

Layanan Real Time Berbasis Websocket Dan Speechsynthesis Pada Website Gereja

Rinam*, David

Teknik Informatika, STMIK Pontianak, Jl. Merdeka, No. 374, Pontianak, Kalimantan Barat,
0561-735555

e-mail: *[1inamrinam72@gmail.com](mailto:inamrinam72@gmail.com), [2david-stmik@stmikpontianak.ac.id](mailto:david-stmik@stmikpontianak.ac.id)

Abstrak

Gereja GPIB Siloam Pontianak selama ini menghadapi masalah dalam pelayanan digital, terutama kurangnya interaksi real-time antara jemaat dan admin gereja serta keterbatasan akses informasi bagi jemaat lanjut usia dan jemaat dengan hambatan visual. Tidak tersedianya fitur komunikasi dua arah secara langsung menyebabkan respons terhadap pertanyaan jemaat menjadi lambat. Selain itu, penyampaian renungan dan pengumuman hanya dalam bentuk teks membuat sebagian jemaat kesulitan menerima informasi dengan baik. Penelitian ini bertujuan mengatasi permasalahan tersebut dengan mengimplementasikan fitur chat real-time menggunakan WebSocket serta fitur text-to-speech menggunakan SpeechSynthesis API. Pengujian menunjukkan bahwa fitur chat mampu merespons pesan dengan waktu rata-rata 56 ms pada kondisi normal dan tetap stabil pada 112 ms saat trafik meningkat hingga 150 koneksi simultan. Fitur SpeechSynthesis berhasil mengonversi konten renungan menjadi audio dengan tingkat keberhasilan pemutaran 100% di browser modern. Kesimpulan penelitian ini adalah bahwa integrasi WebSocket dan SpeechSynthesis secara signifikan meningkatkan kualitas pelayanan digital gereja, mempercepat komunikasi, dan memperluas akses bagi jemaat berkebutuhan khusus. Penelitian ini berkontribusi pada model layanan digital gereja yang lebih inklusif, interaktif, dan responsif.

Kata kunci—WebSocket, SpeechSynthesis, Sistem Informasi Gereja, Laravel.

Abstract

GPIB Siloam Church Pontianak has long encountered challenges in its digital services, particularly the absence of real-time interaction between congregants and church administrators, as well as limited access to information for elderly members and those with visual impairments. The lack of a direct two-way communication feature has resulted in slow responses to congregants' inquiries. Moreover, devotionals and announcements delivered solely in text form make it difficult for certain members to receive information effectively. This study aims to address these issues by implementing a real-time chat feature using WebSocket and a text-to-speech feature powered by the SpeechSynthesis API. Testing results show that the chat feature is able to respond to messages with an average latency of 56 ms under normal conditions and remains stable at 112 ms when traffic increases to 150 simultaneous connections. The SpeechSynthesis feature successfully converts devotional content into audio with a 100% playback success rate across modern browsers. The findings indicate that integrating WebSocket and SpeechSynthesis significantly enhances the church's digital services by improving communication speed and expanding accessibility for congregants with special needs. This research contributes to the development of a more inclusive, interactive, and responsive digital service model for churches.

Keywords—WebSocket, SpeechSynthesis, Church Information System, Laravel.

1. PENDAHULUAN

Pelayanan gereja di era digital menuntut kemampuan untuk menyampaikan informasi secara cepat, mudah, dan dapat diakses oleh seluruh jemaat tanpa kecuali [1]. Namun pada praktiknya, banyak gereja masih bergantung pada metode komunikasi konvensional yang kurang responsif dan tidak mendukung kebutuhan aksesibilitas jemaat lanjut usia atau jemaat dengan hambatan visual [2]. Hal ini juga dialami oleh GPIB Siloam Pontianak, yang memiliki website namun masih mengalami kekurangan pada dua aspek penting: interaksi real-time dan aksesibilitas konten.

Secara umum, permasalahan yang dihadapi gereja adalah kurangnya media komunikasi dua arah yang cepat antara jemaat dan pengurus gereja. Secara khusus, GPIB Siloam Pontianak tidak memiliki fitur chat langsung yang memungkinkan jemaat mengajukan pertanyaan atau meminta informasi secara instan [3]. Selain itu, konten renungan dan pengumuman hanya tersedia dalam bentuk teks sehingga jemaat yang mengalami kesulitan membaca, baik karena faktor usia maupun keterbatasan visual, tidak dapat mengaksesnya secara optimal [4], [5].

Untuk mengatasi permasalahan tersebut, solusi yang diusulkan adalah implementasi dua teknologi modern. Pertama, WebSocket menjadi solusi karena memungkinkan adanya komunikasi dua arah antara klien dan server secara real-time melalui satu koneksi TCP [6]. Kedua, teknologi SpeechSynthesis yang memungkinkan teks diubah menjadi suara diintegrasikan untuk memberikan kemudahan bagi jemaat mendengar informasi tanpa harus membacanya [7]. Pengembangan fitur-fitur ini akan diimplementasikan pada website yang ada dengan menggunakan framework Laravel, yang diharapkan dapat menciptakan platform yang tidak hanya lebih interaktif tetapi juga inklusif bagi seluruh pengguna [8].

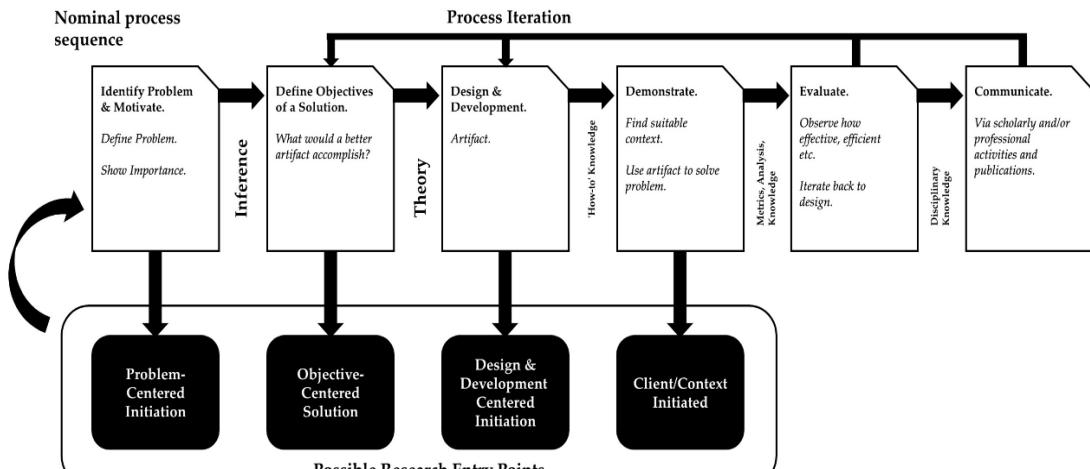
Penelitian terdahulu umumnya menghasilkan website berupa pelayanan dan informasi kegiatan gereja [9] atau aplikasi berbasis web yang menampilkan informasi dan pengelolaan data [10]. Penelitian ini memiliki tujuan untuk merancang sebuah sistem informasi gereja yang mengintegrasikan teknologi WebSocket dan SpeechSynthesis untuk meningkatkan interaksi dan aksesibilitas. Penelitian terdahulu lebih banyak berfokus pada pengembangan sistem informasi gereja secara umum, seperti pengelolaan data jemaat atau penyajian informasi kegiatan. Namun, penelitian yang mengintegrasikan interaksi real-time berbasis WebSocket dan aksesibilitas audio berbasis SpeechSynthesis pada website gereja masih sangat terbatas. Inilah celah (gap) penelitian yang ingin dijawab.

