

Sistem Berbagi Lokasi Real-Time dengan Teknologi Websocket Menggunakan Google Maps API

Muhammad Diyauddin Maulidi¹, Deni Sutaji²

¹ Teknik Informatika, Universitas Muhammadiyah Gresik

² Computer Science, Gazi University

e-mail: mdiyauddin22@gmail.com¹, sutaji.deni@umg.ac.id²

Abstrak

Pada saat ini, fitur berbagi lokasi sangat penting dalam kehidupan sehari-hari. Fitur ini biasanya hanya digunakan untuk mengetahui lokasi pengguna. Namun, dengan kemajuan teknologi saat ini dibutuhkan fitur yang lebih multifungsi, seperti interaksi dengan aplikasi, berbagi ruangan, dan melihat hasil berbagi lokasi. Untuk mengatasi hal ini, dikembangkan sistem berbagi lokasi menggunakan websocket yang memungkinkan pengguna untuk mengakses berbagai fitur dengan mudah. Metode penelitian mencakup enam tahap, yaitu: mulai, studi literatur, rekayasa kebutuhan, perancangan, implementasi, dan pengujian. Hasil menunjukkan bahwa sistem ini berhasil memenuhi spesifikasi yang ditetapkan dan dapat dikembangkan lebih lanjut.

Kata Kunci : *Berbagi Lokasi, Websocket, Teknologi*

Abstract

At this time, location-sharing features are very important in daily life. This feature is usually used only to determine the user's location. However, with today's technological advancements, there is a need for more multifunctional features, such as interaction with applications, room sharing, and viewing the results of location sharing. To address this, a location-sharing system has been developed using websockets, allowing users to easily access various features. The research method includes six stages: initiation, literature review, requirements engineering, design, implementation, and testing. The results indicate that this system successfully meets the established specifications and can be further developed.

Keywords : *Location Sharing, Websocket, Technology*

PENDAHULUAN

Pada saat ini, *Website* menjadi sarana komunikasi satu arah yang cukup efisien. *Website* bisa dikatakan hampir tidak lepas dari keseharian seseorang. Berkat *Website*, sekarang kita bisa dapat melakukan pekerjaan dengan jauh lebih mudah jika dibandingkan dengan tahun-tahun lalu (Maharani, Helmiah, & Rahmadani, 2021). Dalam dunia web, AJAX sangat berperan penting dalam meningkatkan interaktivitas, kecepatan, dan daya tanggap situs. Tetapi penggunaan sumber daya yang berlebihan di sisi server dapat mengakibatkan penurunan kinerja sistem jika AJAX digunakan dengan frekuensi permintaan yang tinggi. Tingkat peminatan AJAX tinggi pada sistem berbagi lokasi. Tetapi akan ada penurunan kinerja karena penggunaan sumber daya AJAX. Dengan memanfaatkan teknologi AJAX, website yang dibuat akan berfungsi layaknya aplikasi desktop (Permadi & Prasida, 2022).

Disisi lain, adapun *Google maps* yang juga masih menjadi layanan untuk mempermudah seseorang dalam melakukan pencarian lokasi. *Google maps* adalah aplikasi berbagi lokasi berbasis layanan Web yang dikembangkan oleh Google. *Google maps* juga bisa dikatakan sebagai layanan yang menampilkan peta berbagai belahan dunia. Google maps memiliki beberapa fitur meliputi kemampuan untuk memanipulasi peta, menambahkan berbagai layanan, serta merancang solusi enterprise di situs webnya. Dengan berbagi lokasi, seseorang bisa memberikan info lokasi terkini nya untuk hal-hal tertentu. Google maps API adalah pengembangan teknologi dari google yang memungkinkan integrasi google maps ke dalam aplikasi yang tidak

dikembangkan oleh google (Karsana & Mahendra, 2021). *Handphone* memudahkan kita untuk dapat mengukur posisi untuk menemukan informasi seseorang berdasarkan letak lokasi nya (Pranatawijaya, 2021).

Pada umumnya, berbagi lokasi hanya bisa digunakan untuk mengetahui lokasi pengguna yang membagikannya saja. Di zaman yang sekarang canggih ini, kita memerlukan fitur yang tidak hanya bisa digunakan untuk satu fungsi saja. Kemajuan teknologi memungkinkan kita untuk dengan mudah terhubung dengan orang lain, termasuk mengetahui lokasi mereka secara real-time (Samantha & Anggalih, 2024). Selain berbagi lokasi, berinteraksi dengan aplikasi, berbagi ruangan, dan melihat hasil dari berbagi lokasi itu juga diperlukan agar kita tidak perlu kesulitan dalam menggunakan sosial media. Perlunya pengembangan fitur berbagi lokasi untuk memudahkan seseorang dalam memakai sebuah aplikasi.

