Implementasi Cloud Computing Pada Jasa Pemesanan Teknisi Berbasis Android Menggunakan Amazon Web Services

Rizki Ilhami[1], Indrawati[2], Ilham Safar[3]

[1,2,3] *Jurusan Tekniknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe*
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA
[1]rizkiilhami27@gmail.com
[2]indrawati@pnl.ac.id (penulis korespondensi)
[3]ilham_safar@pnl.ac.id

***Abstrak*— Untuk mendistribusikan *backend* aplikasi secara lebih aman dan tangguh terhadap bencana alam, diperlukan server dengan kualitas jaringan yang stabil serta kemampuan skalabilitas yang baik. Teknologi *cloud computing*, khususnya *Amazon Web Services* (*AWS*), menawarkan solusi efektif untuk memenuhi kebutuhan ini, memastikan *respons* optimal dan pengalaman pengguna yang memuaskan. Penelitian ini bertujuan untuk mengeksplorasi penggunaan layanan *AWS*, terutama *EC2 Instances tipe t2.micro*, dalam meningkatkan ketersediaan, aksesibilitas, dan keamanan aplikasi. Metode yang digunakan dalam penelitian ini meliputi analisis kinerja *EC2 Instances tipe t2.micro* melalui *stress testing* menggunakan *JMeter*, serta evaluasi kualitas jaringan dengan mengukur parameter *Quality of Service (QoS)* sebagai indikator utama efisiensi layanan jaringan. Hasil pengujian *stress testing* menunjukkan bahwa *EC2 Instances tipe t2.micro* dapat menangani beban hingga 5000 permintaan dengan *throughput* mencapai 224,6 permintaan per detik dan *error* 57,40%. Namun, pada beban 7000 permintaan, tingkat *error* meningkat hingga 50,67% dengan *throughput* turun menjadi 17,1 permintaan per detik. Pengujian *QoS* menggunakan *Wireshark* menunjukkan bahwa *EC2 Instances tipe t2.micro* mencapai *throughput* rata-rata 364,282 *kbps*, *packet loss* rata-rata 0,216%, *delay* rata-rata 8,684 *ms*, dan *jitter* rata-rata 8,419 *ms*. Hasil penelitian ini mengindikasikan bahwa AWS EC2 Instances tipe t2.micro mampu memberikan peningkatan kinerja aplikasi, meskipun diperlukan perbaikan tambahan untuk memastikan stabilitas sistem di bawah kondisi beban tinggi.*
***Kata kunci*— Layanan, Teknisi, Amazon Web Services, Cloud, Android.**

Abstract*— *To distribute backend applications more securely and resiliently against natural disasters, servers with stable network quality and good scalability are required. Cloud computing technology, specifically Amazon Web Services (AWS), offers effective solutions to meet these needs, ensuring optimal responsiveness and user experience. This research aims to explore the use of AWS services, particularly EC2 Instances of the t2.micro type, in enhancing the availability, accessibility, and security of applications. The methods used in this research include analyzing the performance of EC2 Instances t2.micro through stress testing using JMeter, as well as evaluating network quality by measuring Quality of Service (QoS) parameters as the main indicators of network service efficiency. The stress testing results show that EC2 Instances t2.micro can handle loads up to 5000 requests with a throughput of 224.6 requests per second and an error rate of 57.40%. However, at a load of 7000 requests, the error rate increases to 50.67%, with throughput dropping to 17.1 requests per second. QoS testing using Wireshark indicates that EC2 Instances t2.micro achieve an average throughput of 364.282 kbps, an average packet loss of 0.216%, an average delay of 8.684 ms, and an average jitter of 8.419 ms. These results suggest that AWS EC2 Instances t2.micro can enhance application performance, although further improvements are needed to ensure system stability under high load conditions.
***Keywords*— Services, Technicians, Amazon Web Services, Cloud, Android.**

I. Pendahuluan

A. Latar Belakang

Dalam era digital yang terus berkembang, kebutuhan akan layanan teknisi untuk memperbaiki dan merawat perangkat elektronik terus meningkat. Masyarakat sering mengalami kesulitan dalam menemukan dan memesan teknisi yang dapat diandalkan untuk menangani berbagai masalah perangkat keras, perangkat lunak, atau instalasi perangkat. Untuk mengatasi tantangan ini, pemanfaatan teknologi mobile, terutama melalui aplikasi berbasis Android, menjadi solusi yang sangat efektif.

Untuk menjalankan sistem secara optimal, diperlukan perangkat keras (server) agar data dapat diproses secara real-time. Namun, penggunaan perangkat keras ini memerlukan investasi biaya yang cukup signifikan. Dalam konteks perkembangan teknologi saat ini, Cloud Computing muncul sebagai alternatif yang menjanjikan untuk menggantikan perangkat keras dalam menjalankan sistem secara real-time.