Implikasi dari penerapan ini adalah terciptanya komunikasi yang lebih cepat dan efisien, serta meningkatnya jangkauan pelayanan gereja secara digital kepada jemaat berkebutuhan khusus. Kontribusi dan kebaharuan penelitian ini adalah mengimplementasikan teknologi WebSocket dan SpeechSynthesis dalam satu platform website gereja, menghadirkan sebuah sistem yang lebih interaktif dan inklusif dibandingkan penelitian sebelumnya.

Penelitian ini bertujuan untuk mengembangkan fitur chat real-time menggunakan WebSocket untuk meningkatkan interaksi jemaat dengan gereja, mengimplementasikan fitur text-to-speech menggunakan SpeechSynthesis API untuk meningkatkan aksesibilitas konten dan mengukur performa WebSocket dalam kondisi trafik normal dan tinggi. Dengan demikian, penelitian ini diharapkan dapat meningkatkan pelayanan digital gereja melalui interaksi yang cepat, akses informasi yang inklusif, dan platform yang lebih ramah bagi jemaat berkebutuhan khusus.

2. METODE PENELITIAN

Penelitian ini menggunakan Design Science Research Method (DSRM) sebagai kerangka pengembangan solusi, serta Extreme Programming (XP) sebagai metode implementasi perangkat lunak [11], [12]. Tahapan dalam DSRM diilustrasikan pada Gambar 1 dibawah ini.



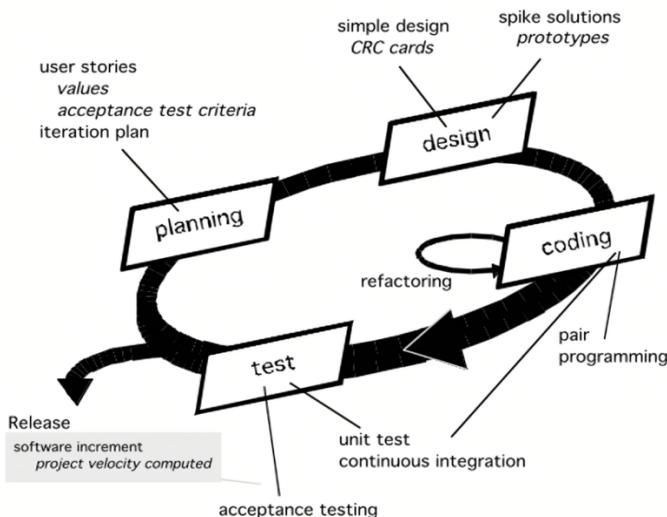
Gambar 1 Tahapan DSRM

1. Identifikasi Masalah atau Peluang
 - a. Website belum memiliki fitur chat real-time.
 - b. Jemaat lansia kesulitan membaca konten renungan dan pengumuman.
 - c. Tidak ada fitur aksesibilitas berbasis audio.
2. Definisi Tujuan dan Kriteria Keberhasilan
 - a. Membangun chat real-time berbasis WebSocket.
 - b. Mengembangkan fitur text-to-speech yang dapat membaca konten renungan.
 - c. Menyediakan platform yang inklusif dan mudah diakses.
3. Desain dan pengembangan
 - a. WebSocket: menggunakan Pusher dan Laravel Echo untuk komunikasi dua arah.
 - b. SpeechSynthesis: mengaktifasi text-to-speech pada elemen tertentu (speakable).
 - c. Frontend: menggunakan Bootstrap 5 dan JavaScript.
 - d. Backend: Laravel sebagai API chat serta penyimpanan data ke MySQL
4. Demonstrasi
 - a. Fitur chat diuji dengan beberapa akun.
 - b. Fitur SpeechSynthesis diuji pada halaman renungan.
5. Evaluasi
 - a. Pengujian fungsi chat.
 - b. Pengujian performa WebSocket pada trafik tinggi.
 - c. Pengujian keberhasilan text-to-speech.
6. Komunikasi

Mendokumentasikan dan menkomunikasikan hasil penelitian dalam bentuk laporan, artikel, presentasi, atau demonstrasi kepada pihak yang berkepentingan, baik akademik maupun praktisi.

Metode perancangan perangkat lunak yang digunakan dalam penelitian ini adalah Extreme Programming. Metode Extreme Programming mencakup seperangkat aturan dan praktik yang terjadi dalam konteks empat kegiatan kerangka kerja, yaitu, perencanaan, desain,

pengkodean, dan pengujian yang dilaksanakan secara iteratif dan berulang.



Gambar 2 Tahapan Extreme Programming

Tahapan-tahapan model Extreme Programming untuk perancangan perangkat lunak [13]:

1. Planning (Perencanaan)

Tahapan ini merupakan langkah awal dalam pembangunan sistem dimana dalam tahapan ini dilakukan beberapa kegiatan perencanaan yaitu identifikasi permasalahan, menganalisa kebutuhan sampai dengan penetapan jadwal pelaksanaan pembangunan sistem.

2. Design (Perancangan)

Tahapan berikutnya adalah perancangan dimana pada tahapan ini dilakukan kegiatan pemodelan yang dimulai dari pemodelan sistem, pemodelan arsitektur sampai dengan pemodelan basis data. memberikan gambaran umum tentang alur kerja dan hubungan antar komponen dalam sistem yang akan dibangun. Pemodelan arsitektur digunakan untuk merancang struktur utama aplikasi, termasuk komponen-komponen perangkat lunak, modul, dan bagaimana interaksi di antara komponen tersebut.

3. Coding (Pengkodean)

Tahap ini merupakan tahapan implementasi atau pembuatan kode program sesuai dengan rancangan sistem yang dibuat pada tahap sebelumnya. Tahap ini dapat dilakukan secara iterative (Code Refactoring) jika terdapat perubahan. Proses iterative memungkinkan pengembang untuk terus memperbaiki kualitas kode dan memastikan bahwa program berjalan sesuai dengan spesifikasi yang diinginkan.