Websocket adalah sebuah teknologi yang bisa dikatakan memiliki performa terbaik jika dibandingkan dengan beberapa teknologi lainnya, karena websocket memiliki tingkat permintaan sistem atau rate-request yang tinggi. Protokol komunikasi client-server yang awalnya dirancang untuk web browser kini dapat digunakan di berbagai jenis client atau sever (Alviando, Bhawiyuga, & Kartikasari, 2023). Pada umumnya, *websocket* dirancang dan bisa digunakan pada *web browser*. Dengan menggunakan *websocket*, *client* langsung mendapatkan data dari *server* tanpa harus melakukan *request* data (Maulana, 2021). Metode *long polling* ialah metode yang muncul sebelum *websocket* update kemampuan secara *realtime*. Adapun diketahui kelebihan *websocket* ialah bahwa *websocket* memiliki sistem yang lebih cepat dari HTTP. Penggantian HTTP menggunakan TCP juga menjadi salah satu kelebihan dari teknologi *websocket*.

Maka dari itu dapat disimpulkan untuk mengembangkan fitur berbagi lokasi pada penelitian ini, kami menggunakan sebuah teknologi komunikasi berbasis yaitu teknologi *Websocket*. Dalam melakukan komunikasi secara *realtime*, standar bagus menggunakan *websocket* karena dasarnya, tanpa melakukan *request* data dahulu, *server* dapat memasukkan data pada *client* (Maulana, 2021). Tujuannya adalah agar seseorang yang memakai fitur ini dapat dengan mudah untuk berinteraksi dengan aplikasi, berbagi ruangan, dan melihat hasil dari berbagi lokasi nya. Perbedaan fitur berbagi lokasi secara *realtime* dengan fitur berbagi lokasi sebelumnya hanya dibagian bisa memberikan update lokasi terkini sesuai pergerakan kita. Pada fitur secara *realtime*, kita juga dapat menentukan berapa lama seseorang bisa melacak keberadaan dengan menentukan waktunya sendiri. Penelitian ini akan berkonsentrasi pada kebutuhan perangkat lunak yang diperlukan untuk membuat sistem berbagi lokasi berbasis web dengan diawali dengan merancang sistem kebutuhan perangkat lalu menguji sistem untuk menguji kualitasnya.

Ajax

AJAX (*Asynchronous JavaScript and XML*) memiliki tujuan yaitu melakukan transmisi data dengan *server* yang terletak di seberang perbatasan, yang dapat diartikan AJAX tidak perlu dibaca secara lengkap setiap kali kita melakukan perubahan. Dengan menggunakan sistem ini, dapat memberikan manfaat untuk mengurangi waktu yang diperlukan untuk memuat data. Adapun manfaat lain AJAX bagi diri kita ialah AJAX memungkinkan aplikasi untuk merespons tindakan pengguna secara *realtime*. Pengurangan beban *server* juga merupakan manfaat untuk diri kita karena hanya data yang diperlukan yang dapat dikirim dan diterima.

Websocket Server

Websocket Server adalah komponen kunci dalam pengembangan aplikasi web *realtime* yang memungkinkan komunikasi dua arah antara *client* dan *server*. Pentingnya aplikasi web *realtime* semakin meningkat, dan *Websocket Server* memainkan peran krusial dalam mendukung komunikasi efisien antara *client* dan *server*. Prinsip kerja *Websocket Server* melibatkan serangkaian langkah yang memungkinkan koneksi persisten antara klien dan *server* serta pertukaran data *realtime*. Dengan persistensi koneksi dan kemampuan dua arahnya, *Websocket Server* membuka pintu untuk pengembangan aplikasi web yang lebih responsif dan interaktif.

Websocket Client

Websocket Client telah menjadi elemen kunci dalam pengembangan aplikasi web yang memanfaatkan secara *realtime*. Dalam konteks berbagi lokasi, responsivitas dan akurasi informasi menjadi kritis. *Websocket Client* memungkinkan pengembang untuk mengembangkan sistem berbagi lokasi, menciptakan pengalaman pemantauan yang lebih baik.