Implementasi Cloud Computing dilakukan dengan tujuan mengoptimalkan biaya pembuatan sistem. Pada penelitian ini, Cloud Computing yang akan diimplementasikan adalah Amazon Web Services. Amazon Web Services sebagai platform Cloud Computing menyediakan berbagai layanan yang mendukung pengembangan aplikasi mobile secara efisien, termasuk keamanan data, dan manajemen pengguna. Pengujian pada penelitian ini akan dilakukan menggunakan Quality of Service (QOS) guna memastikan bahwa sistem berfungsi sesuai harapan.

Tujuan pengembangan pemesanan teknisi berbasis Android adalah untuk memudahkan pengguna mencari, dan

memesan layanan teknisi dengan cepat dan efisien. Seiring dengan peningkatan penggunaan perangkat Android di masyarakat, sistem ini menawarkan solusi yang efektif untuk kendala pencarian teknisi konvensional, yang sering memakan waktu dan memerlukan banyak upaya.

Dengan demikian, penelitian ini diharapkan dapat menciptakan ekosistem yang saling menguntungkan bagi kedua belah pihak, menyederhanakan pengalaman pengguna dalam mendapatkan layanan teknis yang diperlukan, dan membuka peluang bisnis yang lebih luas bagi teknisi.

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

Cloud computing adalah suatu mekanisme di mana sekelompok sumber daya Teknologi Informasi dan Komunikasi (TIK) saling terhubung dan hampir tanpa batas. Cloud computing merupakan sebuah teknologi yang mengumpulkan manajemen dan penyimpanan data dari server yang berada di lokasi yang jauh, agar data tersebut dapat diakses melalui internet [1]. Infrastruktur dan aplikasi dalam *cloud computing* dimiliki dan dikelola sepenuhnya oleh penyedia layanan pihak ketiga. Hal ini memungkinkan pelanggan untuk menggunakan sumber daya tersebut sesuai kebutuhan melalui jaringan, baik itu jaringan *private* maupun *public*, secara *on-demand*.[2]

Pengguna layanan *cloud computing* dapat mengakses berkas secara langsung melalui internet tanpa memerlukan pemasangan pada perangkat komputer lokal.[2]

Cloud Computing merupakan model client-server, arsitektur client-server. *Cloud Computing* memungkinkan pengguna untuk mengakses sumber daya termasuk *server*, penyimpanan, jaringan, dan perangkat lunak dari mana saja kapan saja sebagai sebuah layanan.[3]

Cloud computing adalah penggunaan teknologi komputasi dalam jaringan yang dikembangkan melalui internet (cloud) dan berfungsi untuk menjalankan program atau aplikasi melalui komputer yang terhubung ke internet.[4]

2) Amazon Web Services

Amazon Web Services (AWS) adalah platform *cloud* paling komprehensif dan digunakan secara luas di dunia, menawarkan lebih dari 175 layanan unggulan yang lengkap dari pusat data secara global. Jutaan pelanggan termasuk beberapa startup dengan pertumbuhan tercepat, perusahaan terbesar, dan lembaga pemerintah terkemuka menggunakan *AWS* untuk memangkas biaya, menjadi lebih sigap, dan inovasi lebih cepat.[5]

3) Node.js

Node.js adalah lingkungan *runtime JavaScript open-source* dan *cross-platform*. Ini adalah platform yang populer untuk hampir semua jenis proyek. *Node.js* berjalan dimesin *JavaScript* V8 sehingga dapat mengeksekusi kode *JavaScript* di luar *browser web*. *Node.js* dirancang untuk membangun jaringan aplikasi yang *scalable* sehingga sangat cocok untuk membuat aplikasi jaringan *real-time*, seperti *server web, server game, server chatting* dan sebagainya.[6]

4) MySQL

MySQL adalah salah satu sistem manajemen basis data (*DBMS*) open-source yang sangat terkenal. *DBMS* sendiri merupakan perangkat lunak yang berfungsi untuk menyimpan, mengelola, dan mengakses data yang ada di dalam basis data. *MySQL* biasanya digunakan untuk menyimpan data terstruktur, seperti data dalam bentuk tabel yang saling berhubungan.[7]

MySQL adalah server database *open source* yang sangat populer. Berkat berbagai keunggulannya, *software* ini sering digunakan oleh para profesional dalam mengembangkan proyek. *MySQL* menyediakan fasilitas *API* (*Application Programming Interface*), yang memungkinkan berbagai aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk mengakses basis data *MySQL* dengan mudah.[8]

Sistem ini menggunakan bahasa pemrograman *SQL (Structured Query Language)* sebagai bahasa standar untuk mengelola dan memanipulasi data yang disimpan dalam basis data. Dalam konteks penelitian akademis, seperti penulisan skripsi, *MySQL* sering digunakan untuk menyimpan, mengelola, dan menganalisis data penelitian dengan efisien.

