

Implementasi *Cloud Computing* Pada Aplikasi Kuliner Berbasis Android Menggunakan Teknologi *Mobile Backend As A Service*

Muhammad Munawir¹, Indrawati^{2*}, Ilham Safar³

^{1,2,3} *Jurusan Teknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA*

¹munawir.ucb@gmail.com

^{2*}indrawati@pnl.ac.id (penulis korespondensi)

³ilham_safar@pnl.ac.id

Abstrak— Teknologi memiliki peluang besar pada industri kuliner, untuk mengadopsi teknologi digital guna meningkatkan efisiensi operasional dan daya saing. *Cloud computing* telah menjadi salah satu teknologi kunci yang mendorong transformasi digital, terutama melalui konsep *Mobile Backend as a Service (MBaaS)*. Penggunaan *MBaaS* memungkinkan aplikasi berfokus pada pengembangan fitur tanpa perlu mengelola server, basis data, otentikasi, dan layanan lainnya. Tujuan penelitian adalah untuk mengetahui performa aplikasi yang diintegrasikan dengan *cloud* dan mengetahui beban kinerja aplikasi melalui metode pengujian *Quality of Service*. Hasil penelitian menunjukkan bahwa *cloud* dengan teknologi *MBaaS* menunjukkan performa unggul dalam pengujian *Quality of Service* menggunakan *Wireshark*, *firestore* database mencapai *throughput* 10507 kbps, *packet loss* 0.0%, *delay* 0.0 ms, dan *jitter* -0,003 ms. Berdasarkan analisis performa menggunakan *tools monitoring* dari layanan Google Cloud Platform, data *application load balancer rule - request bytes*, berada stabil dengan rata-rata request bytes 500B/s dengan permintaan tertinggi mencapai 512.08B/s. Analisa *application load balancer rule - response bytes*, berada stabil dengan rata-rata respon bytes 500b/s dan respon tertinggi mencapai 559.97B/s. Berdasarkan hasil penelitian dapat disimpulkan bahwa, *cloud computing* dengan teknologi *MBaaS*, mampu menangani beban permintaan dengan performa yang cukup baik. Namun, terdapat batasan pada kemampuan penanganan beban yang sangat tinggi, seperti peningkatan *error* dan penurunan *throughput* pada pengujian *stress testing*.

Kata kunci— aplikasi, cloud, firebase, google cloud platform, MBaaS.

Abstract— *Technology presents a great opportunity for the culinary industry to adopt digital technology to improve operational efficiency and competitiveness. Cloud computing has become one of the key technologies driving digital transformation, especially through the Mobile Backend as a Service (MBaaS) concept. The use of MBaaS allows applications to focus on feature development without the need to manage servers, databases, authentication, and other services. The purpose of the research is to determine the performance of applications integrated with the cloud and determine the application performance load through the Quality of Service testing method. The results showed that the cloud with MBaaS technology showed superior performance in Quality of Service testing using Wireshark, the firestore database achieved a throughput of 10507 kbps, packet loss 0.0%, delay 0.0 ms, and jitter -0.003 ms. Based on performance analysis using monitoring tools from Google Cloud Platform services, application load balancer rule - request bytes data, is stable with an average request bytes of 500B/s with the highest request reaching 512.08B/s. Analysis of application load balancer rule - response bytes, is stable with an average response bytes of 500B/s and the highest response reaching 559.97B/s. Based on the research results it can be concluded that, cloud computing with MBaaS technology, is able to handle the load of requests with good enough performance. However, there are limitations on the ability to handle very high loads, such as increased errors and decreased throughput in stress testing.*

Keywords— application, cloud, firebase, google cloud platform, MBaaS.

I. PENDAHULUAN

A. Latar Belakang

Perkembangan teknologi informasi yang pesat telah membuka peluang besar bagi berbagai sektor industri, termasuk sektor kuliner, untuk mengadopsi teknologi digital guna meningkatkan efisiensi operasional dan daya saing. Di tengah kemajuan ini, *cloud computing* telah menjadi salah satu teknologi kunci yang mendorong transformasi digital, terutama melalui konsep *Mobile Backend as a Service (MBaaS)*.

Cloud computing adalah sebuah teknologi yang mengelola dan menyimpan data di server yang jauh, yang

memungkinkan orang untuk mengaksesnya melalui internet [1]. *MBaaS* merupakan platform yang menyediakan backend yang sudah siap digunakan oleh pengembang aplikasi mobile, sehingga memungkinkan mereka untuk fokus pada pengembangan fitur-fitur aplikasi tanpa harus memikirkan pengelolaan server, database, otentikasi pengguna, dan layanan lainnya yang mendukung operasional aplikasi. Dengan *MBaaS*, pengembangan aplikasi kuliner berbasis Android dapat dilakukan lebih cepat, efisien, dan terintegrasi dengan berbagai layanan *cloud* yang memungkinkan penyimpanan data, analisis, dan komunikasi real-time [2].

Di Indonesia setiap wilayah memiliki budaya, bahasa, dan makanan khas yang berbeda. Beberapa wilayah juga memiliki

bangunan, monument, dan tempat yang memiliki kisah yang unik [3]. Kuliner khas setiap daerah juga berbeda, khususnya di Aceh Utara, sektor kuliner merupakan salah satu sektor yang memiliki potensi besar namun belum sepenuhnya dioptimalkan melalui penerapan teknologi digital. Desa Bantayan, yang dikenal sebagai salah satu desa wisata terkemuka di Aceh Utara, memiliki kuliner khas seperti "Kuah Laot" yang menarik minat wisatawan. Namun, kurangnya penerapan manajemen digital dalam bisnis kuliner di desa ini menghambat perkembangan ekonomi lokal.