4. Testing (Pengujian)

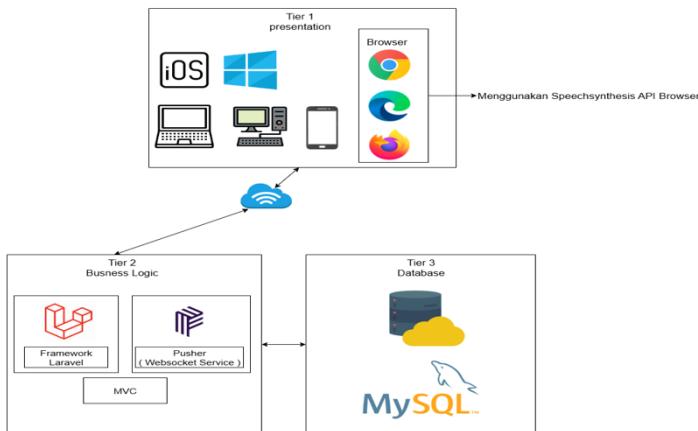
Pada tujuan ini dilakukan pengujian perangkat lunak atau sistem. Pengujian dilakukan pada setiap modul yang sedang dibuat. Hal tersebut bertujuan untuk memastikan sistem yang dikembangkan sesuai dengan permintaan dan kebutuhan penggunaan atau klien. Perbaikan modul akan terus dilakukan sampai modul yang dikembangkan sesuai dengan permintaan.

3. HASIL DAN PEMBAHASAN

Perancangan dimulai dengan menggunakan arsitektur sistem, merancang database dengan normalisasi dan ERD, serta memodelkan sistem dalam bentuk diagram UML, rancangan sistem penggunaan web dari sisi client dapat dijelaskan dengan usecase diagram, activity diagram, class diagram dan sequence diagram.

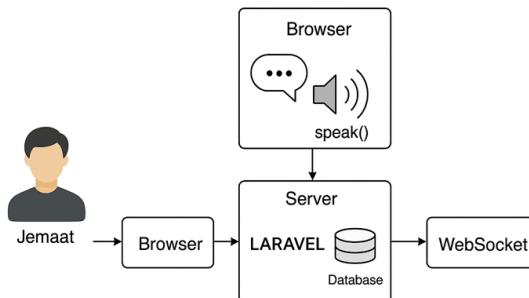
3.1 Desain dan pengembangan

3.1.1 Arsitektur Sistem



Gambar 3 Arsitektur Sistem GPIB Siloam Pontianak

Sistem ini dirancang dengan arsitektur tiga lapis untuk memastikan performa dan keamanan. Lapisan frontend menggunakan Bootstrap 5 untuk antarmuka yang responsif dan aksesibel. Pada backend, framework Laravel mengelola logika bisnis, keamanan, serta fitur-fitur utama seperti chat real-time via WebSocket (Pusher) dan API untuk SpeechSynthesis. Sementara itu, lapisan data menggunakan MySQL yang terstruktur, aman, dan teroptimasi. Pemisahan ini memungkinkan pengembangan dan pengujian yang independen, menghasilkan sistem yang mudah dipelihara, tangguh, dan adaptif.



Gambar 4 Arsitektur Implementasi WebSocket dan SpeechSynthesis pada Sistem Gereja

Gambar 4 menunjukkan alur komunikasi antara browser jemaat, server Laravel, layanan WebSocket, serta API text-to-speech yang dijalankan di sisi browser. WebSocket menangani pesan real-time, sedangkan SpeechSynthesis berjalan di frontend untuk mengonversi teks menjadi suara.

1. Implementasi WebSocket

Implementasi WebSocket pada sistem informasi Gereja GPIB Siloam Pontianak dilakukan untuk menghadirkan komunikasi dua arah secara real-time antara jemaat dan admin gereja. Teknologi ini dipilih karena mampu mempertahankan koneksi yang selalu aktif (persistent connection) sehingga pesan dapat dikirim dan diterima tanpa perlu memuat ulang halaman. Sistem dibangun menggunakan layanan WebSocket berbasis Pusher yang terintegrasi dengan Laravel Echo, sehingga setiap pesan yang dikirim jemaat langsung diproses oleh server dan disiarkan kembali ke klien yang relevan dalam hitungan milidetik.

Pada sisi backend, Laravel mengelola logika penerimaan dan penyiaran pesan. Setiap pesan yang masuk diverifikasi, disimpan ke dalam database, kemudian dikirim kembali ke kanal WebSocket agar dapat diterima admin secara instan. Pada sisi frontend, komponen chat akan otomatis menampilkan pesan baru melalui listener event WebSocket tanpa adanya proses polling. Implementasi ini memberikan pengalaman komunikasi yang jauh lebih responsif dibandingkan metode konvensional berbasis HTTP request atau AJAX polling.

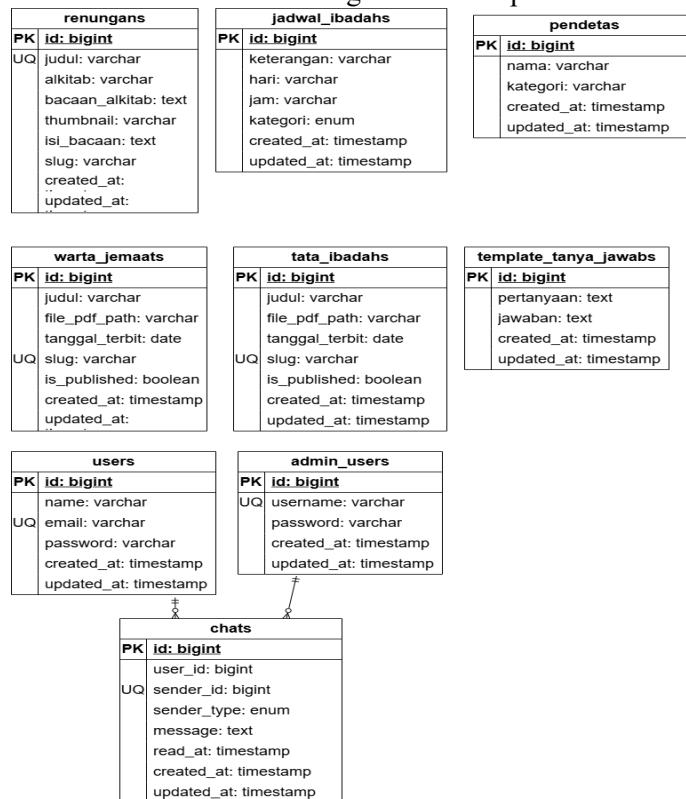
2. Implementasi SpeechSynthesis

Implementasi SpeechSynthesis dilakukan untuk meningkatkan aksesibilitas website, khususnya bagi jemaat lanjut usia atau jemaat dengan hambatan visual yang mengalami kesulitan membaca konten panjang seperti renungan harian dan pengumuman. SpeechSynthesis API diintegrasikan langsung pada sisi frontend, sehingga browser dapat mengonversi teks pada halaman menjadi suara tanpa memerlukan pemrosesan dari server atau perangkat tambahan lainnya.