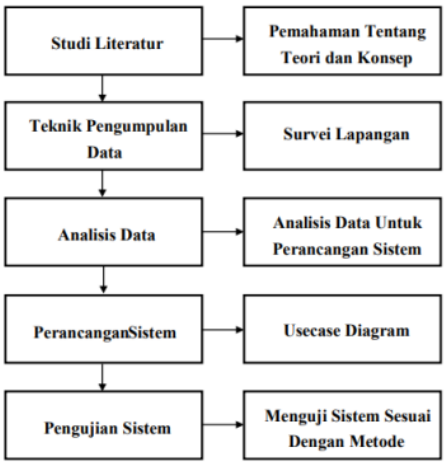
5) Quality of Service

Quality of Service (QoS) adalah metode untuk mengukur seberapa baik kinerja sebuah jaringan dan usaha untuk mendefinisikan karakteristik serta sifat dari suatu layanan. *QoS* digunakan untuk menilai berbagai atribut kinerja yang telah ditentukan dan dikaitkan dengan layanan tersebut.[9]

II. METODOLOGI PENELITIAN

A. Metode Penelitian

Pengumpulan data diperlukan tahapan-tahapan yang akan dilakukan oleh peneliti sehingga dapat menyelesaikan masalah yang akan diteliti. Gambar 1 menjelaskan tahapan dalam proses penelitian.



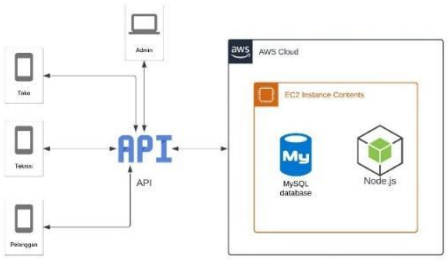
Gambar 1. Kerangka Penelitian

Berdasarkan tahapan penelitian sesuai gambar 1 dapat dijelaskan sebagai berikut:

1) Studi Literatur, pada tahapan penelitian ini dilakukan penelusuran pada berbagai literatur seperti buku, jurnal ilmiah, dan referensi lainnya yang berhubungan dengan judul penelitian ini.
2) Teknik Pengumpulan Data, pada tahapan penelitian ini akan dilakukan pengumpulan data yang diperoleh melalui survei dan observasi langsung ke lapangan yaitu Mahasiswa dan Dosen Pembimbing dan diperoleh dari hasil pengamatan dan wawancara dari bimbingan skripsi.
3) Analisis Data, pada penelitian ini akan dilakukan analisis data yang sesuai untuk merancang sistem berdasarkan parameter yang telah ditentukan.
4) Perancangan Sistem, pada penelitian ini akan dilakukan perancangan dari aplikasi *cloud computing* yaitu *use case* dan *activity diagram* serta perancangan *user interface* dan blok diagram sistem aplikasi.
5) Pengujian Sistem, pada tahapan penelitian ini dilakukan pengujian sistem yang telah dibuat berdasarkan metode *stess testing* yang digunakan untuk menguji kelayakan pada sistem yang dibuat dan pengujian kepuasan aplikasi.

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. *Block Diagram*

Gambar 2 menjelaskan bahwa dalam lingkungan *cloud computing* menggunakan *Amazon Web Services (AWS),* sistem menerima dan mengelola data dari berbagai pengguna melalui antarmuka pemrograman aplikasi. Pengguna terdiri dari beberapa peran yaitu Toko, Teknisi, Pelanggan, dan *Admin* yang berinteraksi dengan *API* untuk mengirim dan menerima data.

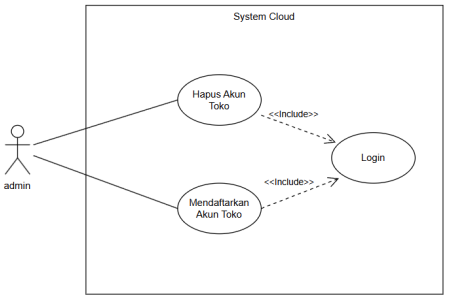
Sistem *cloud* ini menggunakan layanan *AWS* yang mencakup *instance EC2* sebagai *server* utama yang menjalankan aplikasi berbasis *Node.js* dan menggunakan *MySQL* sebagai *database* untuk penyimpanan data. Data yang dikirimkan oleh pengguna disimpan di dalam *database MySQL*, dan aplikasi *Node.js* yang berjalan pada *instance EC2* bertanggung jawab untuk memproses dan mengelola data tersebut.

API berfungsi sebagai penghubung utama antara pengguna dan sistem *backend*, memastikan bahwa setiap permintaan dari pengguna diproses dengan benar dan data yang relevan disinkronkan kembali ke pengguna yang membutuhkan informasi tersebut. Proses ini memastikan bahwa setiap pembaruan data yang dilakukan oleh satu pengguna dapat diakses oleh pengguna lain secara *real-time*, sehingga menjaga konsistensi dan integritas data di seluruh sistem.