Implementasi aplikasi kuliner berbasis Android dengan dukungan MBaaS dapat menjadi solusi untuk mengatasi kendala ini. Aplikasi tersebut tidak hanya memudahkan pelanggan dalam melihat menu, memesan makanan, dan melakukan pembayaran secara online, tetapi juga memungkinkan pemilik usaha untuk memantau dan mengelola bisnisnya dari jarak jauh dengan lebih efisien. Semua data pengguna dan transaksi disimpan di cloud, sehingga aman dan dapat diakses kapan saja oleh pemilik bisnis.

Penelitian ini bertujuan untuk mengkaji implementasi cloud computing pada aplikasi kuliner berbasis Android dengan menggunakan teknologi MBaaS. Analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan adalah semua langkah dalam proses pengembangan aplikasi yang akan menggunakan metode *waterfall*. Evaluasi terhadap implementasi ini akan dilakukan untuk mengukur efektivitasnya dalam meningkatkan kualitas aplikasi, serta menyimpulkan hasil penelitian terkait dampak teknologi MBaaS terhadap bisnis kuliner.

Dengan adopsi teknologi MBaaS, diharapkan aplikasi kuliner ini tidak hanya meningkatkan efisiensi operasional, tetapi juga menjadi contoh bagi pengembangan bisnis kuliner di daerah-daerah lain yang memiliki potensi serupa.

B. Tinjauan Teoritis

Penelitian ini menggunakan teori yang sudah pernah dipublikasikan pada artikel jurnal penelitian sebelumnya dan buku publikasi yang pernah diterbitkan.

1) *Cloud Computing*: Salah satu metode terbaik untuk pengorganisasian data adalah komputasi awan, juga dikenal sebagai cloud, yang memberikan sumber daya kepada manusia melalui internet. Sistem cloud memiliki banyak teknologi yang memiliki metode dan protocol yang berbeda [4]. Teknologi ini menyediakan akses ke jaringan, keamanan, perangkat lunak aplikasi, dan tempat penyimpanan data yang terpusat di dalam lingkungan internet [1].

Adanya teknologi penyimpanan di awan, atau *cloud computing*, adalah penggunaan teknologi komputasi (komputasi) dalam suatu jaringan yang dikembangkan berbasis internet, yang memungkinkan pengguna menjalankan program atau aplikasi melalui komputer yang terkoneksi ke *internet*. Namun untuk meningkatkan kenyamanan dan kemudahan pengaksesan data di berbagai perangkat yang berbeda diperlukan sistem yang secara otomatis tersinkronisasi secara realtime. Dengan demikian dapat

diterapkan websocket sebagai solusi untuk sinkronisasi data secara realtime [5].

2) *Google Cloud Platform*: Google Cloud Platform (GCP) adalah salah satu produk layanannya yang membantu bisnis di seluruh dunia mengembangkan aplikasi bisnis dan menawarkan platform sebagai layanan. Setiap produk dan layanan yang ditawarkan oleh Google Cloud Platform (GCP) memiliki serangkaian fitur dan manfaat yang berbeda yang sesuai dengan kebutuhan dan dapat digunakan untuk membangun infrastruktur *server* dengan tingkat standar yang tinggi.

3) *Firebase*: Gunakan Firebase, layanan yang dimiliki oleh Google, untuk membantu *developer* aplikasi membuat aplikasi yang lebih mudah. *Firebase Cloud Service Provider* dan *Backend as a Service* adalah solusi yang ditawarkan oleh Google yang membantu *developer* membuat aplikasi yang berjalan di ponsel atau di internet. Dengan Firebase, para *developer* aplikasi dapat fokus pada mengembangkan aplikasi mereka tanpa perlu membangun fitur-fitur yang dibuat pada *backend* dan infrastruktur dari awal. Ini memungkinkan para *developer* untuk mengembangkan aplikasi yang luar biasa tanpa mengeluarkan banyak upaya. Firebase memiliki banyak SDK (*Software Development Kit*) yang memungkinkan untuk mengintegrasikan layanan ini dengan beberapa platform seperti Android, iOS, JavaScript, C++, hingga Unity [6].

Firebase adalah *API (Application Programming Interface)* yang disediakan oleh Google yang memungkinkan aplikasi berbasis Android, iOS, atau web menyimpan dan menyesuaikan data. *Realtime database* adalah salah satu fasilitas yang memungkinkan data disimpan ke database dan diambil darinya dengan cepat, tetapi firebase memiliki banyak fitur lebih dari itu, seperti authenticator, database, penyimpanan, *hosting*, pemberitahuan, dan banyak lagi [7].

4) *Mobile Backend as a Service*: Banyak layanan teknologi baru digunakan dalam *cloud computing*, seperti *Mobile Backend as a Service (MBaaS)*, model layanan yang memungkinkan pengembang aplikasi web dan *mobile* menghubungkan aplikasi mereka ke penyimpanan *cloud backend* melalui *API*. Aplikasi backend juga menyediakan fitur seperti pengelolaan pengguna, pemberitahuan push, dan integrasi dengan layanan jejaring sosial. Layanan *cloud mobile backend* ini disediakan melalui penggunaan *Software Development Kit* dan antarmuka aplikasi pemrograman (*API*) [8].

5) *Perbedaan MbaaS Dengan Tegnologi Lain*: Perbandingan MBaaS dengan teknologi cloud lainnya: *IaaS (Infrastructure as a Service)*, *PaaS (Platform as a Service)*, dan *SaaS (Software as a Service)*, perbedaan utama mereka terletak pada tingkat abstraksi dan fokus layanan.