Setiap elemen yang dapat dibacakan diberikan penanda kelas (class) khusus, sehingga sistem dapat mendeteksi dan mengatur antrian teks yang akan disintesis menjadi audio. Ketika pengguna menekan tombol “Play”, mekanisme SpeechSynthesis akan mengekstrak isi teks, memformatnya agar lebih natural, lalu mengirimkannya ke mesin text-to-speech bawaan browser. Proses ini juga disertai kontrol interaksi seperti tombol “Pause”, “Stop”, dan penyorotan (highlight) pada teks yang sedang dibacakan untuk meningkatkan kenyamanan pengguna.

3.1.2 Basis Data

Perancangan basis data dilakukan agar semua data dapat tersimpan secara terstruktur di dalam database, setiap web wajib memiliki basis data sesuai dengan keperluannya masing-masing, maka perlu dibuat satu basis data baru yang akan tersimpan ke dalam server agar dapat digunakan secara bersamaan [14]. Diagram hubungan entitas atau entity relation diagram digunakan untuk menggambarkan relationship antar entitas yang relevan dari suatu sistem yang umumnya digunakan untuk memodelkan rancangan atau blue print dari basis data.



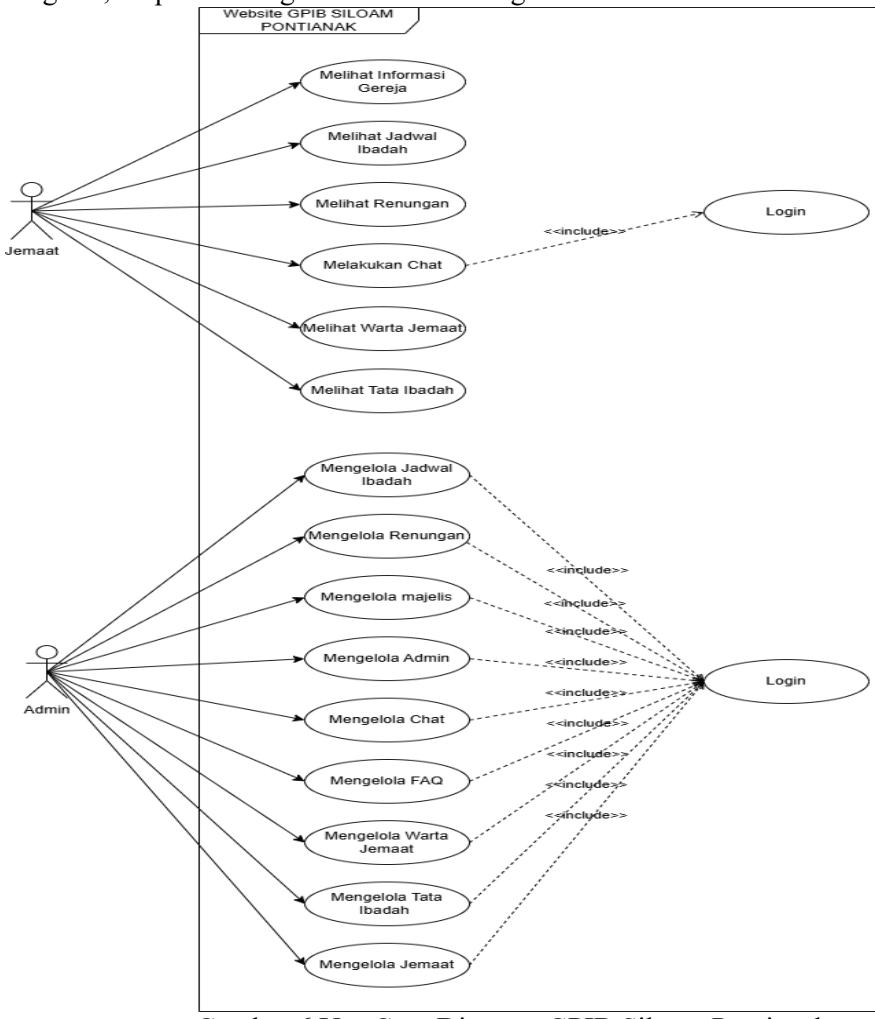
Gambar 5 Entity Relatiionship Diagram

Gambar 5 ditunjukkan diagram entitas-relasi yang memodelkan sembilan entitas utama dalam sistem, meliputi users, admin_users, chats, renungans, jadwal_ibadahs, pendetas, warta_jemaats, tata_ibadahs, dan template_tanya_jawabs. Setiap tabel dihubungkan melalui primary key dan foreign key untuk merepresentasikan relasi satu-ke-banyak, misalnya users ke chats serta pendetas ke jadwal_ibadahs, sehingga alur data dapat terlacak dengan jelas. Dengan

struktur ERD ini, sistem dapat menjamin integritas referensial, mengoptimalkan kecepatan query, serta memudahkan pengembangan dan pemeliharaan basis data secara berkelanjutan.

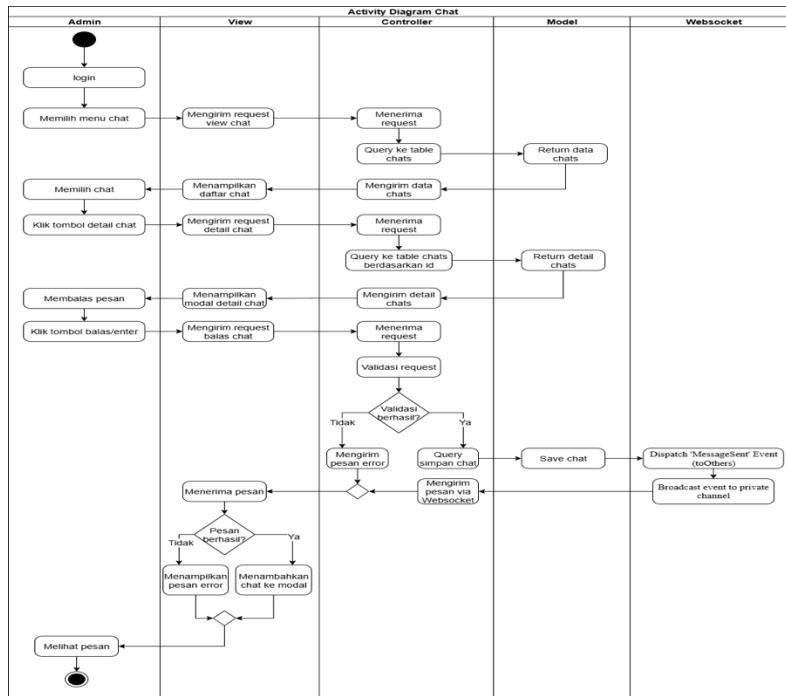
3.2 Unified Modelling Language

UML digunakan untuk membantu memetakan model yang diperlukan pada pengembangan sistem. Unified modeling language (UML) adalah bahasa pemodelan visual yang digunakan untuk menspesifikasi, memvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak. Pada tahapan perancangan ini, dibuat gambaran sistem menggunakan UML yang terbagi menjadi 4 diagram utama, yaitu Use Case Diagram, Activity Diagram, Sequence Diagram dan Class Diagram.