Dengan menggunakan arsitektur berbasis *cloud* ini, sistem dapat menangani banyak pengguna secara efisien, menyediakan akses yang andal dan cepat ke data yang dibutuhkan oleh berbagai peran pengguna, serta memastikan bahwa layanan tetap berjalan dengan kinerja optimal melalui infrastruktur *AWS*.

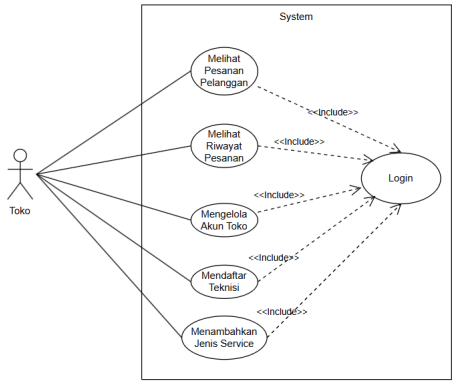
C. *Diagram Use Case*

Aplikasi ini memiliki interaksi yang sangat kompleks. Diagram use case digunakan untuk menggambarkan suatu sistem secara umum agar mudah dipahami. Diagram use case ditunjukkan pada gambar berikut.



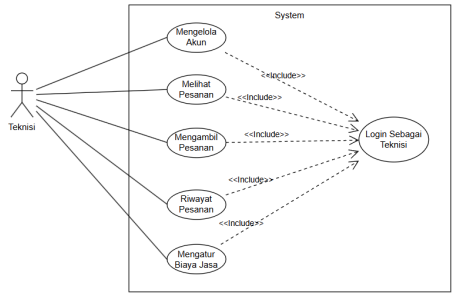
Gambar 3. *Diagram Use Case Admin*

Berdasarkan gambar 3 Diagram *use case* ini menggambarkan interaksi antara *admin* dengan sistem *cloud* dalam aplikasi layanan. *Admin* memiliki kemampuan untuk mengelola akun toko, yang melibatkan dua *use case* utama: "Hapus Akun Toko" dan "Mendaftarkan Akun Toko".



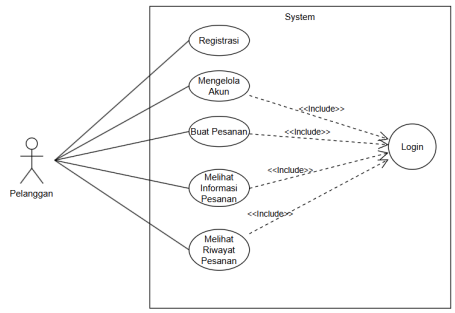
Gambar 4. *Diagram Use Case* Toko

Berdasarkan gambar 4 menjelaskan tentang alur sistem yang akan dibuat, Diagram *use case* ini menggambarkan interaksi antara Toko dengan sistem dalam konteks aplikasi layanan. Toko harus melakukan *login* terlebih dahulu ke dalam sistem menggunakan *email* dan *password* yang sesuai untuk mengakses fitur lainnya.



Gambar 5. *Diagram Use Case* Teknisi

Pada gambar 5 *Diagram use case* ini menggambarkan interaksi antara Teknisi dengan sistem dalam aplikasi. Teknisi harus melakukan *login* terlebih dahulu ke dalam sistem menggunakan *email* dan *password* yang sesuai untuk mengakses fitur. Setelah *login*, teknisi dapat mengelola akun mereka, termasuk memperbarui informasi pribadi.



Gambar 6. *Diagram Use Case* Pelanggan

Pada gambar 6 *Diagram use case* ini menggambarkan interaksi antara Pelanggan dengan sistem dalam aplikasi. Pelanggan dapat melakukan beberapa aktivitas melalui sistem, dan setiap aktivitas diilustrasikan sebagai *use case*.

Pelanggan dapat melakukan *registrasi* untuk membuat akun baru dalam sistem. Setelah berhasil registrasi, pelanggan harus *login* menggunakan *email* dan *password* yang sudah dibuat untuk mengakses fitur-fitur lainnya.

III. Hasil dan Pembahasan

A. Rancangan User Interface

Interface adalah sebuah tampilan pada suatu sistem yang memungkinkan pengguna untuk berinteraksi dengan perangkatnya secara langsung atau melalui jaringan Rangkaian Perangkat.[10]

Interface adalah suatu bagian yang berhubungan langsung dengan pengguna aplikasi. Desain antar muka ini, didesain berdasarkan keperluan dalam membangun aplikasi dan bertujuan untuk menghasilkan aplikasi agar kelihatan lebih menarik dan mudah dalam penggunaannya.[11]

6) User Interface Admin

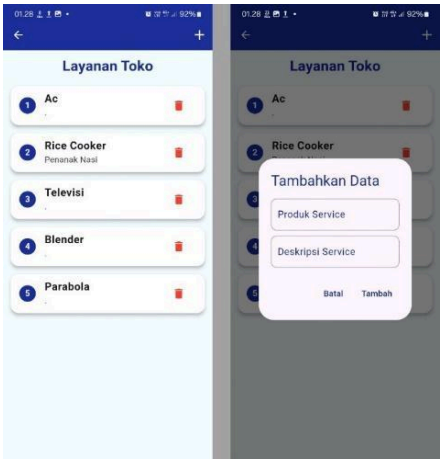
Halaman *dashboard* dalam sistem layanan pemesanan teknisi dirancang untuk memudahkan *admin* dalam mengelola dan memantau data yang terkait dengan toko-toko yang terdaftar.