- *Infrastructure-as-a-Services (IaaS)*. IaaS Memberikan akses langsung ke sumber daya *hardware*, seperti

server, storage, dan jaringan. Pengembang memiliki kendali penuh atas infrastruktur, namun juga harus mengelola semuanya secara manual. Amazon Web Services, Microsoft Azure, IBM, dan Google Cloud Platform adalah vendor IaaS utama.

- *Platform-as-a-Services (PaaS)*. Layanan PaaS menyediakan hardware, pengembang aplikasi dapat fokus pada aplikasi yang akan dikembangkan tanpa harus memikirkan tentang *scaling infrastruktur*, *load balancing*, operasi sistem, dan masalah lainnya [9]. Sebagian besar PaaS dibuat untuk pengembang aplikasi yang ingin membuat proses pembuatan dan penggunaan perangkat lunak menjadi lebih mudah. Misalnya, pengembang web mungkin menggunakan PaaS yang mencakup perangkat lunak sistem operasi, perangkat lunak server web, basis data, dan alat pengembangan web yang terkait. Beberapa vendor PaaS terkemuka di antaranya adalah, Amazon Web Services, Microsoft Azure, IBM, dan GCP [2].
- *Software-as-a-Services (SaaS)*. SaaS merupakan adalah model layanan dalam komputasi awan di mana perangkat lunak atau aplikasi disediakan kepada pengguna melalui internet sebagai layanan [10]. Beberapa contoh SaaS yang terkenal di antaranya Salesforce, Microsoft Office 365, Google G Suite, Dropbox, Adobe Creative Cloud, dan lainnya [2].

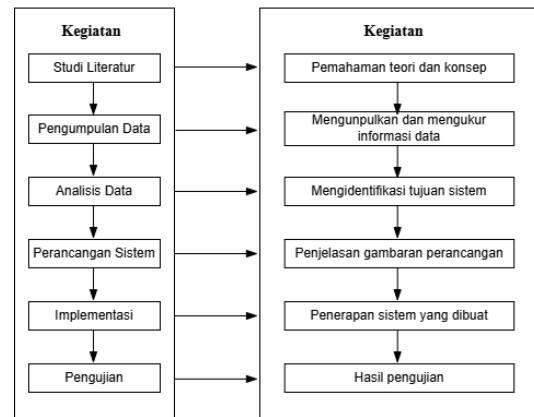
MBaaS berada di antara PaaS dan SaaS. MBaaS memberikan tingkat abstraksi yang lebih tinggi daripada PaaS, sehingga pengembang dapat fokus pada pengembangan fitur aplikasi tanpa perlu khawatir tentang infrastruktur. Namun, MBaaS tidak se-abstrak SaaS, karena pengembang masih perlu melakukan beberapa konfigurasi dan pengembangan kustom.

II. METODOLOGI PENELITIAN

A. Metode Penelitian

Model siklus hidup perangkat lunak, juga dikenal sebagai model *waterfall*, menekankan linieritas. Ada tanggung jawab untuk setiap tahap, dan setiap tahap harus diselesaikan sebelum dilanjutkan ke tahap berikutnya. Karakteristiknya yang sangat terstruktur dan linier adalah ciri khas model ini.

Langkah-langkah metode *waterfall* yang digunakan dalam penelitian ini adalah sebagai berikut, lihat gambar 1.



Gambar 1. Tahapan pengujian

Gambar 1 menunjukkan tahapan penelitian, dan pembahasan setiap tahapan dapat diuraikan sebagai berikut:

1) *Studi literatur*: Pada tahap ini, dilakukan penelusuran literatur terkait dengan judul pengujian ini, seperti buku, jurnal ilmiah, referensi-referensi melalui perpustakaan dan internet, antara lain.

2) *Pengumpulan Data*: Pada langkah ini, data dikumpulkan dalam dua bentuk: data primer dari wawancara dan observasi di lokasi desa wisata Bantayan, dan data sekunder dari artikel dan jurnal.

3) *Analisis Sistem*: Analisis sistem juga digunakan untuk merancang struktur *front-end* dan *back-end*. Ini adalah tahapan di mana data dikumpulkan, yang akan digunakan untuk mengidentifikasi masalah yang terjadi dan memudahkan langkah selanjutnya.

4) *Perancangan Sistem*: Perancangan sistem adalah sebuah aktivitas untuk merancang sistem *front-end* dan *back-end* seperti, diagram use case, activity diagram, user interface, dan perancangan database.

5) *Implementasi*: Pada tahap ini menerapkan rancangan aplikasi sesuai dengan rancangan yang sudah direncanakan. Implementasi aplikasi dilakukan secara linear untuk memastikan semua tahap rancangan terselesaikan.

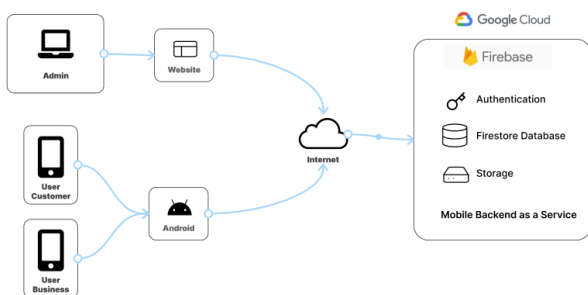
6) *Pengujian Sistem*: Pada tahap ini akan dilakukan pengujian dengan menggunakan metode *QoS (Quality of Service)* untuk mengukur kualitas jaringan latency dan delay dengan menggunakan aplikasi Wireshark.

Pemilih metode pengembangan *waterfall* untuk memastikan tahapan yang linier, termasuk analisis, desain, implementasi, pengujian, dan pemeliharaan, sehingga setiap langkah dapat disesuaikan dengan kebutuhan proyek secara lebih sistematis.