Google Maps API

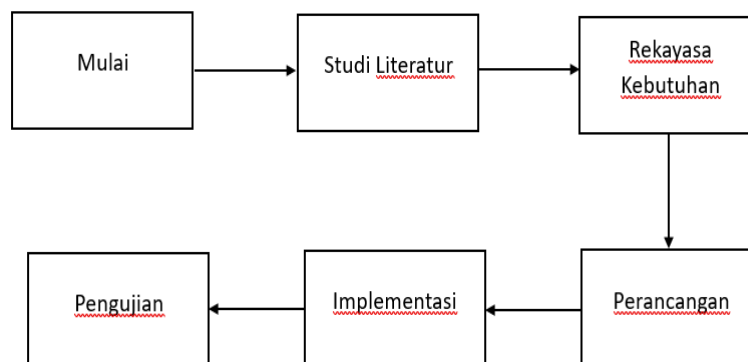
Google maps ialah sebuah aplikasi peta online yang sudah disediakan oleh *Google* dan dapat diakses secara gratis. *Google maps API* adalah layanan dari *google* yang memungkinkan pengguna untuk menyisipkan peta ke dalam situs web dengan memanfaatkan *JavaScript* (Hamdani & Utomo, 2021). *Google maps API* adalah sekumpulan layanan berbasis peta yang disediakan oleh *Google* untuk membantu pengembang dalam mengintegrasikan fitur pemetaan ke dalam aplikasi mereka. API ini memungkinkan pengguna untuk menampilkan peta interaktif, menambahkan penanda lokasi, menampilkan informasi lalu lintas, serta memberikan navigasi secara real-time. Dengan menggunakan API, pengguna dapat menampilkan peta secara interaktif, menambahkan penanda lokasi, dan memberikan informasi rute kepada pengguna.

Location Based Service (LBS)

Layanan Berbasis Lokasi atau yang biasa dikenal oleh masyarakat dengan sebutan *Location Based Service* (LBS) ialah sebuah teknologi yang menampilkan gambar sebuah lokasi perangkat yang sedang kita gunakan. Dengan menggunakan sebuah perangkat yang dilengkapi dengan GPS, *Location Based Service* (LBS) dapat menawarkan berbagai aplikasi (seperti tempat makan, belanja, dll). Aplikasi navigasi seringkali juga menggunakan layanan ini (Yuwamahendra & Ratnasari, 2020). Beberapa aplikasi seluler berbasis lokasi yang populer meliputi: *Google maps*, *Instagram*, *WhatsApp*, *Waze*, dll. *Location Based Service* (LBS) dapat membantu pengguna untuk menjelajahi dunia di sekitar pengguna dengan cara yang efisien.

METODE

Adapun metode pada penelitian ini mencakup 6 langkah-langkah yang diantaranya ialah:



Gambar 1. Metode Penelitian

Studi Literatur

Tahap ini ialah melakukan pencarian dan analisis literatur serta referensi yang memiliki relevansi dengan topik penelitian. Tujuan utamanya adalah untuk memperoleh pemahaman mendalam terkait dengan tema penelitian yang sedang dijalankan. Selain itu, penelitian literatur sebelumnya juga dilakukan guna menyelidiki teori terkait yang dapat menjadi landasan untuk mengembangkan penelitian ini. Adapun beberapa yang menjadi fokus dalam penelitian yang kami bahas ialah teknologi komunikasi menggunakan *Websocket*, Sistem yang berbasis lokasi, serta berorientasikan pada objek.

Rekayasa Kebutuhan

Proses ini melibatkan penerapan metode analisis berorientasi objek untuk menangani rekayasa yang diperlukan dalam konteks penelitian ini. Dilakukan penentuan, definisi, dan pemodelan kebutuhan sistem pada tahap ini. Spesifikasi kebutuhan diperiksa secara menyeluruh dalam menguji validitas sistem dikarenakan untuk menguji kembali bahwa sistem ini memenuhi beberapa parameter validitas, yaitu keberlakuan, konsistensi, kelengkapan, realisme, dan

verifikasi. Tujuannya untuk memastikan bahwa setiap elemen kebutuhan sistem dapat diandalkan dan sesuai dengan kebutuhan yang sebenarnya.

Perancangan

Pada tahap perancangan sistem, dilakukan dari beberapa perancangan diantaranya ialah arsitektur, basis data, dan antarmuka pengguna. Sebuah ilustrasi antar class yang ditampilkan pada sistem bisa disebut *class diagram* (Hidayah , Agustina, Indirawati, & Amalia, 2021). Pada perancangan arsitektur sistem menghasilkan sebuah *class diagram*, sedangkan dihasilkan juga tabel-tabel yang merupakan hasil dari perancangan basis data. Dan terakhir model tampilan antarmuka pengguna dan komponen-komponen antarmuka merupakan hasil dari sistem perancangan antarmuka pengguna.

Implementasi

Dalam tahap implementasi sistem, fokus utama pada pengembangan sistem *client* dan *server*. Pada sisi *server*, proses implementasi melibatkan *JavaScript* berorientasi objek dalam lingkungan runtime *Node.JS*. *Server* ini dibangun dengan menggunakan non-relational database *MongoDB*, dimana *framework Express.JS* dan *Mongoose.JS* turut dimanfaatkan.

