

INTEGRASI TEKNOLOGI NUXTJS DAN LARAVEL 12 BERBASIS WEB SERVICE PADA APLIKASI ADMINISTRASI (STUDI KASUS : GPDI SAHABAT ALLAH PALEMBANG)

David Gultom, Hendrik Fery Herdiatmoko

Informatika, Universitas Katolik Musi Charitas

Jl. Bangau No.60, 9 Ilir, Kec. Ilir Tim. II, Kota Palembang, Sumatera Selatan 30114

davidgultom669@gmail.com

ABSTRAK

Digitalisasi telah mendorong banyak industri, termasuk lembaga keagamaan, untuk beralih dari sistem administrasi konvensional ke sistem yang lebih efisien berbasis teknologi. Untuk menangani permohonan administrasi, GPDI Sahabat Allah Palembang masih menggunakan proses manual, sehingga jemaat yang tinggal jauh dari gereja harus hadir langsung untuk mengurus surat-menyurat. Hal ini menyebabkan keterlambatan dan kesalahan pencatatan. Tujuan penelitian ini adalah untuk membuat dan membangun sistem administrasi digital yang menghubungkan jemaat dan kantor gereja secara *online* melalui *web service*. Metode *Waterfall* digunakan untuk pengembangan, yang mengumpulkan data melalui observasi, wawancara, dan penelitian literatur. Metode *Black Box* juga digunakan untuk menguji sistem. Hasil penelitian menunjukkan bahwa sistem dibangun menggunakan integrasi *framework Laravel 12* sebagai *backend* dan *NuxtJS* sebagai *frontend*. Mereka berkomunikasi melalui *REST API* berformat *JSON* dan diamankan dengan autentikasi *JWT*. Fitur pengajuan surat digital, validasi oleh sekretariat, pemantauan status permohonan secara *real-time*, dan pengelolaan konten informasi publik gereja mempercepat proses administrasi.

Kata kunci : *Web Service, REST API, Laravel 12, NuxtJS, Administrasi Digital.*

1. PENDAHULUAN

Perkembangan teknologi yang semakin cepat telah mendorong berbagai sektor untuk memanfaatkan digitalisasi dalam mendukung aktivitas sehari-hari, khususnya dalam pengelolaan data. Digitalisasi terbukti mampu mempercepat administrasi, mengoptimalkan pemrosesan data, meminimalkan kesalahan, serta memperluas akses terhadap layanan penting. Namun, untuk sepenuhnya memanfaatkan potensi digitalisasi, diperlukan kebijakan dan mekanisme dukungan yang tepat guna mengatasi tantangan yang muncul dalam penerapan teknologi dan layanan digital[1].

Dalam berbagai bidang seperti bisnis, pendidikan, pemerintahan, maupun organisasi keagamaan, surat-menyurat dan permohonan dokumen merupakan bagian yang tidak terpisahkan dari proses administrasi. Pelayanan publik yang baik harus mampu memberikan kemudahan dan efisiensi waktu bagi semua pihak yang terlibat[2]. Sebagai contoh, pengembangan layanan berbasis web pada sistem administrasi gereja Katolik MRPD terbukti membantu mengatasi permasalahan administrasi dan layanan umat secara signifikan[3].

GPDI Sahabat Allah Palembang saat ini masih menerapkan metode konvensional dalam menangani permohonan administrasi. Jemaat yang tinggal jauh dari gereja menghadapi kendala karena harus datang langsung untuk mengajukan permohonan surat atau dokumen. Kondisi ini mengakibatkan banyak surat-menyurat mengalami keterlambatan, proses pelayanan menjadi tidak efisien, dan risiko kesalahan administrasi semakin tinggi. Selain itu, tidak adanya

sistem pencatatan digital menyebabkan dokumentasi sulit dikelola secara terorganisir[4].

Tujuan penelitian ini adalah merancang dan membangun sistem layanan administrasi berbasis *website* guna meningkatkan efektivitas dan efisiensi proses administrasi di GPDI Sahabat Allah Palembang. Sistem yang dikembangkan diharapkan dapat mempermudah akses jemaat dalam mengajukan permohonan administrasi kapan saja tanpa harus hadir langsung ke gereja, mempercepat proses pengelolaan permohonan oleh pihak gereja, serta menjamin dokumentasi yang rapi dan terorganisir.

Untuk mengatasi permasalahan tersebut, penelitian ini mengusulkan pembangunan sistem berbasis *web service* menggunakan integrasi *framework NuxtJs* dan *Laravel 12*. Sistem terdiri dari dua website yang saling terhubung: satu portal pengajuan permohonan oleh jemaat, dan satu sistem pengelolaan oleh pihak gereja. Kedua sistem berkomunikasi melalui *REST API*, di mana data respons disimpan dalam format *JSON*[6][7]. Pendekatan *web service* dipilih karena mampu melakukan otomatisasi permohonan, verifikasi, dan penyimpanan data, sekaligus berfungsi sebagai penghubung antara berbagai sistem tanpa bergantung pada *platform* atau bahasa pemrograman tertentu [5]. Dengan memanfaatkan *framework Laravel* dan *NuxtJs*, sistem dikembangkan agar terstruktur, fleksibel, dan mudah dikembangkan lebih lanjut, sehingga dapat mendukung layanan administrasi gereja secara menyeluruh dan berkelanjutan.

Penelitian ini mengangkat judul "Integrasi *NuxtJs* dan *Laravel 12* Berbasis *Web Service* pada

Sistem Permohonan Administrasi di GPdI Sahabat Allah Palembang", dengan harapan sistem ini dapat membantu jemaat dan pengurus gereja dalam mengajukan serta mengelola permohonan administrasi secara lebih cepat, kapan saja, dan dengan risiko kesalahan yang minimal.