Tujuan penelitian mencakup peningkatan kualitas aplikasi dan kepuasan pengguna, dengan pengumpulan data melalui survei dan analisis penggunaan fitur obrolan. Proses implementasi *cloud* computing melibatkan identifikasi kebutuhan infrastruktur cloud, konfigurasi *MBaaS*, dan integrasi fitur *chat* dengan layanan *cloud*. Evaluasi keseluruhan akan dilakukan untuk mengukur efektivitas implementasi *cloud* dan fitur obrolan, serta menyimpulkan hasil penelitian terkait peningkatan kualitas aplikasi dan kepuasan pengguna.

B. Perancangan Sistem

Agar penelitian ini berjalan dengan lancar dan sistematis, diperlukan sebuah perancangan sistem yang jelas. Perancangan awal sistem yang akan digunakan dalam penelitian ini dapat dilihat pada gambar 2.



Gambar 2. Rancangan Sistem

Gambar 2 menunjukkan detail perancangan sistem. Dalam sistem, ada tiga tingkat peran pengguna: *admin*, *customer*, dan *business*. Ada dua jenis aplikasi dalam sistem, website digunakan secara khusus oleh admin, dan aplikasi android digunakan oleh akun *level-customer*, dan *level-business*. Kedua jenis aplikasi terhubung dengan internet untuk mendapatkan data dari *cloud*.

Penggunaan cloud dengan teknologi *MBaaS* dalam sistem untuk mempermudah perancangan keamanan basis data. Teknologi *MBaaS* menyediakan fitur siap pakai dengan teknologi terbaru. Fitur *cloud* yang digunakan dalam sistem adalah sebagai berikut :

1) *Firestore Database*: Firestore Database adalah layanan NoSQL cloud-based yang disediakan oleh Google Cloud Platform. Ia menawarkan fleksibilitas dan skalabilitas tinggi untuk penyimpanan data realtime. Berbeda dengan database relasional tradisional yang menggunakan tabel dan skema yang kaku, Firestore menggunakan koleksi dokumen yang terstruktur dalam format JSON (JavaScript Object Notation). Hal ini memungkinkan data untuk diakses dan diperbarui secara real-time, sehingga cocok untuk aplikasi yang membutuhkan interaksi pengguna yang cepat dan dinamis. Firestore juga menyediakan fitur-fitur seperti query, indexing, dan keamanan untuk membantu mengelola data dengan efisien [6].

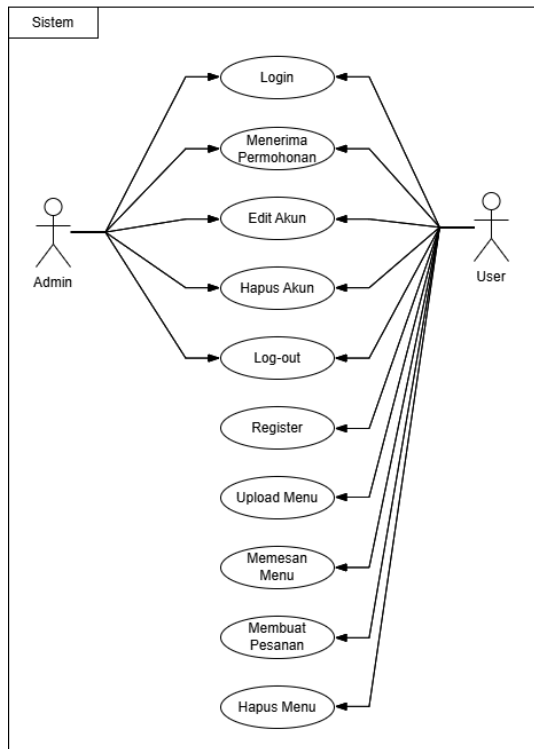
2) *Firestore Database*: Firestore Database adalah layanan NoSQL cloud-based yang disediakan oleh Google Cloud Platform. Ia menawarkan fleksibilitas dan skalabilitas tinggi untuk penyimpanan data realtime. Berbeda dengan database relasional tradisional yang menggunakan tabel dan skema yang kaku, Firestore menggunakan koleksi dokumen yang terstruktur dalam format JSON (JavaScript Object Notation). Hal ini memungkinkan data untuk diakses dan diperbarui secara real-time, sehingga cocok untuk aplikasi yang membutuhkan interaksi pengguna yang cepat dan dinamis. Firestore juga menyediakan fitur-fitur seperti query, indexing, dan keamanan untuk membantu mengelola data dengan efisien [6].

3) *Firebase Storage*: Firebase Storage adalah layanan penyimpanan cloud yang disediakan oleh Google Cloud Platform. Firebase Storage memungkinkan menyimpan dan mengelola file seperti gambar, video, dan dokumen secara aman dan efisien. Firebase Storage terintegrasi dengan Firebase Authentication, sehingga dapat mengontrol akses pengguna ke file-file mereka. Selain itu, Firebase Storage juga menyediakan fitur-fitur seperti aturan keamanan, metadata, dan integrasi dengan produk Firebase lainnya seperti Cloud [6].

C. Perancangan UML Diagram Use Case

Sebuah standar untuk mendokumentasikan dan membangun perangkat lunak dalam bentuk diagram adalah *Unified Modeling Language (UML)*, yang digunakan untuk membangun system. Untuk menjelaskan bagaimana produk atau sistem tertentu digunakan, dapat menggunakan diagram kasus. Diagram ini dapat dianggap sebagai diagram yang paling mudah karena pembuat umumnya memahami kegunaan produk mereka. Kegunaan produk akan terlihat secara keseluruhan, meskipun ini sederhana. Ini akan memberikan gambaran umum tentang bisnis [11].