Pengujian

Dalam penelitian ini, ada satu jenis evaluasi dilaksanakan, yakni evaluasi terhadap kebutuhan sistem. Adapun tahap-tahap pada evaluasi kebutuhan sistem yaitu, tingkat unit, integrasi unit, dan pada istem. Proses evaluasi unit dan integrasi unit memiliki pelaksanaan yang sama yakni dilakukan dengan pendekatan white box menggunakan basis path testing metode. Dengan menggunakan metode ini, dapat diketahui cakupan seluruh test yang akan menampilkan hasil jumlah test case secara lebih luas (Sie, Musdar, & Bahri, 2022). Sementara itu, pada tingkat sistem, evaluasi dilakukan dengan pendekatan *black-box* menggunakan *requirement testing*.

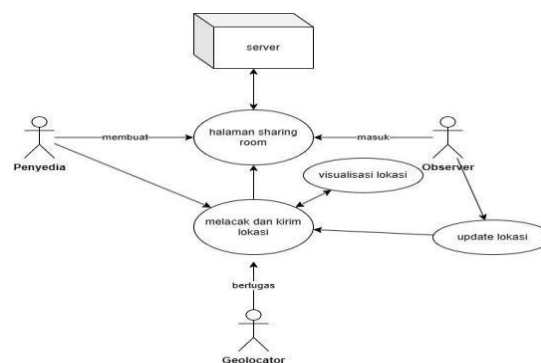
HASIL DAN PEMBAHASAN

Rekayasa Kebutuhan

Pada tahap rekayasa kebutuhan, terdapat beberapa aspek penting yang harus diperhatikan. Seperti arsitektur, basis data, dan antarmuka pengguna. Sebuah ilustrasi antar class yang ditampilkan pada sistem bisa disebut *class diagram* (Hidayah , Agustina, Indirawati, & Amalia, 2021). Proses perancangan arsitektur sistem menghasilkan *class diagram* yang memberikan Gambaran jelas mengenai struktur sistem yang akan dibangun.

Selain itu, pembahasan yang akan disajikan di bawah ini merupakan hasil dari perancangan basis data yang mendukung pengelolaan data dalam sistem. Model tampilan antarmuka pengguna dan komponen nya juga merupakan hasil dari proses desain antarmuka pengguna yang bertujuan untuk menciptakan pengalaman pengguna yang efisien.

Dibawah ini merupakan diagram use case yang menampilkan antara pengguna dan sistem untuk memahami bagaimana fungsionalitas sistem diimplementasikan dalam skenario penggunaan yang spesifik.



Gambar 2. Use case diagram

Setelah menampilkan *use case diagram* pada Gambar 1, langkah selanjutnya adalah menjelaskan secara mendetail aktor-aktor yang terlibat pada sistem. Aktor dalam maksud ini merujuk pada sebuah entitas yang berinteraksi dengan sistem, baik itu pengguna, sistem yang lain, atau sebuah perangkat yang mendukung operasional sistem. Pada Tabel 1 di bawah ini menyajikan informasi lengkap mengenai setiap aktor, termasuk deskripsi, peran, serta tanggung jawab mereka dalam sistem. Tujuan memahami mendalam tentang aktor-aktor ini ialah untuk memastikan bahwa semua aspek sistem dapat berfungsi dengan baik.

Tabel 1. Aktor Dalam Sistem

Aktor	Deskripsi
Penyedia	Aktor dalam sistem yang mentransmisikan lokasi geografisnya ke dalam platform ini adalah pengguna yang secara khusus menetapkan posisinya saat membuat sebuah ruang berbagi baik chat maupun lokasi dalam sistem.
<i>Observer</i>	Aktor dalam sistem yang berperan sebagai penerima informasi lokasi yang dikirimkan oleh Penyedia. Posisi aktor ini ditetapkan khusus untuk pengguna yang melakukan login di dalam suatu ruang berbagi.
<i>Geolocator</i>	Aktor eksternal dalam sistem yang berperan untuk mendukung sistem dalam melaksanakan proses pelacakan lokasi Penyedia, <i>observer</i> dan pengguna.

Dengan pemahaman mengenai aktor-aktor ini, kita dapat melanjutkan untuk melibatkan mereka dalam skenario pengguna yang tercantum dalam diagram use case. Skenario pengguna berfungsi sebagai panduan untuk menggambarkan bagaimana aktor berinteraksi dengan sistem dalam berbagai situasi. Melalui pendekatan ini, dapat dipastikan bahwa setiap elemen sistem diimplementasikan dengan baik untuk memenuhi kebutuhan pengguna dan mencapai tujuan sistem secara keseluruhan. Keterlibatan aktor dalam skenario penggunaan akan membantu dalam merancang interaksi yang lebih efektif, sehingga pengguna dapat dengan mudah memahami cara menggunakan sistem.