Gambar 7. *Dashboard Admin*

Pada gambar 7, halaman ini dirancang untuk memberikan tampilan daftar toko yang telah terdaftar dalam sistem layanan pemesanan teknisi. Di bagian atas halaman, terdapat tombol "Tambah Toko" yang memungkinkan *admin* untuk menambahkan toko baru ke dalam daftar. Selain itu, terdapat kolom pencarian di sebelah kanan untuk memudahkan admin dalam mencari toko berdasarkan nama.

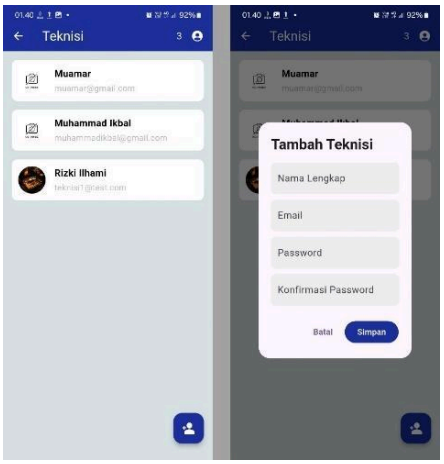
7) User Interface Toko

Halaman Pesanan pada aplikasi ini dirancang khusus untuk dilihat oleh pihak toko, yang memungkinkan mereka untuk memantau progres pesanan yang sedang dikerjakan oleh teknisi.



Gambar 8. Halaman Layanan Toko

Pada gambar 8, halaman "Layanan Toko" ini dirancang untuk memungkinkan toko menambah atau mengelola jenis layanan yang mereka tawarkan. Pada halaman ini, terdapat daftar layanan yang sudah ada beserta deskripsinya. Setiap layanan memiliki tombol hapus yang memungkinkan toko untuk menghapus layanan yang tidak diperlukan lagi.

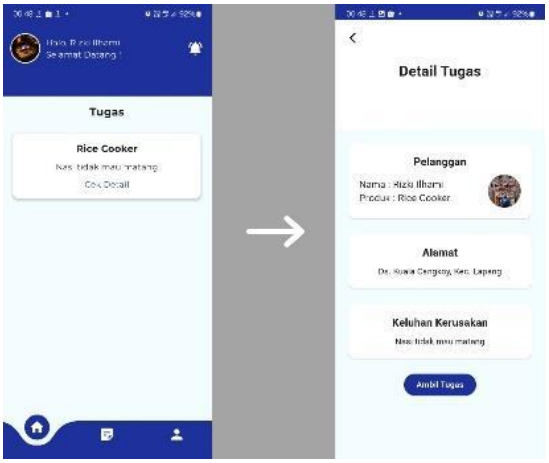


Gambar 9. Halaman Daftar Teknisi

Pada gambar 9, halaman "Teknisi" ini dirancang untuk memungkinkan toko mengelola akun teknisi yang bekerja untuk mereka. Pada halaman ini, terdapat daftar teknisi yang sudah terdaftar beserta informasi kontak mereka. Setiap teknisi ditampilkan dengan nama, email, dan gambar profil jika tersedia. Teknisi yang sudah terdaftar di antaranya adalah Muamar, Muhammad Ikbal dan Rizki Ilhami.

8) User Interface Teknisi

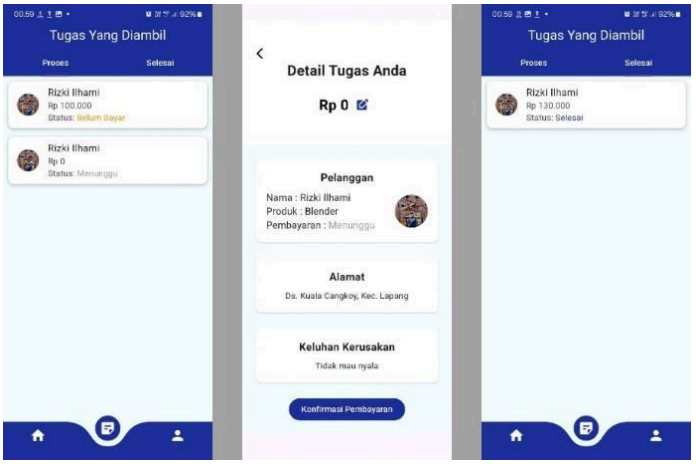
Halaman Beranda pada aplikasi teknisi dirancang untuk memudahkan teknisi dalam melihat dan mengambil tugas perbaikan yang tersedia.