Dengan menggunakan metodologi berorientasi objek *Unified Modeling Language*, perancangan sistem akan digambarkan dalam beberapa bentuk diagram, seperti *use case diagram* dan *activity diagram* [12].



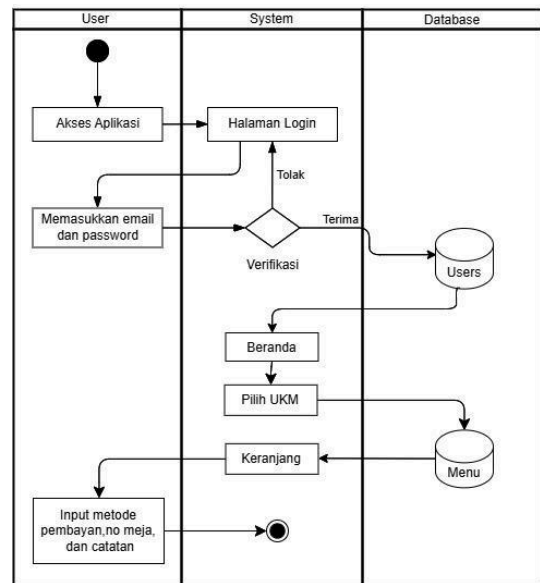
Gambar 3. Rancangan use case pada sistem

Berdasarkan rancangan sistem, dua user dengan dua jenis aplikasi, website dan aplikasi android. Aplikasi web memiliki fungsi yang berbeda dari aplikasi android, maka secara menyeluruh use case sistem.

D. Perancangan UML Activity Diagram

Diagram aktivitas, atau diagram aktivitas, adalah jenis diagram alir yang membantu memetakan berbagai kegiatan secara berurutan. Masing-masing tugas akan dihubungkan ke garis yang menunjukkan urutannya dalam proses operasi yang direncanakan. [11].

Perancangan aktivitas diagram pada sistem aplikasi pemesanan kuliner di mulai dari aktivitas login dan berakhir sampai proses pemesanan selesai, lihat gambar 4.



Gambar 4. Alur pemesanan pengguna aplikasi

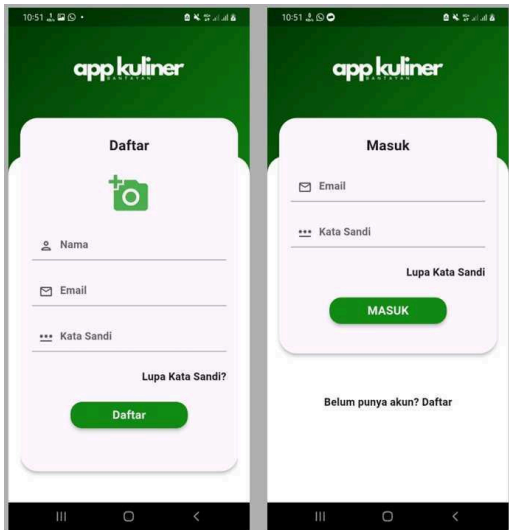
Berdasarkan gambar 4 diagram activity ini menggambarkan proses transaksi pada sebuah aplikasi pemesanan makanan atau menu dari UKM. Pengguna diajak untuk login terlebih dahulu. Setelah berhasil login, pengguna akan masuk ke dalam tampilan utama yang menampilkan UKM terdaftar. Pengguna dapat memilih UKM favorit dan menu yang tersedia dalam UKM tersebut, kemudian pengguna menambahkan menu ke keranjang sebelum pesanan dibuat.

III. HASIL DAN PEMBAHASAN

A. Implementasi User Interface Pada Aplikasi

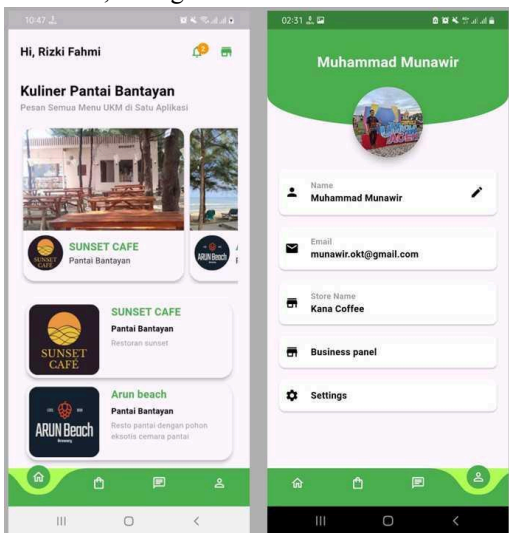
Pengembangan UI (User Interface) aplikasi mengikuti rancangan yang telah dibuat. Bagian ini menjelaskan fungsi dari setiap halaman yang terkait meliputi tampilan user interface customer dan business. Implementasi interface adalah sebuah tampilan pada suatu sistem yang memungkinkan pengguna untuk berinteraksi dengan perangkatnya secara langsung atau melalui jaringan rangkaian perangkat [13].

1) *Tampilan Layar Register Dan Login:* Pada layar register antarmuka pengguna, pengguna dapat melakukan pendaftaran akun dengan sistem. Layar register merupakan layar pilihan bagi pengguna yang belum memiliki. Layar login aplikasi kuliner ini memiliki desain yang sederhana dan intuitif. Terdapat judul "Masuk" yang jelas di bagian atas, diikuti oleh dua field input utama: email dan password. Desain field password dengan titik-titik (***) merupakan standar keamanan yang baik untuk menyembunyikan karakter password, lihat gambar 5.



