus menangani banyak pesan secara bersamaan sehingga dapat menyebabkan penumpukan proses dan keterlambatan respons. Kondisi ini juga berdampak pada meningkatnya response time, yaitu waktu yang dibutuhkan sistem untuk memberikan jawaban kepada pengguna. Dalam layanan customer service, respons yang lambat dapat menurunkan kualitas pengalaman pengguna, terutama saat terjadi kendala transaksi yang memerlukan penanganan cepat.

Selain itu, arsitektur awal memiliki keterbatasan scalability karena peningkatan kapasitas hanya mengandalkan vertical scaling pada satu server, yang kurang fleksibel dan tidak selalu efisien dari sisi biaya. Arsitektur ini juga memiliki fault tolerance yang rendah karena kegagalan pada chatbot server dapat menyebabkan layanan tidak dapat digunakan. Tidak adanya mekanisme message queue membuat seluruh pesan diproses secara sinkron, sehingga ketika backend service atau database mengalami beban tinggi, proses respons chatbot ikut terhambat dan kinerja

sistem secara keseluruhan menjadi kurang optimal[11].

C. Arsitektur Usulan dengan Load Balancing dan Message Queue

Untuk mengatasi kelemahan arsitektur awal, penelitian ini mengusulkan arsitektur sistem terdistribusi yang memanfaatkan load balancing dan message queue. Arsitektur ini terdiri atas beberapa komponen utama, yaitu user, load balancer, chatbot service cluster, message queue, worker service, layanan bisnis, database, dan monitoring service. Komponen-komponen tersebut saling terhubung untuk membentuk sistem chatbot customer service yang lebih skalabel, responsif, dan andal.



(gambar 3. Arsitektur Usulan Chatbot Customer Service E-Commerce dengan Load Balancing dan Message Queue)

Berdasarkan Gambar 3, proses dimulai ketika pengguna mengirimkan pesan melalui aplikasi e-commerce. Permintaan pengguna diterima oleh load balancer dan didistribusikan ke beberapa chatbot server yang tergabung dalam chatbot service cluster. Setiap chatbot server bertugas menerima pesan, memahami intent pengguna, dan menentukan jenis layanan yang diperlukan. Mekanisme ini memungkinkan pembagian beban kerja secara merata serta meningkatkan ketersediaan layanan karena permintaan dapat dialihkan ke server lain apabila terjadi gangguan pada salah satu node[12].

Permintaan yang memerlukan proses lanjutan kemudian dikirim ke message queue sebagai antrean pesan yang diproses secara asinkron oleh worker service. Selanjutnya,

worker service meneruskan permintaan ke layanan bisnis yang sesuai, seperti order service, payment service, atau shipping service, yang kemudian berinteraksi dengan database untuk mengakses maupun menyimpan data. Selain itu, monitoring service digunakan untuk memantau kondisi seluruh komponen sistem secara real-time sehingga potensi gangguan dapat dideteksi lebih awal dan kualitas layanan tetap terjaga[7][13].

D. Perbandingan Arsitektur Awal dan Arsitektur Usulan

Perbandingan antara arsitektur awal dan arsitektur usulan dilakukan berdasarkan beberapa indikator teknis, yaitu scalability, response time, reliability, availability, fault tolerance, dan maintainability. Indikator tersebut dipilih karena berkaitan langsung dengan kebutuhan sistem chatbot customer service pada e-commerce berskala besar.

| Indikator       | Arsitektur Awal                                                                    | Arsitektur Usulan                                                                                           |
|-----------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| Scalability     | Terbatas karena bergantung pada satu chatbot server.                               | Lebih fleksibel karena dapat menambah beberapa chatbot server dalam cluster.                                |
| Response time   | Berpotensi meningkat ketika trafik tinggi karena semua pesan masuk ke satu server. | Lebih terkendali karena permintaan dibagi oleh load balancer dan pesan dapat diantri melalui message queue. |
| Reliability     | Rendah karena kegagalan pada server utama dapat mengganggu seluruh layanan.        | Lebih baik karena beberapa node chatbot dapat saling menggantikan.                                          |
| Availability    | Rentan mengalami downtime apabila chatbot server gagal.                            | Lebih tinggi karena load balancer dapat mengalihkan trafik ke node lain yang aktif.                         |
| Fault tolerance | Memiliki single point of failure pada chatbot server.                              | Lebih toleran terhadap kegagalan karena sistem terdiri atas beberapa komponen dan node layanan.             |