Gambar 10. Halaman Pekerjaan dari Pelanggan

Pada gambar 10, halaman ini dirancang untuk teknisi agar dapat melihat dan mengambil tugas perbaikan yang tersedia. Pada tampilan awal, teknisi akan disambut dengan pesan selamat datang yang disertai nama teknisi. Di bawah pesan selamat datang, terdapat daftar tugas yang perlu ditangani.

Contoh tugas yang ditampilkan adalah perbaikan "Rice Cooker" dengan keluhan "Nasi tidak mau matang". Terdapat tombol "Cek Detail" yang memungkinkan teknisi untuk melihat informasi lebih lanjut mengenai tugas tersebut.

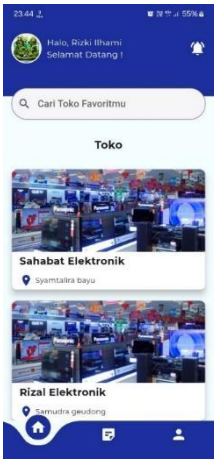


Gambar 11. Halaman Tugas Yang Diambil Teknisi

Pada gambar 11, halaman ini dirancang untuk teknisi agar dapat memantau status tugas yang telah mereka ambil, yang terbagi dalam dua tab utama: "Proses" dan "Selesai". Pada tab "Proses", teknisi dapat melihat daftar tugas yang sedang dikerjakan. Contoh tugas yang ditampilkan adalah perbaikan "Blender" dengan status "Menunggu" dan "AC" dengan status "Belum Bayar". Untuk tugas dengan status "Belum Bayar", terdapat informasi biaya sebesar Rp 100.000 dan status pembayaran yang menunggu konfirmasi dari pelanggan.

9) User Interface Pengguna

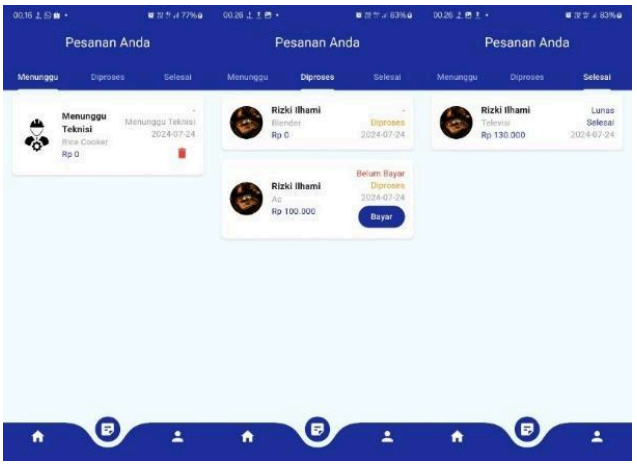
Halaman ini dirancang untuk memberikan pengalaman yang mudah dan intuitif bagi pengguna.



Gambar 12. Halaman Utama Pelanggan

Pada gambar 12, halaman *dashboard* pelanggan ini dirancang untuk memberikan pengalaman yang mudah dan intuitif bagi pelanggan dalam memilih toko untuk membuat pesanan.

Dengan *dashboard* ini, pelanggan dapat dengan cepat dan mudah memilih toko favorit mereka dan membuat pesanan sesuai kebutuhan mereka.



Gambar 13. Halaman *list* pesanan yang telah dibuat

Pada gambar 13, halaman ini dirancang untuk memungkinkan pelanggan memantau status pesanan mereka yang terbagi dalam tiga tab utama: Menunggu, Diproses, dan Selesai. Pada tab "Menunggu", pelanggan dapat melihat pesanan yang telah dibuat dan sedang menunggu teknisi untuk diambil. Contoh pesanan di tab ini adalah untuk "*Rice Cooker*" dengan status "Menunggu Teknisi". Di sini, pelanggan juga memiliki opsi untuk membatalkan pesanan.

B. Pengujian

1) Pengujian Stress Testing Menggunakan Jmeter

Dari seluruh rangakain pengujian stress yang telah dilakukan dengan berbagai variasi beban didapatkan data pengujian seperti pada tabel 4.8.

TABEL I
HASIL PENGUJIAN *STRESS TESTING*

Beban	Average	Error	Troughput
1000	2612	0,80%	148,9/s
2000	4534	4,43%	11,5/s
3000	7296	9,80%	158/s
4000	9118	11,80%	167,6/s
5000	15385	57,40%	224,6/s
6000	16112	43,73%	18,7/s
7000	16660	50,67%	17.1/s

Dari hasil pengujian ini, terlihat bahwa peningkatan beban permintaan secara signifikan mempengaruhi stabilitas dan efisiensi *server*, dengan meningkatnya tingkat *error* dan waktu *respons*, serta variasi pada *throughput*. Hal ini menunjukkan bahwa server memerlukan optimasi lebih lanjut untuk dapat menangani beban yang lebih tinggi dengan lebih efisien.