Gambar 5. Tampilan layar register dan login

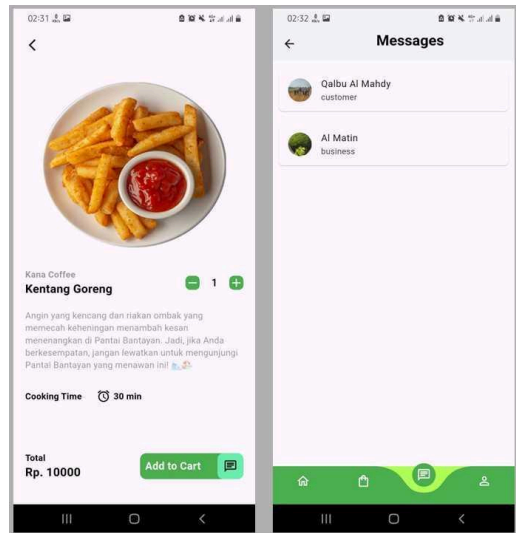
2) *Tampilan Layar Beranda Dan Detail Pengguna:* Layar beranda aplikasi ini menampilkan desain yang sederhana dan informatif. Terdapat sapaan personal untuk pengguna "Hi, User". Di bagian bawah layar, terdapat ikon navigasi yang umum digunakan pada aplikasi *mobile*, seperti ikon beranda, keranjang belanja, notifikasi, dan profil. Layar detail pengguna ini menampilkan informasi pribadi dan bisnis pengguna dengan tampilan yang sederhana dan mudah dipahami. Terdapat foto profil pengguna di bagian atas, nama lengkap, alamat email, nama toko, dan dua opsi menu untuk mengelola bisnis, lihat gambar 6.



Gambar 6. Tampilan layar beranda dan detail pengguna

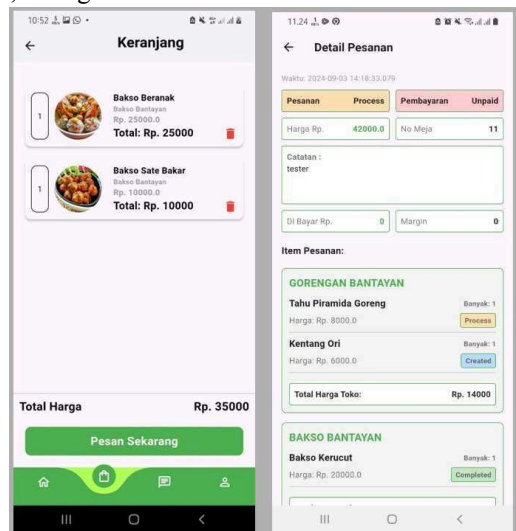
3) *Tampilan Layar Detail Item Dan Obrolan:* Pada layar detail item dirancang untuk menampilkan gambar dari sebuah menu. Gambar menu diambil sesuai dengan gambar yang akan diunggah oleh user business. Pelanggan dapat menambah menu sesuai keinginan, dan pelanggan melakukan penambahan dalam keranjang atau melakukan obrolan dengan

akun bisnis. Layar obrolan menampilkan profil dari akun tujuan obrolan dan level dari akun tujuan obrolan, lihat gambar 7.



Gambar 7. Tampilan layar detail item dan obrolan

4) *Perancangan Layar Keranjang Dan Detail Pesanan:* Pada layar keranjang memperlihatkan item yang masuk keranjang, setiap item memiliki detail informasi seperti nama produk, nama restoran UKM, dan nilai harga produk. Tampilan detail pesanan dirancang menampilkan rincian dari pesanan yang sudah dilakukan, seperti status pesanan, status pembayaran, total harga yang harus dibayar dan lokasi meja pemesan. Tampilan detail pesanan menampilkan total nilai yang telah dibayar dan *margin* selisih harga pembayaran. Setiap itema pesanan memiliki rincian status pesanan dari setiap item yang dipesan, lihat gambar 8.



Gambar 8. Tampilan layar keranjang dan detail pesanan

B. Pengujian

Melihat respon aplikasi terhadap layanan *cloud* dari Google Cloud Platform (GCP) dapat menggunakan tools *Monitoring* yang disediakan GCP. Aplikasi dipantau berdasarkan tindakan nyata dari 10 perangkat android. Rumus *Quality of Service* digunakan untuk menguji performa *cloud*. Tujuan rumus ini adalah untuk mengukur kualitas layanan yang diberikan oleh suatu sistem atau layanan *cloud*.

1) *Pengujian Kualitas Dengan Tools Monitoring GCP*: Google cloud platform menyediakan *tools monitoring* yang dapat menampilkan hasil kinerja aplikasi secara *real-time*.

Menganalisa *application load balancer rule - request bytes*. Selama percobaan dalam rentang waktu dari pukul 7:30 AM hingga 10:00 AM dalam zona waktu UTC+7. Jumlah total byte yang diterima oleh aplikasi melalui *load balancer* eksternal dalam rentang waktu tertentu, ditampilkan dalam grafik gambar 9 berikut:



Gambar 9. Grafik Application Load Balancer Rule - Request Bytes

Berdasarkan gambar 9 menjelaskan bahwa grafik menunjukkan fluktuasi yang sangat tinggi pada jumlah byte yang diterima. Ada periode di mana jumlah *byte* sangat rendah, dan ada pula periode di mana jumlah *byte* melonjak tinggi samapai mencapai 512.08B/s. Terdapat beberapa puncak yang sangat tajam, mengindikasikan adanya lonjakan trafik yang tiba-tiba. Secara umum, tidak terlihat pola yang jelas dan konsisten dalam fluktuasi ini.