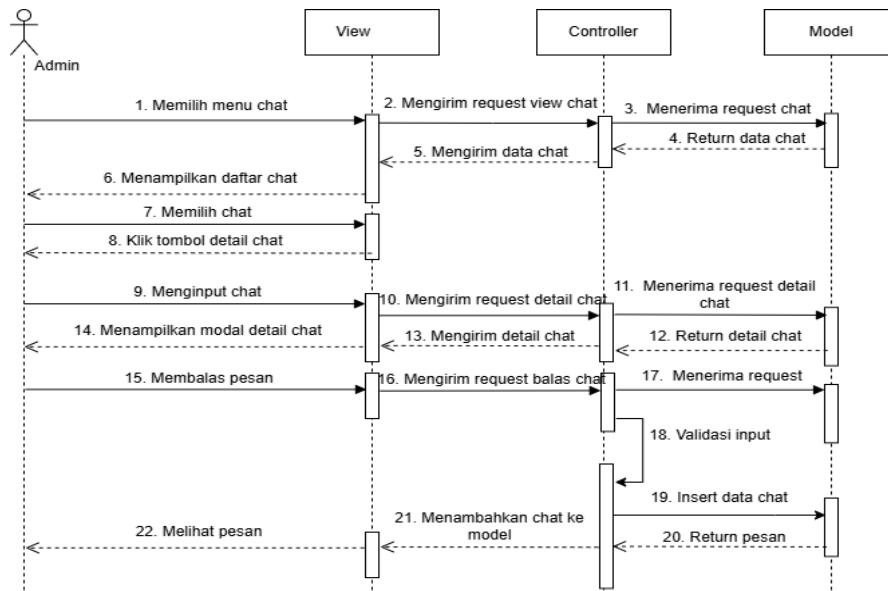
Gambar 6 Use Case Diagram GPIB Siloam Pontianak

Use Case Diagram adalah salah satu jenis diagram dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara pengguna dalam hal ini Admin dan Jemaat dengan sistem yang dikembangkan. Diagram ini memberikan gambaran umum tentang fungsi atau layanan utama yang disediakan oleh sistem dan pengguna berinteraksi dengan layanan tersebut.



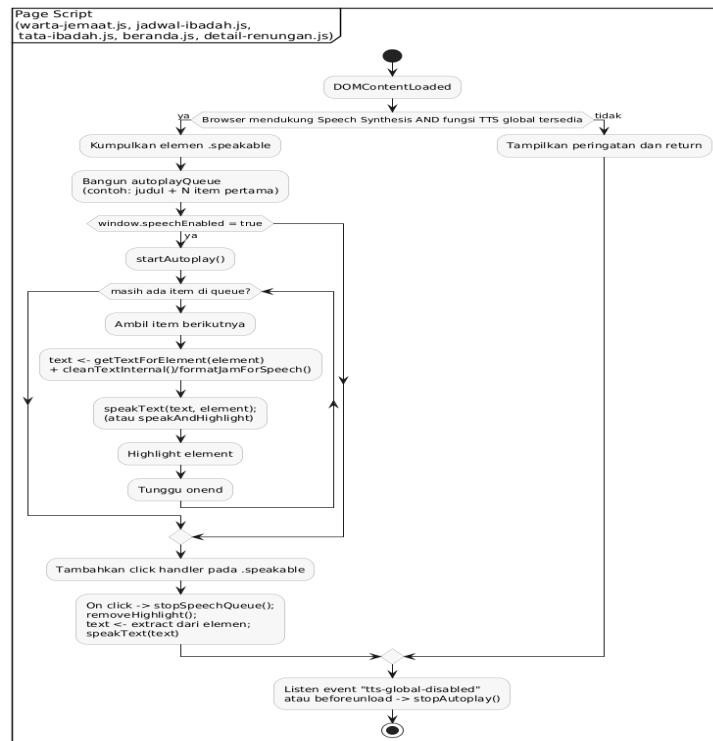
Gambar 7 Activity Diagram Kelola Admin Chat Dengan Websocket

Gambar 7 menjelaskan alur interaksi real-time antara admin dan jemaat, secara keseluruhan diagram ini memastikan validasi pesan, penyimpanan yang terdokumentasi, dan pembaruan instan bagi pengirim maupun penerima.



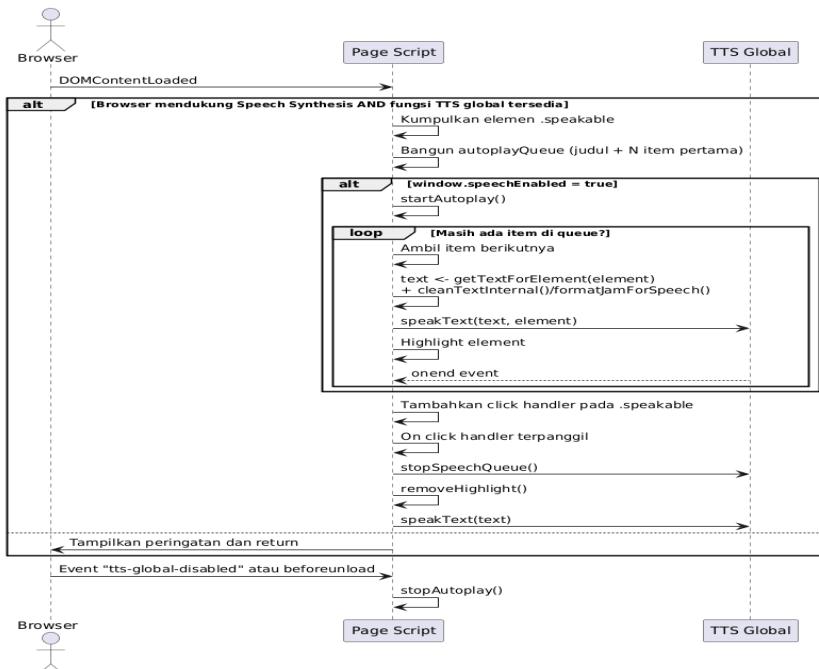
Gambar 8 Sequence Diagram Admin Chat Dengan Websocket

Gambar 8 menjelaskan pengelolaan chat oleh admin terbagi menjadi dua alur utama. Pertama, melihat daftar chat dan melihat detail dan membalas chat. Seluruh alur memastikan validasi input dan pembaruan antarmuka sehingga admin dapat memantau dan menindaklanjuti percakapan dengan jelas.



Gambar 9 Activity Diagram Speechsynthesis

Setelah DOMContentLoaded, sistem mengecek dukungan Web Speech Synthesis dan fungsi TTS global. Jika tidak didukung, ditampilkan peringatan; jika didukung, halaman mengumpulkan elemen speakable, menyusun antrean autoplay, lalu (jika speechEnabled aktif) membacakan tiap item berurutan, menyorot elemen, dan menunggu selesai sebelum lanjut. Setiap elemen speakable juga memiliki handler klik untuk menghentikan antrean/ucapan, menghapus sorotan, menekstrak teks, dan membacakan ulang; event tts-global-disabled/beforeunload menghentikan autoplay secara aman.