Setelah mengidentifikasi aktor-aktor dalam sistem, langkah berikutnya adalah memahami fungsionalitas utama yang diperlukan oleh sistem. Fungsionalitas ini mencakup berbagai fitur dan kemampuan yang harus dimiliki oleh sistem agar dapat bekerja secara optimal. Di bawah ini merupakan Tabel 2 yang akan menjelaskan rincian fungsionalitas, yang akan memberikan gambaran tentang bagaimana sistem beroperasi dan berinteraksi dengan aktor.

Tabel 2. Daftar Fungsional Sistem Yang Akan Dibutuhkan

No	Kebutuhan Fungsional Sistem	Kasus Penggunaan
1	Sistem perlu menyajikan sebuah platform dimana pengguna dapat membuat kelompok baru untuk berbagi lokasi. Ini berfungsi menjadi wadah yang baru untuk mendistribusikan informasi lokasi dengan pengguna lain.	<i>Sharing room</i>
2	Platform ini menawarkan sebuah laman yang memungkinkan pengguna mendaftar sebagai anggota dalam ruang berbagi yang telah dibuat sebelumnya.	<i>Join sharing room</i>
3	Saat berada di halaman pelacakan lokasi, sistem perlu mengidentifikasi posisi pengguna yang ingin membagikan lokasinya dan mengirimkannya ke <i>server</i> .	Lacak dan kirim lokasi
4	Pada saat sistem menerima data lokasi dari pengguna yang ingin berbagi lokasi, sistem perlu mengirimkan informasi lokasi tersebut kepada pengguna lain yang telah terdaftar sebagai Pengamat (<i>Observer</i>) dalam ruang berbagi yang telah dibuat oleh pengguna pengirim.	<i>Update lokasi</i>
5	Sistem perlu memberikan fungsi kepada pengguna yang menerima data lokasi untuk merangkum informasi lokasi tersebut dalam format peta dua dimensi.	Visualisasi lokasi

Dengan rincian fungsionalitas ini, kita dapat mengidentifikasi bagaimana kasus penggunaan melibatkan aktor-aktor dan bagaimana sistem secara keseluruhan merespons kebutuhan pengguna untuk berinteraksi dan berbagi lokasi. Proses ini akan membantu dalam merancang alur kerja yang efisien dan intuitif bagi pengguna. Selain itu, pemahaman yang baik tentang fungsionalitas ini akan memungkinkan pengguna untuk mengantisipasi potensi masalah yang mungkin muncul selama implementasi dan penggunaan sistem.

Perancangan

Pada tahap perancangan, dilanjutkan menjelaskan daftar fungsionalitas dalam Tabel 2. Langkah berikutnya adalah memahami entitas data yang diperlukan untuk mendukung operasional sistem pada Tabel 3. Entitas Data apa saja yang diperlukan pada Sistem *Sharing room* memberikan gambaran tentang karakteristik dan tipe data yang terlibat. Berikut penjelasannya:

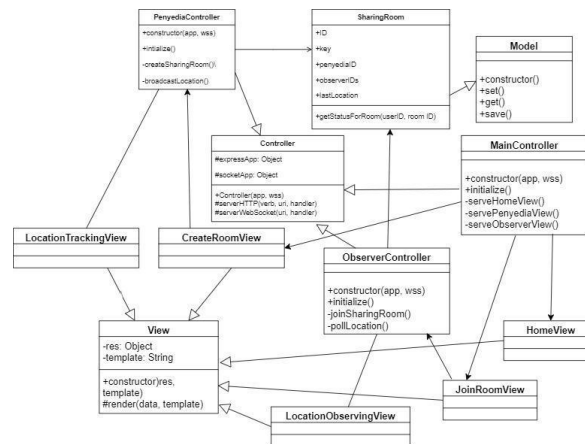
Tabel 3. Entitas Data Pada Sistem *Sharing Room*

Karakteristik	Tipe Data	Keterangan
<i>Room id</i>	<i>String</i>	Primary key untuk setiap entitas dalam ruang berbagi, otomatis dibuat oleh sistem dan wajib ada saat entitas tersebut sedang dibuat.
Key	<i>String</i>	Kata kunci digunakan untuk autentikasi pada proses pendaftaran ke dalam <i>sharing room</i> . Berasal data sebelumnya yang kemudian dimasukkan ke awal pembuatan <i>sharing room</i> oleh pengguna. Oleh karena itu harus ada saat pembuatan entitas berlaku.
Id penyedia	<i>String</i>	Identifier dari Penyedia yang membuat entitas ruang berbagi, Perlu hadir pada saat penciptaan entitas sedang berlangsung.
<i>Observer id</i>	<i>Array</i>	Array yang berisi daftar identifikasi (identifier) dari <i>Observer</i> yang telah mendapat ID untuk mendaftar sebagai penerima dalam ruang berbagi.
<i>Latitude</i>	<i>Float</i>	Informasi garis lintang dari lokasi terakhir yang dikirimkan dalam sistem opsional.
Longitude	<i>Float</i>	Informasi garis bujur dari lokasi terakhir yang dikirimkan dalam sistem opsional.