Menganalisa *application load balancer rule - response bytes*. Selama percobaan dalam rentang waktu dari pukul 7:30 AM hingga 10:00 AM dalam zona waktu UTC+7. Jumlah total response byte yang diterima oleh aplikasi melalui *load balancer* eksternal dalam rentang waktu tertentu, ditampilkan dalam grafik gambar 10 berikut:



Gambar 10. Grafik Application Load Balancer Rule - Response Bytes

Berdasarkan gambar 10 menjelaskan grafik menunjukkan fluktuasi yang sangat tinggi pada jumlah *byte* yang dikirimkan. Ada periode di mana jumlah *byte* sangat rendah, dan ada pula periode di mana jumlah *byte* melonjak tinggi mencapai 559.97B/s. Terdapat beberapa puncak yang sangat tajam, mengindikasikan adanya lonjakan permintaan dan pengiriman data yang tiba-tiba. Secara umum, tidak terlihat pola yang jelas dan konsisten dalam fluktuasi ini.

2) *Pengujian Kualitas Menggunakan Metode QoS*: Penilaian kualitas fitur pengiriman tugas secara *real-time* dari pelanggan dalam aplikasi dilakukan dengan menerapkan metode *Quality of Service* menggunakan perangkat lunak Apache JMeter dan Wireshark. Pengujian performa fitur ini melibatkan pengukuran parameter throughput, packet loss, delay, dan jitter untuk memastikan bahwa fitur tersebut dapat berfungsi secara optimal.

Pengujian ini memfokuskan pada endpoint API yang telah di monitoring selama sistem aplikasi berjalan dengan HTTP Toolkit, dan menemukan API <https://firestore.googleapis.com/google.firestore.v1.Firestore/Listen/channel>. Aplikasi yang menggunakan *Software Development Kit* sebagai media penghubung, semua *API* telah disediakan dalam *SDK*. Pengujian *Quality of Service* bertujuan untuk mengontrol lalu lintas dan memastikan kinerja aplikasi kritis dengan kapasitas jaringan yang terbatas.

Untuk menguji performa aplikasi, dilakukan simulasi dengan mengirim *HTTP request* dengan berbagai jumlah request, yaitu 50, dan 100. Setiap jumlah permintaan merepresentasikan jumlah pengguna virtual yang mengirim data secara bersamaan ke server. Jumlah permintaan ini ditentukan oleh tiga parameter, yaitu jumlah thread, periode *ramp-up*, dan jumlah pengulangan. Periode *ramp-up* disetel selama 1 detik dan jumlah pengulangan dilakukan sebanyak satu kali. Pengujian dilakukan menggunakan aplikasi Apache Jmeter dan monitoring menggunakan wireshark.

Pengujian pengiriman data dari pengguna ke server menggunakan 50 Permintaan dilakukan dengan Apache JMeter dan Wireshark. Apache JMeter mensimulasikan permintaan dari 50 pengguna untuk mengukur waktu respons, throughput, dan tingkat kesalahan, sementara Wireshark menganalisis lalu lintas jaringan untuk mendapatkan wawasan mendalam mengenai latensi dan potensi masalah. Data dapat dilihat pada tabel 1.

TABEL 1
HASIL PENGUJIAN DENGAN 50 REQUEST

Parameter QoS	Hasil Perhitungan	Kategori
Throughput	$= \frac{\text{paket yang diterima}}{\text{waktu pengamatan}} = \frac{61710}{5,011} = 12,315 \text{ bits}$	Bagus
Packet Loss	$= \frac{\text{paket yang dikirim} - \text{paket data diterima}}{\text{paket yang dikirim}} \times 100\%$	Sangat Bagus

Parameter QoS	Hasil Perhitungan	Kategori
Delay	$= \frac{\text{total delay}}{\text{paket yang diterima}} = \frac{0,0}{475} = 0,0 \text{ s} \times 1000 =$	Sangat Bagus
Jitter	$= \frac{\text{total variasi delay}}{\text{paket yang diterima} - 1} = \frac{0,0}{474} = 0,000 \text{ s} \times 10$	Bagus

Hasil pengujian pada tabel 4.2 menunjukkan bahwa nilai throughput mencapai 98 kbits permintaan per detik, dan packet loss mencapai 0%. delay yang tercatat adalah 0,000 ms. jitter yang tercatat adalah -0,001 ms.

Pengujian pengiriman data dari pengguna ke server menggunakan 100 Permintaan dilakukan dengan Apache JMeter dan Wireshark. Apache JMeter mensimulasikan permintaan dari 100 pengguna, sementara Wireshark menganalisis lalu lintas jaringan untuk mendapatkan wawasan mendalam mengenai latensi dan potensi masalah. Data dapat dilihat pada tabel 2.

TABEL 2
HASIL PENGUJIAN DENGAN 100 REQUEST

Parameter QoS	Hasil Perhitungan	Kategori
Throughput	$= \frac{\text{paket yang diterima}}{\text{waktu pengamatan}} = \frac{7894974}{6011} = 1313,421 \text{ t}$	Bagus

$$\text{Packet Loss} = \frac{\text{paket dikirim} - \text{paket diterima}}{\text{paket yang dikirim}} \times 100\% = \frac{8506 - 8506}{8506} \text{ Sangat Bagus}$$

$$\text{Delay} = \frac{\text{total delay}}{\text{paket yang diterima}} = \frac{0,0}{8506} = 0,0 \text{ s} \times 1000 = \text{Sangat Bagus}$$

$$\text{Jitter} = \frac{\text{total variasi delay}}{\text{paket yang diterima} - 1} = \frac{0,028}{8505} = 0,000 \text{ s} \times 10 \text{ Bagus}$$

Hasil pengujian pada tabel 4.3 menunjukkan bahwa nilai throughput mencapai 10507 kbits permintaan per detik, dan packet loss mencapai 0%. delay yang tercatat adalah 0,000 ms. jitter yang tercatat adalah -0,003 ms.