Gambar 10 Sequence Diagram Speechsynthesis

Gambar 10 menjelaskan Setelah DOMContentLoaded, Page Script mengecek dukungan TTS, mengumpulkan elemen speakable, menyusun antrean, lalu jika aktif menjalankan autoplay,

lalu ekstrak atau rapikan teks, lalu TTSGlobal.speakText, highlight dan lanjut saat onend. Klik pada speakable menghentikan antrean/ucapan, menghapus highlight, dan membacakan teks yang dipilih. Jika tidak didukung ditampilkan peringatan, maka event tts-global-disabled atau beforeunload menghentikan autoplay.

3.3 Pengkodean

Tahapan coding menghasilkan tampilan dari sistem yang dirancang untuk merealisasikan serta mengimplementasikan hasil rancangan agar dapat digunakan sesuai tujuan [15]. Struktur yang ada pada Framework Laravel yang menjadi acuan. Adapun Route berperan sebagai penghubung antara user dengan keseluruhan framework. Setiap url browser akan melewati route terlebih dahulu pada Laravel. Route akan menghubungkan melalui controller sehingga dapat memproses tugas selanjutnya tergantung perintah apa yang dituliskan pada bagian Controller.

```

1  Route::middleware(['auth:admin_users'])
2      ->prefix('dashboard/chat')
3      ->name('admin.chat.')
4      ->group(function () {
5
6          Route::get('/', [ChatController::class, 'index'])
7              ->name('index');
8
9          Route::get('/users-datable', [ChatController::class, 'getChatUsersForDataTable'])
10             ->name('users.datatable');
11
12         Route::get('/history/{user}', [ChatController::class, 'getChatHistoryForAdmin'])
13             ->name('history');
14
15         Route::post('/send-admin', [ChatController::class, 'storeAdminMessage'])
16             ->name('send.admin');
17
18         Route::post('/send-template', [ChatController::class, 'sendAdminTemplateMessage'])
19             ->name('send.template');
20
21         Route::post('/mark-read', [ChatController::class, 'markAdminMessagesAsRead'])
22             ->name('markread');
23     });

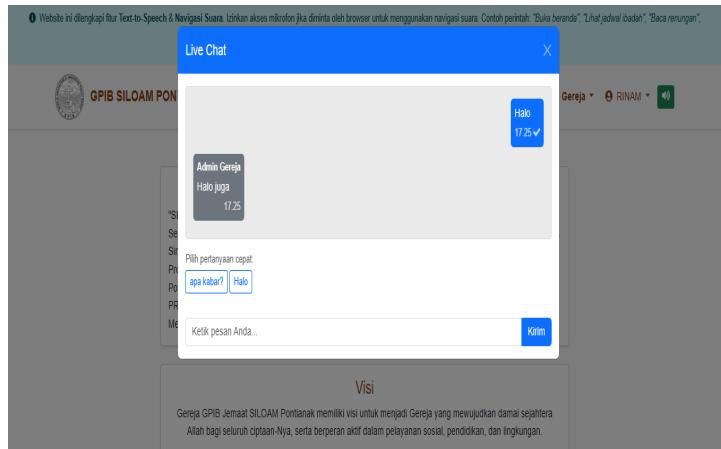
```

Gambar 11 Code Route

3.4 Demonstrasi

3.4.1 Interface System

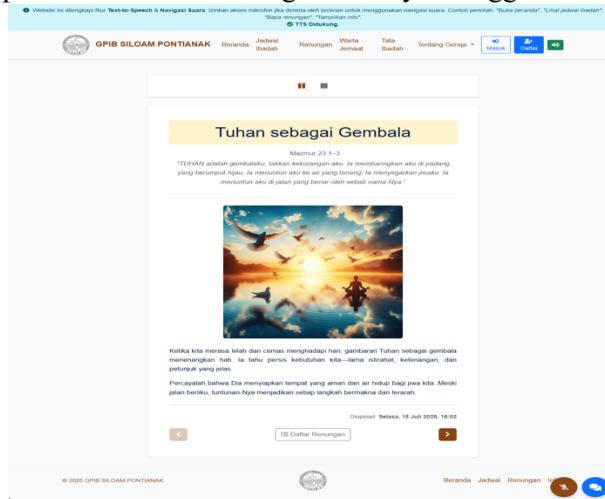
Tahapan ini akan membuat tampilan interface web dengan fokus pada tampilan atau gaya penulis untuk melampirkan tampilan interface yang menjadi penghubung langsung dengan pengguna yang terdapat pada website GPIB Siloam Pontianak. Berikut beberapa tampilan yang menjadi interface GPIB Siloam Pontianak.



Gambar 12 Tampilan Chat Jemaat

Gambar 12 menampilkan modal chat yang berfungsi sebagai sarana komunikasi langsung antara jemaat dan admin gereja. Modal yang berjudul live chat ini menampilkan riwayat percakapan, di mana pesan dari jemaat berada di sisi kanan dan balasan dari admin gereja di sisi kiri, keduanya dilengkapi dengan penanda waktu. Bagian bawahnya, terdapat fitur pilih

pertanyaan cepat yang menyarankan pertanyaan umum. Jemaat juga dapat mengetik pesan custom di kolom isian "Ketik pesan Anda..." dan mengirimkannya menggunakan tombol kirim.



Gambar 13 Implementasi Speechsynthesis

Evaluasi Implementasi fitur Text-to-Speech pada halaman renungan ini diwujudkan melalui tombol kontrol audio mainkan dan hentikan yang terletak tepat di bawah judul. Ketika jemaat menekan tombol mainkan, sistem secara otomatis menggunakan API SpeechSynthesis bawaan. Fitur ini, yang juga diinformasikan melalui notifikasi di bagian atas situs, dirancang untuk meningkatkan aksesibilitas, memungkinkan jemaat untuk mendengarkan firman Tuhan tanpa harus membacanya.

Pada penelitian ini menggunakan tindakan pengujian sebagai evaluasi sistem pada website real time GPIB Siloam Pontianak. Pengujian perangkat lunak White box didasarkan pada pemeriksaan ketat implementasi prosedural detail dan implementasi struktur data. Jalur logis melalui perangkat lunak dan kolaborasi antar komponen adalah fokus pengujian integrasi White box [8]. White-Box testing memiliki beberapa teknik dalam melakukan pengujian perangkat lunak, namun untuk pengujian kali ini akan digunakan White-Box testing dengan teknik Basis Path. Basic Path memungkinkan perancang test case untuk menghasilkan pengukuran kompleksitas logika dari perancangan sistem dan mengidentifikasi jalur dasar eksekusi.