Dengan penjelasan tentang entitas data ini, kita dapat melihat bagaimana setiap elemen informasi dipetakan dan disimpan dalam sistem untuk memastikan integritas dan fungsionalitas keseluruhan. Data-data ini merupakan pondasi penting untuk merancang struktur database yang efektif dan mendukung kebutuhan operasional sistem dengan baik.

Dengan menjelaskan secara jelas mengenai aktor-aktor, fungsionalitas sistem, dan entitas data yang terlibat, tahap perancangan sistem, kita bisa menciptakan kerangka kerja yang kokoh untuk implementasi sistem berbasis lokasi. *Use case diagram* (Gambar 1) memberikan gambaran visual interaksi antara pengguna dan sistem, sedangkan tabel-tabel (Tabel 1, Tabel 2, dan Tabel 3) menyajikan rincian mengenai peran aktor, fungsionalitas yang dibutuhkan, dan struktur data yang mendukung.

Class diagram menampilkan hasil dari perancangan arsitektur sistem, sedangkan perancangan basis data menghasilkan tabel yang menjelaskan struktur entitas data nonvolatil dalam sistem. Perancangan antarmuka pengguna menghasilkan peta atau model tampilan antarmuka beserta fungsinya tentang komponen beberapa antarmuka. Di bawah ini merupakan *class diagram* pada sistem ini:



Gambar 3. Class diagram

Penjelasan pada gambar yaitu, pertama dimulai dengan penyedia membuat sebuah plat *sharing room* beserta memasukkan nama nya. Lalu dilanjut dengan *server* yang menciptakan sebuah plat baru dan mengarahkan langsung penyedia ke halaman web yang dimaksud. Pada halaman ini, penyedia mendapatkan kode unik dari plat *sharing room* yang sudah diciptakan oleh sistem. Fungsi dari kode unik ini ialah agar *observer* bisa mengakses sistem dan lanjut beralih bergabung ke *sharing room*. Kode unik dibagikan oleh penyedia kepada orang lain yang akan menjadi *observer*. Dari *observer*, ia mengakses sistem dan bergabung dengan kode nama dan room name yang tertera pada kode unik dan diberikan oleh penyedia. Langkah selanjutnya, melakukan pengiriman lokasi ke *server* yang dilakukan oleh si Penyedia. Lalu, lokasi diterima oleh *observer* dengan disiarkannya pada halaman yang sudah terhubung halaman *sharing room*. Penjelasan terakhir, didapat yaitu pengguna yang berada pada room tersebut bisa saling melakukan share location, tidak hanya share location saja, setiap pengguna juga bisa melakukan komunikasi yang baik jika mereka masih berada pada halaman *sharing room* yang sama.

Dengan pondasi ini, perancangan arsitektur, basis data, dan antarmuka pengguna dapat dilakukan dengan lebih terfokus dan efisien. Melalui integrasi elemen-elemen, diharapkan sistem dapat memberikan pengalaman pengguna yang intuitif dan responsif, memungkinkan pengguna untuk dengan mudah berbagi lokasi dan berinteraksi dalam ruang berbagi.

Langkah selanjutnya adalah mengimplementasikan rancangan ini ke dalam sebuah sistem yang berjalan dengan lancar dan memenuhi kebutuhan pengguna dengan baik. Proses ini akan melibatkan pengembangan perangkat lunak, uji coba, dan iterasi untuk memastikan bahwa setiap elemen berfungsi secara optimal sesuai dengan spesifikasi yang telah dirancang sebelumnya.

Implementasi

Dalam tahap implementasi sistem, fokus utama pada pengembangan sistem *client* dan *server*. Pada sisi *server*, proses implementasi melibatkan *JavaScript* berorientasi objek dalam lingkungan runtime *Node.JS*. *Server* ini dibangun dengan menggunakan non-relational database *MongoDB*, dimana *framework Express.JS* dan *Mongoose.JS* turut dimanfaatkan. berikut hasil sistem di sisi *server* :

```

index.html
server
  utils
  server.js
  .gitignore
  package-lock.json
  package.json
package.json

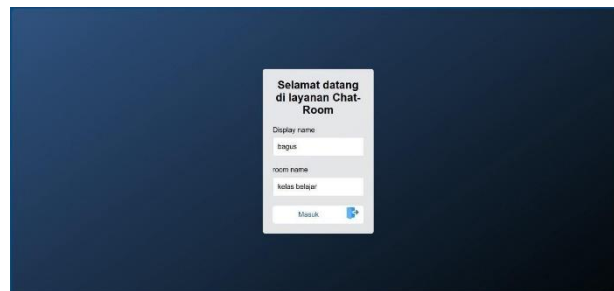
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Debugger attached.
> chat-app@1.0.0 start
> node server/server.js
Debugger attached.
Server is up on port 3000
A new user just connected
A new user just connected
A new user just connected