|                           |                                                                                            |                                                                                                      |
|---------------------------|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| Maintainability           | Pemeliharaan server dapat dilakukan secara bertahap pada masing-masing node atau komponen. | Pemeliharaan dapat dilakukan secara bertahap pada masing-masing node atau komponen.                  |
| Pengelolaan antrean pesan | Belum memiliki mekanisme antrean pesan yang baik.                                          | Message queue mengatur pesan secara asinkron dan bertahap.                                           |
| Beban database            | Database menerima langsung dari permintaan.                                                | Beban database lebih terkendali karena permintaan diproses melalui worker service dan message queue. |

(Tabel I. Perbandingan Arsitektur Awal dan Arsitektur Usulan)

Berdasarkan Tabel I, arsitektur usulan memiliki keunggulan dibandingkan arsitektur awal. Pada aspek scalability, arsitektur usulan memungkinkan penambahan chatbot server sesuai kebutuhan trafik. Hal ini berbeda dengan arsitektur awal yang hanya bergantung pada satu server. Pada aspek response time, arsitektur usulan lebih baik karena permintaan pengguna tidak menumpuk pada satu server, melainkan didistribusikan melalui load balancer dan dikelola melalui message queue.

Pada aspek reliability dan availability, arsitektur usulan lebih unggul karena tidak bergantung pada satu chatbot server. Jika salah satu node mengalami gangguan, node lain masih dapat melayani permintaan pengguna. Hal ini membuat sistem tetap berjalan meskipun terdapat komponen yang bermasalah. Pada aspek fault tolerance, penggunaan chatbot service cluster dan message queue dapat mengurangi risiko kegagalan total layanan.

#### E. Manfaat Arsitektur Usulan

Penerapan load balancing dan message queue pada arsitektur chatbot customer service memberikan beberapa manfaat. Pertama, sistem menjadi lebih skalabel karena kapasitas layanan dapat ditingkatkan dengan menambahkan chatbot server baru ke dalam cluster. Kedua, sistem menjadi lebih responsif karena permintaan pengguna dapat dibagi ke beberapa server sehingga mengurangi

penumpukan pada satu titik. Ketiga, message queue membantu mengatur antrean pesan agar proses yang membutuhkan waktu lebih lama tidak langsung membebani backend service dan database.

Keempat, arsitektur usulan meningkatkan reliability dan availability sistem. Dengan adanya beberapa chatbot server, layanan tidak langsung berhenti ketika salah satu server mengalami gangguan. Kelima, arsitektur ini mendukung fault tolerance karena kegagalan pada salah satu komponen dapat diminimalkan dampaknya melalui pengalihan trafik dan pemrosesan antrean pesan. Keenam, sistem menjadi lebih mudah dipelihara karena komponen layanan dapat dikelola secara terpisah.

Secara keseluruhan, arsitektur usulan dapat menjadi pendekatan konseptual yang relevan untuk meningkatkan kualitas layanan chatbot customer service pada platform e-commerce berskala besar. Arsitektur ini tidak hanya mengurangi risiko bottleneck dan single point of failure, tetapi juga membantu sistem dalam menghadapi lonjakan trafik, khususnya pada momen promosi besar, flash sale, atau event belanja yang meningkatkan jumlah interaksi pengguna.