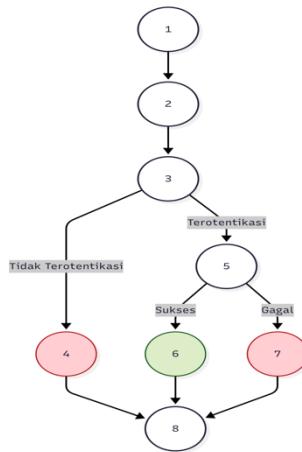
```

1 public function storeAdminMessage(Request $request)
2 {
3     $validated = $request->validate([
4         'message' => 'required|string|max:2000',
5         'target_user_id' => 'required|integer|exists:users,id',
6         'client_message_id' => 'nullable|string|max:36',
7     ]);
8
9     /** @var \App\Models\AdminUser|null $admin */
10    $admin = Auth::guard('admin_users')->user();
11
12    if (!$admin) {
13        return response()->json(['message' => 'Unauthorized (Admin)'], 401);
14    }
15
16    try {
17        $chat = $admin->sentChats()->create([
18            'user_id' => $validated['target_user_id'],
19            'message' => $validated['message'],
20        ]);
21
22        $chat->load('sender');
23
24        broadcast(new MessageSent($chat))->toOthers();
25
26        return response()->json([
27            'message' => 'Pesan admin berhasil terkirim',
28            'data' => new ChatResource($chat)
29        ], 201);
30    } catch (\Exception $e) {
31        Log::error("Error storing admin message from Admin ID {$admin->id} to User ID {$validated['target_user_id']} : " . $e->getMessage());
32        return response()->json(['message' => 'Gagal menyimpan pesan admin'], 500);
33    }
34 }

```

Gambar 14 Code Tambah Chat

Berdasarkan Gambar 14, dapat dibuat flowgraph fungsi chat. Adapun Gambar 13 akan memperlihatkan flowgraph fungsi ini.



Gambar 15 Flow Graph Tambah Chat

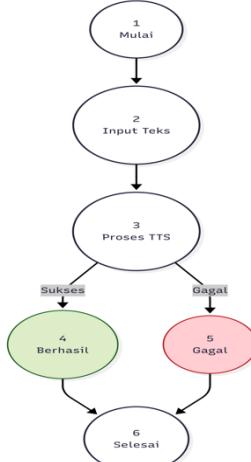
Pada gambar 15 diatas memiliki 3 (tiga) region yakni: Region pertama dimulai dari 1, 2, 3, 4, 8; region kedua dimulai dari 1, 2, 3, 5, 6, 8 dan region ketiga dimulai dari 1, 2, 3, 5, 7, 8. Proses ini adalah alur otentikasi. Mulai dari langkah awal (Node 1-3), proses akan bercabang. Jika otentikasi gagal, alur menuju Node 4. Jika otentikasi berhasil, alur berlanjut ke Node 5 untuk validasi, yang bisa berakhir sukses (Node 6) atau gagal (Node 7). Semua alur pada akhirnya berakhir di titik penyelesaian (Node 8).

```

1 const synthesisAvailable =
2   "speechSynthesis" in window && "SpeechSynthesisUtterance" in window;
3 const synth = synthesisAvailable ? window.speechSynthesis : null;
4 let voices = [];
5 window.speechEnabled = true;
6
7 let utteranceQueue = [];
8 let isSpeakingQueue = false;
9 let currentQueueUtterance = null;
10
11 const supportEl = document.getElementById("browserSupport");
12 const toggleBtn = document.getElementById("btnToggleSpeech");
13
14 function checkBrowserSupport() {
15   if (!supportEl) return;
16   let supportText = "";
17   if (synthesisAvailable) {
18     supportText += '<i class="fas fa-check-circle text-success me-1"></i>TTS Didukung. ';
19   } else {
20     supportText += '<i class="fas fa-times-circle text-danger me-1"></i>TTS Tidak Didukung. ';
21   }
22   window.speechEnabled = false;
23   if (toggleBtn) {
24     toggleBtn.disabled = true;
25     updateToggleButton(toggleBtn, false);
26   }
27 }
28 supportEl.innerHTML = supportText;
29
30
  
```

Gambar 16 Code Speechsynthesis

Berdasarkan Gambar 16, dapat dibuat flowgraph fungsi chat. Adapun Gambar 17 akan memperlihatkan flowgraph fungsi ini.



Gambar 17 Flow Graph Speechsynthesis

Pada Gambar 17 di atas terdapat 2 (dua) region: Region pertama 1, 2, 3, 4, 6 dan Region kedua 1, 2, 3, 5, 6. Flowgraph ini menggambarkan alur SpeechSynthesis (TTS): dari Mulai 1, sistem menerima/menangkap teks masukan 2, lalu memprosesnya menjadi suara 3. Bila pemrosesan sukses, alur menuju Berhasil 4 dan berakhir di Selesai 6. Bila pemrosesan gagalmisalnya browser tidak mendukung, voice tidak tersedia, atau terjadi error sintesis alur menuju Gagal 5 dan tetap berakhir di Selesai 6.

3.4.2 Analisis Performa WebSocket

Analisis performa WebSocket dilakukan untuk menguji ketahanan sistem ketika menerima beban koneksi yang tinggi secara simultan. Pengujian ini penting untuk memastikan bahwa fitur chat real-time tetap responsif pada kondisi aktivitas jemaat meningkat, khususnya saat ibadah online, pengumuman besar, atau acara gereja yang memicu interaksi digital lebih intens. Pengujian dilakukan menggunakan simulasi koneksi paralel sebanyak 20, 75, 150, hingga 250 klien untuk menggambarkan kondisi normal hingga kondisi stress-test. Parameter yang diukur meliputi latensi rata-rata (waktu respon pesan), tingkat packet loss, serta kestabilan koneksi.

Tabel 1 hasil pengujian WebSocket

| Kondisi Trafik | Jumlah Klien | Latensi Rata-Rata (ms) | Packet Loss (dalam %) | Status |
|-----------------------------|--------------|------------------------|-----------------------|--------------------|
| Normal | 20 | 56 | 0 | Stabil |
| Menengah | 75 | 84 | 0.30 | Stabil |
| Tinggi | 150 | 112 | 1.10 | Stabil |
| Sangat Tinggi (Stress Test) | 250 | 189 | 3.40 | Mulai tidak stabil |

Hasil pengujian menunjukkan bahwa komunikasi real-time bekerja dengan baik dalam kondisi normal maupun beban pengguna menengah. Waktu rata-rata pengiriman pesan berada pada kisaran 56 ms, dan tetap stabil hingga 112 ms ketika diuji dengan 150 koneksi simultan, yang menandakan bahwa sistem mampu menangani trafik jemaat yang cukup tinggi secara konsisten. Dengan demikian, WebSocket terbukti meningkatkan efektivitas pelayanan digital melalui fitur komunikasi gereja yang cepat, responsif, dan interaktif.

Hasil implementasi menunjukkan bahwa fitur SpeechSynthesis dapat memutar konten dengan tingkat keberhasilan 100% pada browser modern seperti Chrome, Edge, dan Firefox. Proses inisiasi suara sangat cepat, dengan waktu mulai bicara rata-rata kurang dari 1 detik, sehingga jemaat dapat langsung mendengarkan isi renungan tanpa hambatan. Fitur ini terbukti sangat membantu dalam memperluas aksesibilitas sistem dan meningkatkan kualitas pelayanan digital gereja secara keseluruhan.