```

Gambar 4. Hasil sistem di sisi server

Gambar 1 menampilkan antarmuka sistem sisi *server* yang berhasil diimplementasikan menggunakan runtime environment *Node.JS* dan *JavaScript* berorientasi objek. Sistem *server* ini menggunakan non-relational database *MongoDB* untuk menyimpan data dengan struktur yang dinamis. *Framework Express.JS* dan *Mongoose.JS* mendukung manajemen rute, middleware, dan interaksi efisien dengan *MongoDB*. Gambar 1 memberikan ikhtisar visual tentang kinerja optimal *server*, menggambarkan integrasi sinergis teknologi kunci. Sistem di sisi *server* menjamin kelancaran, kehandalan, dan dasar yang kokoh untuk fungsionalitas keseluruhan aplikasi.

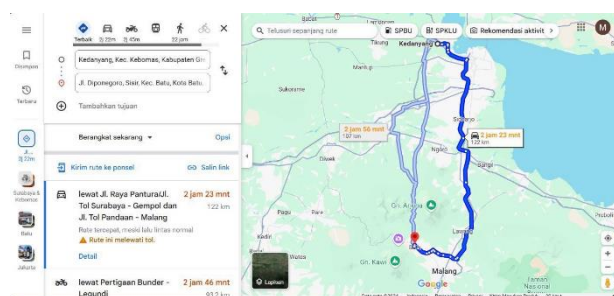
Sementara itu, pada sisi *client*, implementasi dilakukan dengan menggabungkan kekuatan *HTML*, *CSS*, dan *JavaScript* untuk menciptakan antarmuka yang responsif dan intuitif. Melalui tahap ini, berhasil dibangun sebuah prototipe sistem yang dapat dioperasikan. Berikut hasil sistem di sisi *client*:



Gambar 5. Tampilan awal



Gambar 6. Halaman sharing room



Gambar 7. Hasil berbagi lokasi

Gambar 5, 6, dan 7 menunjukkan antarmuka sistem di sisi *client* yang berhasil diimplementasikan. Pengembangan ini menggunakan kombinasi *HTML*, *CSS*, dan *JavaScript* untuk menciptakan tampilan yang responsif dan intuitif. Gambar 5 menggambarkan tampilan awal aplikasi, memberikan gambaran menyeluruh tentang desain dan navigasi.

Gambar 6 menampilkan halaman khusus untuk berbagi ruangan, menonjolkan elemen-elemen yang mendukung interaksi pengguna dalam pengaturan ruangan. Sementara itu, Gambar 7 memberikan ilustrasi visual tentang hasil dari fitur berbagi lokasi, menunjukkan kemampuan sistem untuk melibatkan pengguna dalam pertukaran informasi Lokasi dan tidak hanya itu dengan menggunakan *Google Maps API* dapat memberikan pengalaman bagi pengguna secara lebih baik

dengan memberikan informasi terkait rute tercepat dan informasi kendala atau info dari penggunaan rute tersebut.

Melalui desain yang ramah pengguna, sistem di sisi *client* tidak hanya memberikan tampilan yang menarik tetapi juga menawarkan fungsionalitas yang diperlukan. Pengguna juga dapat dengan mudah berinteraksi dengan aplikasi, berbagi ruangan, dan melihat hasil dari berbagi lokasi dari setiap pengguna dan menentukan rute tercepat dan estimasi waktu perjalanan.

Pengujian

Pada pengujian, evaluasi tingkat sistem dilakukan melalui pembuatan beberapa skenario, uji eksekusi skenario tersebut dan analisis hasil untuk menentukan apakah dapat membuktikan pencapaian definisi kebutuhan atau sebaliknya. Dalam menguji kebutuhan non-fungsional sistem, digunakan Network Panel di Microsoft Edge Developer Tools. Alat perangkat lunak ini berfungsi untuk mengevaluasi interaksi jaringan antara *client* dan *server* pada proses berbagi lokasi dengan fokus pada penggunaan atau tidaknya protokol *Websocket*.

Pada evaluasi tingkat unit dan evaluasi tingkat komponen, hasil yang diperoleh menunjukkan bahwa setiap unit (kelas) dalam sistem mampu menjalankan fungsinya baik saat diuji secara mandiri maupun saat diintegrasikan dengan komponen terkait.