#### F. Tantangan dan Keterbatasan Arsitektur Usulan

Meskipun arsitektur usulan menawarkan berbagai kelebihan seperti peningkatan skalabilitas, ketersediaan, dan keandalan sebagaimana telah dijelaskan sebelumnya, penerapannya juga memiliki sejumlah kelemahan dan tantangan yang perlu diperhatikan. Dari sisi kelemahan, penambahan komponen seperti load balancer, message queue, dan worker service membuat arsitektur menjadi lebih kompleks dibandingkan model server tunggal, sehingga membutuhkan pemahaman teknis yang lebih matang dalam perancangan, konfigurasi, dan pemeliharannya. Dari sisi tantangan implementasi, organisasi perlu memastikan konsistensi data antar node chatbot, menentukan strategi load balancing yang tepat,

serta mengelola integrasi antara message queue dengan layanan bisnis yang sudah ada.

Selain itu, penambahan komponen baru pada arsitektur usulan juga berpotensi memunculkan risiko bottleneck baru, misalnya pada load balancer maupun message queue itu sendiri apabila kapasitasnya tidak disesuaikan dengan volume trafik yang terus meningkat. Oleh karena itu, kedua komponen tersebut tetap perlu dirancang agar scalable dan tidak menjadi titik kegagalan yang baru. Dari sisi biaya, penerapan arsitektur terdistribusi umumnya membutuhkan investasi tambahan, baik untuk infrastruktur server, lisensi maupun operasional message queue, hingga sumber daya manusia yang mengelola sistem, sehingga perlu dipertimbangkan dari sisi efisiensi biaya terhadap manfaat yang diperoleh. Penelitian ini juga masih bersifat konseptual berdasarkan studi literatur sehingga belum dilengkapi dengan validasi empiris, seperti pengujian performa atau simulasi beban, yang diperlukan untuk mengukur secara kuantitatif manfaat arsitektur usulan pada kondisi nyata.

#### IV. KESIMPULAN

Berdasarkan hasil analisis, arsitektur chatbot customer service yang masih menggunakan server tunggal memiliki beberapa kelemahan, seperti risiko bottleneck, keterlambatan respons saat trafik tinggi, keterbatasan skalabilitas, dan potensi single point of failure. Untuk mengatasi masalah tersebut, penelitian ini mengusulkan arsitektur sistem terdistribusi yang memanfaatkan load balancing dan message queue. Load balancing membantu mendistribusikan permintaan pengguna ke beberapa chatbot server, sedangkan message queue mengelola antrean pesan secara asinkron sehingga proses dapat berjalan lebih teratur dan tidak membebani sistem secara langsung.

Hasil pembahasan menunjukkan bahwa kombinasi load balancing dan message queue mampu meningkatkan skalabilitas, keandalan, ketersediaan, dan responsivitas layanan

chatbot secara konseptual. Dengan demikian, tujuan penelitian untuk menganalisis penerapan kedua mekanisme tersebut pada chatbot customer service e-commerce telah tercapai. Namun, penelitian ini masih terbatas pada studi literatur dan analisis konseptual, sehingga penelitian selanjutnya dapat difokuskan pada implementasi dan pengujian performa untuk memperoleh hasil yang lebih terukur[4][5] [6][7].

Kontribusi ilmiah dari penelitian ini terletak pada penyusunan kerangka konseptual arsitektur sistem terdistribusi yang secara spesifik menggabungkan load balancing dan message queue untuk konteks chatbot customer service pada platform e-commerce berskala besar, yang belum banyak dibahas secara khusus pada penelitian-penelitian sebelumnya. Kerangka ini dapat menjadi acuan awal bagi peneliti maupun praktisi dalam merancang arsitektur chatbot yang lebih skalabel dan andal. Penelitian ini memiliki keterbatasan karena hanya menggunakan pendekatan studi literatur dan analisis konseptual, tanpa disertai implementasi sistem, pengujian performa, maupun simulasi beban trafik pada kondisi nyata. Dengan demikian, manfaat arsitektur usulan yang disampaikan pada penelitian ini masih bersifat teoretis dan perlu diuji lebih lanjut secara kuantitatif, misalnya melalui pengukuran response time, throughput, dan tingkat ketersediaan sistem pada skenario trafik yang mendekati kondisi operasional sebenarnya.