IV. KESIMPULAN

Berdasarkan hasil analisa menggunakan tools monitoring dari google cloud platform. Dan pengujian *Quality of Service (QoS)*, jaringan menunjukkan performa yang baik dalam menangani beban tinggi.

Analisa *application load balancer rule - request bytes*, berada stabil dengan rata-rata *request bytes* 500b/s. Analisa *application load balancer rule - response bytes*, berada stabil dengan rata-rata *response bytes* 500b/s.

Throughput meningkat seiring dengan peningkatan jumlah permintaan, dengan rata-rata sebesar 10507 kbps. Meskipun packet loss tetap rendah hingga 100 permintaan, terdapat peningkatan kecil menjadi 0,047% pada 100 permintaan, menunjukkan adanya batasan kapasitas jaringan saat beban sangat tinggi. *Delay* tetap pada posisi 0,0 ms sampai 100 permintaan. Jitter mengalami penurunan seiring bertambahnya permintaan, menunjukkan stabilitas yang cukup baik dalam pengiriman data.

Infrastruktur *cloud* yang menggunakan layanan GCP yang terintegrasi dengan Firebase untuk mendapatkan layanan *MBaaS*. dengan *firestore database* saat ini mampu menangani beban permintaan dengan performa yang cukup baik. Namun, terdapat batasan pada kemampuan penanganan beban yang sangat tinggi, seperti yang terlihat dari peningkatan *error* dan penurunan throughput pada pengujian *stress testing*. Hal ini menunjukkan bahwa sistem *cloud* GCP dengan dengan teknologi *MBaaS* dari Firebase memerlukan optimasi lebih lanjut untuk memastikan layanan tetap stabil di bawah kondisi beban puncak.

REFERENSI

- [1] A. Nanda, H. Toha Hidayat, and Mahlil, "Implementasi Cloud Computing Untuk Media Pembelajaran Interaktif Bahasa Inggris

- Berbasis Android,” *JAISE : Journal of Artificial Intelligence and Software Engineering*, vol. 3, no. 2, 2023.
- [2] Aris Cahyadi Risdianto and Onno W. Purbo, *TEKNOLOGI VIRTUALISASI: Fondasi Utama Layanan Berbasis Cloud*. Penerbit Andi, 2021.
- [3] Mursyidah, Husaini, and Herri Mahyar, “Aplikasi Augmented Reality Pengenalan Landmark Pariwisata Aceh,” *Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan*, vol. 8, no. 1, pp. 45–50, Jun. 2023.
- [4] Nadya Apriliani, Arief Riski Indra Pratama, and Agung Susilo Yuda Irawan, “Perbandingan antara Teknologi dan Peforma pada Komputasi Awan (Cloud Computing),” *Infotek*, vol. 5, no. 2, Jul. 2022.
- [5] M. Mustafa, Anwar, and Indrawati, “Prototipe Sinkronisasi Penyimpanan Dokumen Secara Realtime Menggunakan Websocket Pada Cloud Storage,” *Jurnal Teknologi Rekayasa Informasi dan Komputer*, vol. 7, no. 1, Sep. 2023.
- [6] Rosyana Fitria Purnomo, Onno W. Purbo Dan Rz. Abd. Aziz, and RZ. Abd. A. Onno W. Purbo, *Firebase: Membangun Aplikasi Berbasis Android*. Yogyakarta: Andi Offset, 2020.
- [7] G. R. Payara and Radius Tanone, “Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 4, no. 3, pp. 397–406, Dec. 2018.
- [8] A. Hidayat and A. Sintawati, “Aplikasi Sistem Informasi Perusahaan Otobus Pariwisata Di Dki Jakarta Menggunakan Teknologi Android MbaaS,” *Jurnal Ilmiah Informatika Komputer*, vol. 24, no. 1, pp. 61–75, 2019, doi: 10.35760/ik.2019.v24i1.1990.
- [9] Yunie Cancer and Zikrul Alim, “Platform as a Service (PaaS) Sebagai Layanan Sistem Operasi Cloud Computing,” *Jurnal TIMES*, vol. 5, no. 1, pp. 32–35.
- [10] Novira Dwina, Nisha Khairani, Muhammad Nasir, and Indrawati, “Penerapan Metode Advanced Encryption Standard pada Sistem Penyimpanan Data Menggunakan Cloud Computing Sebagai Software-as-a-Service,” *JAISE : Journal of Artificial Intelligence and Software Engineering*, vol. 3, no. 1, 2023.
- [11] Erwin Sutanto, *Pemrograman Android Dengan Menggunakan Eclipse & StarUML*. Indonesia: Airlangga University Press, 2020.
- [12] Andri Irwan Zahri, Hari Toha Hidayat, and Aswandi, “Rancang Bangun E-Commerce Menggunakan QR-Code pada Perusahaan Retail Berbasis Android,” *eProceeding of TIK*, vol. 1, no. 2, pp. 12–20, 2021.
- [13] Risma Arifa, Husaini, and F. Yanuar Rudi F, “Implementasi Aplikasi Enkripsi Dekripsi File Dokumen Menggunakan Algoritma Aes Pada Cloud Computing (Studi Kasus Prodi Teknologi Rekayasa Komputer Jaringan),” *JAISE : Journal of Artificial Intelligence and Software Engineering*, vol. 4, no. 1, 2024.