Tabel 4. Pengujian

Titik-Titik Observasi	Koordinat		Hasil Pengujian Maps	
	Latitude	Longitude	Latitude	Longitude
1	7.1667525	112.6103196	7.1667525	112.6128945
2	7.2574719	112.7495134	7.2574719	112.7520883
3	7.178224	112.6090841	7.178224	112.611659

Berdasarkan evaluasi terhadap kebutuhan fungsional dan non-fungsional sistem, dapat disimpulkan bahwa implementasi sistem berhasil memenuhi semua spesifikasi yang telah ditetapkan. Adanya selisih yang memunculkan saran penggunaan sistem terhubung dengan fitur GPS yang aktif, agar tingkat akurasi dari berbagi lokasi tersebut bisa mendapatkan nilai yang optimal dan tingkat penyimpangan rata-rata yang lebih kecil jika dibandingkan dengan tidak memakai penggunaan koneksi GPS.

SIMPULAN

Adapun beberapa kesimpulan yang dapat ditarik dari seluruh proses penelitian ini. Proses rekayasa kebutuhan sistem diperiksa secara menyeluruh dalam menguji validitas sistem. Tujuannya untuk memastikan bahwa setiap elemen kebutuhan sistem dapat diandalkan dan sesuai dengan kebutuhan yang diinginkan. Pada tahap perancangan, ditampilkan *use case diagram* untuk menjelaskan struktur entitas data dalam sistem. Dijelaskan juga pada beberapa tabel mengenai entitas data serta apa saja yang dibutuhkan sistem. Sedangkan proses implementasi memperoleh sebuah prototipe sistem yang bisa dioperasikan. *Server* dari sistem dibangun di atas runtime environment *Node.JS* menggunakan non-relational database *MongoDB* dengan memanfaatkan *framework Express.JS* dan *Mooser.JS*.

Dapat disimpulkan bahwa rencana penelitian ini adalah benar. Hasil ditunjukkan pada bagian pengujian menunjukkan evaluasi yang menyimpulkan bahwa implementasi sistem berhasil memenuhi semua spesifikasi yang sudah diatur yang berarti sistem ini dapat dikembangkan menggunakan *Websocket*. Studi ini menunjukkan bahwa teknologi *Websocket*, yang berfungsi sebagai teknologi komunikasi yang bagus dalam proses berbagi lokasi sehingga dapat digunakan untuk mengembangkan sistem berbagi lokasi secara *realtime*.

DAFTAR PUSTAKA

Alviando, L., Bhawiyuga, A., & Kartikasari, D. P. (2023). Penerapan *Websocket* pada Sistem Live Chat berbasis Web (Studi Kasus Website Kwikku.com). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 854-862.

- Hamdani, M. A., & Utomo, S. (2021). Sistem Informasi Geografis (SIG) Pariwisata Kota Bandung Menggunakan Google Maps API dan PHP. *Jurnal Teknologi Informasi dan Komunikasi*.
- Hidayah, A., Agustina, D. S., Indirawati, D., & Amalia, S. (2021). Sistem Informasi Pelatihan Lisensi Penerbangan Pada Citilink Berbasis Web. *JTIM: Jurnal Teknik Informatika Mahakarya*, 61-68.
- Karsana, I. W., & Mahendra, G. S. (2021). Sistem Informasi Geografis Pemetaan Lokasi Puskesmas Menggunakan Google Maps API di Kabupaten Bandung. *J-Icon : Jurnal Komputer dan Informatika*, 160-167.
- Maharani, D., Helmiah, F., & Rahmadani, N. (2021). Penyuluhan Manfaat Menggunakan Internet dan Website Pada Masa Pandemi Covid-19. *Abdiformatika: Jurnal Pengabdian Masyarakat Informatika*, 1-7.
- Maulana, R. (2021). Implementasi Web Socket Pada Sistem Pelayanan Pasien Rawat Jalan Pada Puskesmas Kabupaten Gowa. *Jurnal INSTEK (Informatika Sains dan Teknologi)*.
- Permadi, I. W., & Prasida, T. A. (2022). Penerapan Teknologi Ajax pada Desain Website Pariwisata Kota Salatiga menggunakan UML dan UCD. *Jurnal Bina Komputer*, 39-50.
- Pranatawijaya, V. H. (2021). Penerapan Location Based Servis (LBS) dalam Prototipe Pengenalan Ruangan dengan Metode Extreme Programming. *Jurnal Teknologi Informasi : Jurnal Keilmuan dan Aplikasi Bidang Teknik Informatika*, 92-99.
- Samantha, P. D., & Anggalih, N. N. (2024). Perancangan Desain UI/UX Fitur Fun Aplikasi Berbagi Lokasi Secara Real-Time. *Jurnal Desgrafia*, 49-63.
- Sie, J. B., Musdar, I. A., & Bahri, S. (2022). Pengujian White Box Testing Terhadap Website Room Menggunakan Teknik Basis Path. *Jurnal Ilmu Komputer Kharisma Tech*, 45-57.
- Yuwahendira, K. A., & Ratnasari, C. I. (2020). Penerapan Teknologi Location-Based Services Dalam Mobile Application: Suatu Tinjauan Literatur. *Journal UII*.