#### UCAPAN TERIMA KASIH

Bagian ini memberikan apresiasi kepada perorangan maupun organisasi yang memberikan bantuan kepada penulis dan atau kepada pihak sponsor apabila ada.

#### REFERENSI

- [1] D. S. Manap Solihat and D. Sandika, "E-commerce di Industri 4.0," *J. Ilm. Bisnis dan Ekon. Asia*, vol. 16, no. 2, pp. 273–281, 2022, doi:

- 10.32812/jibeka.v16i2.967.
- [2] B. Luo, R. Y. K. Lau, C. Li, and Y. W. Si, "A critical review of state-of-the-art chatbot designs and applications," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 12, no. 1, pp. 1–26, 2022, doi: 10.1002/widm.1434.
- [3] C. (Mitsu) Feng, E. Botha, and L. Pitt, "From HAL to GenAI: Optimizing chatbot impacts with CARE," *Bus. Horiz.*, vol. 67, no. 5, pp. 537–548, 2024, doi: 10.1016/j.bushor.2024.04.012.
- [4] M. Ilham, I. F. Hanif, and F. Adiansyah, "Arsitektur Sistem Terdistribusi Untuk Enterprise: Model Desain Dan Implementasi," *J. Rekayasa Sist. Inf. dan Teknol.*, vol. 3, no. 1, pp. 60–67, 2025, doi: 10.70248/jrsit.v3i1.2503.
- [5] I. Online, "Sarcouncil Journal of Engineering and Computer Sciences Demystifying Distributed Systems : Scalability , Modularity , and Fault Tolerance Explained," vol. 0, pp. 949–959, 2025.
- [6] E. Gures, I. Shaye, M. Ergen, M. H. Azmi, and A. A. El-Saleh, "Machine Learning-Based Load Balancing Algorithms in Future Heterogeneous Networks: A Survey," *IEEE Access*, vol. 10, pp. 37689–37717, 2022, doi: 10.1109/ACCESS.2022.3161511.
- [7] R. Maharjan, M. S. H. Chy, M. A. Arju, and T. Cerny, "Benchmarking Message Queues," *Telecom*, vol. 4, no. 2, pp. 298–312, 2023, doi: 10.3390/telecom4020018.
- [8] E. M. Asih, "Analisis pada Shopee sebagai E-Commerce Terpopuler di Indonesia," *J. Ekon. Bisnis Antart.*, vol. 2, no. 1, pp. 73–79, 2024, doi: 10.70052/jeba.v2i1.299.
- [9] R. Andrico, Hetty Rohayani, T. M. Novia, and S. Amandha, "Rancang Bangun Sistem Terdistribusi pada Toko Roti Valentine Tanjung Jabung Barat," *Polyg. J. Ilmu Komput. dan Ilmu Pengetah. Alam*, vol. 2, no. 4, pp. 157–165, 2024, doi: 10.62383/polygon.v2i4.197.
- [10] V. Z. Erikasari, T. F. Maharani, and Elkin Rilvani, "Literature Review: Blockchain dan Edge Computing dalam Optimalisasi Sistem Terdistribusi Masa Depan," *Sisfotenika*, vol. 15, no. 1, pp. 43–54, 2025, doi: 10.30700/sisfotenika.v15i1.531.
- [11] F. M. Khaninezhad, S. Mouhebati, and P. Analysis, "An Analytical Model for Enhancing Scalability and Reliability in Computer Networks," 2025, doi: 10.63053/ijset.101.
- [12] B. Kruekaew and W. Kimpan, "Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm with Reinforcement Learning," *IEEE Access*, vol. 10, pp. 17803–17818, 2022, doi: 10.1109/ACCESS.2022.3149955.
- [13] M. S. H. Chy, M. A. R. Arju, S. M. Tella, and T. Cerny, "Comparative Evaluation of Java Virtual Machine-Based Message Queue Services: A Study on Kafka, Artemis, Pulsar, and RocketMQ," *Electron.*, vol. 12, no. 23, 2023, doi: 10.3390/electronics12234792.