Hasil pengujian menunjukkan bahwa WebSocket memiliki performa yang sangat baik dalam menangani trafik hingga 150 koneksi simultan. Pada kondisi ini, latensi masih berada pada kisaran 112 ms, yang masih tergolong cepat dan tidak mengganggu interaksi pengguna. Packet loss pada level ini juga sangat minim (1.1%), sehingga pesan tetap dapat terkirim dengan tingkat kegagalan yang sangat rendah.

Pada skenario beban sangat tinggi (250 koneksi), sistem mulai menunjukkan tanda-tanda ketidakstabilan. Latensi meningkat menjadi 189 ms dan packet loss mencapai 3.4%. Meskipun masih operasional, keterlambatan mulai terasa dan beberapa pesan harus dikirim ulang. Namun, kondisi ini berada di luar perkiraan trafik normal website gereja sehingga masih dianggap aman untuk kebutuhan pelayanan gereja.

Interpretasi keseluruhan menunjukkan bahwa arsitektur WebSocket yang diterapkan memiliki ketahanan yang baik dan dapat diandalkan untuk aktivitas jemaat dalam jumlah besar. Hasil pengujian memberikan bukti bahwa teknologi WebSocket merupakan solusi efektif untuk mendukung fitur komunikasi real-time dalam sistem gereja. Dibandingkan metode tradisional seperti AJAX polling atau refresh periodik, WebSocket mampu menjaga koneksi tetap aktif dan meminimalkan overhead jaringan.

4. KESIMPULAN

Masalah utama pada website GPIB Siloam Pontianak adalah kurangnya fitur interaksi dan aksesibilitas. Secara spesifik, tidak adanya fitur chat langsung telah menghambat komunikasi dua arah secara real-time antara jemaat dan gereja. Di sisi lain, ketiadaan fitur text-to-speech membuat konten sulit diakses oleh jemaat yang memiliki kendala membaca, seperti lansia atau mereka dengan keterbatasan visual, yang pada akhirnya membuat penyampaian informasi menjadi kurang efektif dan inklusif.

Sebagai solusi, penelitian ini mengimplementasikan dua teknologi utama pada website. Pertama, teknologi WebSocket digunakan untuk membangun fitur chat yang responsif, yang mencakup balasan otomatis untuk pertanyaan umum serta obrolan langsung dengan admin bagi jemaat yang sudah login. Kedua, teknologi SpeechSynthesis diintegrasikan untuk mengubah konten tekstual, seperti renungan dan pengumuman, menjadi format audio. Seluruh sistem ini dibangun di atas fondasi framework Laravel dan basis data MySQL untuk memastikan pengelolaan data yang terstruktur dan aman.

Dampak dari implementasi ini sangatlah signifikan dan positif. Fitur chat berbasis WebSocket berhasil membuat komunikasi menjadi jauh lebih cepat, responsif, dan interaktif, memungkinkan jemaat mendapatkan informasi secara instan. Sementara itu, fitur SpeechSynthesis berhasil meningkatkan aksesibilitas secara drastis, sejalan dengan visi pelayanan gereja yang lebih inklusif. Secara keseluruhan, pengembangan ini tidak hanya meningkatkan efisiensi penyampaian informasi tetapi juga berhasil memperkuat hubungan digital antara jemaat dan pengurus gereja.

5. SARAN

Saran untuk penelitian selanjutnya antara lain pengembangan fitur notifikasi push agar jemaat selalu mendapat pemberitahuan instan, integrasi kalender digital, maupun pengembangan aplikasi mobile khusus gereja untuk meningkatkan kemudahan akses.

UCAPAN TERIMA KASIH

Dengan penuh rasa syukur, peneliti mengucapkan terima kasih kepada pemilik Gereja GPIB Siloam Pontianak yang telah memberikan dukungan financial dan mengijinkan saya untuk meneliti dan menyelesaikan penelitian ini dengan sangat baik.

DAFTAR PUSTAKA

- [1] B. Owen, Y. Dwiandiyanta, and C. Anuryanti, “Pembangunan sistem informasi dengan visualisasi data pada Gereja Santa Maria Dengan Tidak Bernoda asal Tulungagung,” *Jurnal Informatika Atma Jogja*, vol. 3, no. 2, pp. 117–125, 2022.
- [2] R. Rachmatullah, B. Sumboro, and F. W. Setianingsih, “Sistem informasi administrasi Gereja Paroki St. Stephanus Jumapolo berbasis Android,” *Go Infotech: Jurnal Ilmiah STMIK AUB*, vol. 27, no. 1, pp. 130–144, 2021.
- [3] T. G. Sena, R. Pattiasina, N. G. Basatha, and Reinaldo, “Perancangan dan pembuatan website kuis daring menggunakan WebSocket communication protocol,” vol. 4, no. 1, pp. 100–101, 2024.
- [4] Sugiharto, V. W. Anshori, and A. Anshori, “Penggunaan artificial intelligence dalam mempersiapkan khutbah yang efektif,” *Jurnal Teologi*, vol. 8, no. 2, pp. 2–4, 2024.
- [5] S. Smith, *Responsive web development with HTML5 and CSS: Building modern and user-friendly websites for all devices*. 2023.

- [6] M. A. A. Dzakwan, “Pemanfaatan teknologi WebSocket pada pengiriman data IoT berbasis realtime scenario,” Jember, Indonesia, 2024.
 - [7] X. Tan, Neural text-to-speech synthesis. Beijing, China: Springer, 2023.
 - [8] M. Stauffer, Laravel: Up & running: A framework for building modern PHP apps. Sebastopol, CA, USA: O'Reilly Media, 2019.
 - [9] J. Y. Barigas, “Sistem informasi gereja berbasis web menggunakan framework Laravel,” Jurnal Informatika dan Teknologi Informasi, vol. 2, no. 1, pp. 80–100, 2021.
 - [10] R. Basatha and B. B. B. Keraf, “Analisis dan desain sistem informasi berbasis website Gereja Katolik Santo Yusup Jember,” Jurnal Teknologi dan Sistem Informasi, vol. 2, no. 1, pp. 80–91, 2022.
 - [11] A. Hevner and A. Maedche, Design science research: Cases. Cham, Switzerland: Springer Nature Switzerland AG, 2020.
 - [12] J. vom Brocke, A. Hevner, and A. Maedche, Design science research: Cases. Cham, Switzerland: Springer Nature Switzerland AG, 2020.
 - [13] R. S. Pressman and B. R. Maxim, Software engineering: A practitioner's approach. New York, NY, USA: McGraw-Hill, 2020.
 - [14] A. Gunawan, S. Ningsih, and D. A. Lantana, Pengantar basis data. Jakarta, Indonesia: PT Literasi Nusantara Abadi Grup, 2023.
 - [15] A. D. Sole, Visual Studio Code distilled: Evolved code editing for Windows, macOS, and Linux, 3rd ed. Italy: Apress, 2023.
-