2) Pengujian Kualitas Respons Realtime dalam Pengiriman Tugas Pelanggan Menggunakan Quality of Service

Penilaian kualitas fitur pengiriman tugas secara *real-time* dari pelanggan dalam aplikasi dilakukan dengan menerapkan metode *Quality of Service* (*QoS*) 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.

Untuk menguji performa aplikasi, dilakukan simulasi dengan mengirim data berupa pembuatan pesanan menggunakan berbagai jumlah permintaan, yaitu 10, 20, 50, 100, dan 200. 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.

TABEL II
HASIL PENGUJIAN 10 PERMINTAAN

Parameter *QoS*	Hasil Perhitungan	kategori
Throughput	$= \frac{paket\ diterima}{waktu\ pengamatan} = \frac{7.720}{1143} = 6,75\ bits/s\ x\ 8$	Bagus

Packet Loss	$= \frac{paket\ dikirim - paket\ diterima}{paket\ dikirim}\ x\ 100\% = \frac{50-50}{50} x$	Sangat Bagus
Delay	$= \frac{total\ delay}{paket\ diterima} = \frac{1,142819}{49} = 0,023\ s\ x\ 1000 =$	Sangat Bagus
Jitter	$= \frac{total\ variasi\ delay}{paket\ diterima-1} = \frac{1,097662}{48} = 0,023\ s\ x\ 100$	Bagus

Hasil pengujian pada tabel II menunjukkan bahwa nilai *throughput* mencapai 54,003 permintaan per detik, dan *packet loss* mencapai 0%. *delay* yang tercatat adalah 23,323 *ms*. *jitter* yang tercatat adalah 22,867 *ms*.

TABEL III
HASIL PENGUJIAN 50 PERMINTAAN

Parameter *QoS*	Hasil Perhitungan	kategori
Throughput	$= \frac{paket\ diterima}{waktu\ pengamatan} = \frac{38.600}{1.187} = 32,519\ bits$	Sangat Bagus
Packet Loss	$= \frac{paket\ dikirim - paket\ diterima}{paket\ dikirim}\ x\ 100\% = \frac{250-2}{250}$	Sangat Bagus
Delay	$= \frac{total\ delay}{paket\ diterima} = \frac{1,187}{249} = 0,005\ s\ x\ 1000 =$	Sangat Bagus
Jitter	$= \frac{total\ variasi\ delay}{paket\ diterima-1} = \frac{1,156}{248} = 0,005\ s\ x\ 100$	Bagus

Hasil pengujian pada tabel III menunjukkan bahwa nilai *throughput* mencapai 260,152 permintaan per detik, dan *packet loss* mencapai 0%. *delay* yang tercatat adalah 4,766 *ms*. *jitter* yang tercatat adalah 4,663 *ms*.

TABEL IV
HASIL PENGUJIAN 100 PERMINTAAN

Parameter QoS	Hasil Perhitungan	kategori
Throughput	$= \frac{paket\ diterima}{waktu\ pengamatan} = \frac{76.660}{1.208} = 63,46\ bits/s\ x$	Sangat Bagus
Packet Loss	$= \frac{paket\ dikirim - paket\ diterima}{paket\ dikirim}\ x\ 100\% = \frac{490-490}{490}$	Sangat Bagus
Delay	$= \frac{total\ delay}{paket\ diterima} = \frac{1,208312}{489} = 0,002\ s\ x\ 1000 =$	Sangat Bagus
Jitter	$= \frac{total\ variasi\ delay}{paket\ diterima-1} = \frac{1,186}{488} = 0,002\ s\ x\ 1000 =$	Bagus

Hasil pengujian pada tabel IV menunjukkan bahwa nilai *throughput* mencapai 507,682 permintaan per detik, dan *packet loss* mencapai 0%. *delay* yang tercatat adalah 2,471 ms. jitter yang tercatat adalah 2,430 ms.

TABEL V
HASIL PENGUJIAN 200 PERMINTAAN

Parameter QoS	Hasil Perhitungan	kategori
Throughput	$= \frac{paket\ diterima}{waktu\ pengamatan} = \frac{151.958}{1.366} = 111,243\ bits/$	Sangat Bagus
Packet Loss	$= \frac{paket\ dikirim - paket\ diterima}{paket\ dikirim}\ x\ 100\% = \frac{926-916}{926}$	Sangat Bagus
Delay	$= \frac{total\ delay}{paket\ diterima} = \frac{1,366}{925} = 0,001\ s\ x\ 1000$ $= 1,477\ ms$	Sangat Bagus

Jitter	$= \frac{total\ variasi\ delay}{paket\ diterima-1} = \frac{1,356}{924} = 0,001\ s\ x\ 1000 =$	Bagus

Hasil pengujian pada tabel V menunjukkan bahwa nilai *throughput* mencapai 889,944 permintaan per detik, dan *packet loss* mencapai 1,08%. *delay* yang tercatat adalah 1,477 ms. jitter yang tercatat adalah 1,468 ms.

IV. KESIMPULAN

Berdasarkan hasil pengujian *Quality of Service (QoS)* yang dilakukan, jaringan menunjukkan performa yang sesuai dengan standar TIPHON dalam menangani beban tinggi. *Throughput* mengalami peningkatan seiring dengan peningkatan jumlah permintaan, dengan rata-rata *throughput* mencapai 364,282 *kbps*. Meskipun *packet loss* tetap rendah hingga jumlah permintaan mencapai 100, terjadi peningkatan menjadi 1,08% pada 200 permintaan, yang mendekati ambang batas maksimum yang diizinkan oleh standar TIPHON, yaitu 1%. Ini menunjukkan bahwa jaringan mulai mengalami batasan kapasitas saat beban mendekati tingkat yang sangat tinggi. Selain itu, *delay* dan *jitter* juga tetap berada dalam batas yang wajar sesuai standar TIPHON, yang mencerminkan stabilitas jaringan dalam pengiriman data pada kondisi uji ini.

Infrastruktur *cloud* yang menggunakan layanan *AWS* dengan *instance t2.micro* saat ini mampu menangani beban permintaan dengan performa yang cukup memadai. Namun, terdapat batasan pada kemampuan penanganan beban yang sangat tinggi, yang terlihat dari peningkatan *error* dan penurunan *throughput* selama pengujian *stress* testing. Ini mengindikasikan bahwa sistem *cloud AWS* dengan *instance t2.micro* memerlukan optimasi lebih lanjut untuk memastikan performa tetap stabil di bawah kondisi beban puncak.

V. REFERENSI

[1] Asri Nanda, Hari Toha Hidayat, and Mahlil, "Implementasi Cloud Computing Untuk Media Pembelajaran Interaktif Bahasa Inggris Berbasis Android," vol. 3, no. 2, pp. 44–49, 2023.

[2] Matheus Supriyanto Rumetna, "Pemanfaatan Cloud Computing Pada Dunia Bisnis: Studi Literatur," vol. 5, no. 3, pp. 305–314, Aug. 2018.

[3] Dinda Lusita Fristiani Anissa and Ria Andryani, "Penerapan Cloud Computing Dalam Aplikasi Panggil Teknisi Berbasis Android Menggunakan Google Cloud Platform," vol. 6, no. 2, pp. 1290–1300, Sep. 2022.

[4] Muhammad Mustafa, Anwar, and Indrawati, "Prototipe Sinkronisasi Penyimpanan Dokumen Secara Realtime Menggunakan Websocket Pada Cloud Storage," vol. 7, no. 1, pp. 93–99, 2023.

[5] Mus Mulyadi Usman, Xaverius B.N. Najoan, and Meicsy E. I. Najoan, "Rancang Bangun Aplikasi MonitoringKetinggian Air Sungai Berbasis Internet of ThingsMenggunakan Amazon Web Service," vol. 9, no. 2, pp. 73–80, 2020.

[6] Gusti Putu Mahendra Putra, Andi Tenriawaru, and Gunawan, "Rancang Bangun Virtual Assistant Chatbot Menggunakan Node.Js pada Layanan Sistem Informasi Akademik," vol. 1, no. 1, pp. 345–352, 2023.

[7] 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," vol. 3, no. 1, pp. 25–29, 2023.

[8] Muhammad Syauqi, Huzaeni, and Mahdi, "Rancang Bangun Aplikasi Agenda Berbasis Android Dengan Fitur Push Notification dan Reminder (Studi Kasus : Dewan Perwakalian Rakyat Aceh)," vol. 3, no. 1, pp. 36–42, 2023.

[9] Rika Wulandari, "Analisis Qos (Quality of Service) Pada Jaringan Internet (Studi Kasus : Upt Loka Uji Teknik Penambangan Jampang Kulon – Lipi)," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 2, no. 2, Aug. 2016.

[10] Risma Arifa, Husaini, and Fachri Yannuar Rudi F, "Implementasi Aplikasi Enkripsi Dekripsi File Dokumen Menggunakan Algoritma Aes Pada Cloud Computing (Studi Kasus Prodi Teknologi Rekayasa Komputer Jaringan)," *Journal of Artificial Intelligence and Software Engineering*, vol. 4, no. 1, pp. 8–12, 2024.

[11] Andri Irwan Zahri, Hari Toha Hidayat, and Aswandi, "Rancang Bangun E-Commerce Menggunakan QR-Code pada Perusahaan Retail Berbasis Android," vol. 1, no. 2, pp. 12–20, Dec. 2021.