2. TINJAUAN PUSTAKA

Kajian terhadap penelitian sebelumnya merupakan fondasi penting untuk memetakan perkembangan teknologi terbaru dan memastikan bahwa karya tersebut unik. Penelitian yang relevan tentang *Web service* telah membahas optimalisasi arsitektur *RESTful*, kemandirian format pertukaran data seperti *JSON*, dan penerapan protokol keamanan sistem. Untuk memastikan integrasi data yang aman dan cepat antar platform yang berbeda, fokus utama penelitian tersebut adalah meningkatkan skalabilitas dan interoperabilitas layanan.

2.1. Tinjauan Pustaka

Penelitian pertama menunjukkan bahwa implementasi *web service* dengan arsitektur *RESTful API* pada sistem layanan Gereja Katolik Paroki MRPD Pontianak berhasil mengoptimalkan integrasi data yang sebelumnya dilakukan secara manual. Melalui metodologi *Action Research* berbasis *Extreme Programming* dan *framework CodeIgniter*, sistem ini mempermudah akses informasi serta pengelolaan data sekretariat. Secara keseluruhan, karakteristik *REST* yang *stateless* dan *cacheable* terbukti menjadi solusi efektif untuk membangun layanan web yang ringan, *interoperabel*, dan fleksibel di berbagai platform.

Penelitian Wicaksonoberfokus pada pembentukan "Ayo BerAksi", sebuah sistem untuk melaporkan dan mengadukan korupsi di Balai Besar Karantina Pertanian Belawan. Studi ini berhasil mendigitalisasi proses pelaporan konvensional yang sebelumnya memakan waktu lama menjadi sebuah sistem multiplatform yang lebih efektif dengan menggunakan arsitektur *Representational State Transfer (REST) API* dan *framework Laravel*. Hasil pengujian menunjukkan bahwa penggunaan *REST API* dalam format data *JSON* dapat meningkatkan penggunaan bandwidth dan mempercepat pertukaran data antara aplikasi *mobile* dan berbasis *website*[6].

Studi yang ditulis oleh Hendrik Fery Herdiyatomoko mempelajari bagaimana membuat sistem backend berbasis *REST API* dengan *framework Laravel 7*. Membangun *REST Server* yang mendukung keamanan, otomatisasi dengan fitur *Artisan*, dan kemudahan penggunaan paket dalam arsitektur *Model-View-Controller (MVC)* adalah fokus utama penelitian ini. Hasil penelitian menunjukkan bahwa *Laravel* dapat digunakan dengan sangat baik untuk menjalankan fungsi dasar *CRUD (Create, Read, Update, Delete)* dengan format pertukaran data *JSON*. Ini memungkinkan klien *REST* untuk berkomunikasi dengan data dengan aman dan efisien[8].

Arianto dan Susetyo membangun sistem informasi manajemen SDM dengan menggunakan *RESTful Web service* dan *framework Laravel*. Studi ini menekankan betapa pentingnya integrasi data terpusat dalam sebuah organisasi untuk mempercepat proses penyebaran data dan mengurangi jumlah data yang tidak diperlukan. Kolaborasi dengan *Library Laravel Passport* meningkatkan keamanan pertukaran data dalam sistem ini. Ini menjamin bahwa informasi tetap aman selama proses transaksi data antar sistem[7].

Studi yang dilakukan oleh Baharuddin, menyelidiki penggunaan *web service* yang menggunakan metode *REST API* untuk integrasi data *COVID-19* di Sulawesi Selatan. Tujuan utama penelitian ini adalah menciptakan jembatan yang memungkinkan berbagai sistem dan platform berkomunikasi satu sama lain tanpa terhubung secara langsung. Studi ini menunjukkan bahwa penggunaan arsitektur *RESTful* memudahkan manajemen data yang dinamis dan menjamin bahwa informasi yang akurat tersedia bagi masyarakat melalui sinkronisasi data yang cepat antar *platform website*[9].

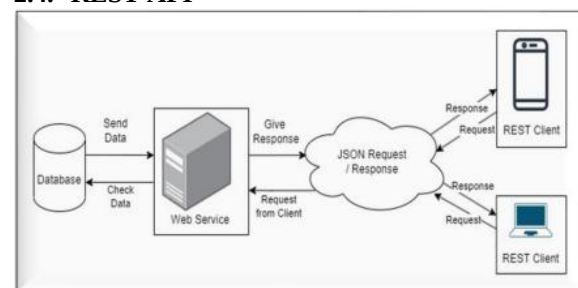
2.2. Administrasi

Tujuan administrasi adalah untuk meningkatkan efisiensi dan efektivitas penyelenggaraan kegiatan operasional pendidikan dalam mencapai tujuan tertentu. Administrasi adalah rangkaian kegiatan yang direncanakan dan sistematis yang melibatkan dan mengikutsertakan sumber-sumber potensial yang ada, baik sarana dan prasarana maupun manusia, untuk mencapai tujuan dengan lebih efektif dan efisien[10].

2.3. Web service

Aplikasi lintas *platform* yang dikenal sebagai *Web service* dapat diakses melalui jaringan, seperti intranet dan internet, dan memiliki berbagai fungsi yang memungkinkan aplikasi berinteraksi satu sama lain. Selain itu, *web service* dapat dianggap sebagai metode pertukaran data. Ini berlaku tanpa memperhitungkan lokasi *database*, bahasa yang digunakan, platform yang digunakan, dan lima aplikasi yang menggunakan data. Dengan memungkinkan *interoperabilitas*, layanan web dapat berfungsi sebagai jembatan yang menghubungkan berbagai sistem yang sudah ada[11].

2.4. REST API



Gambar 1. Proses kerja *REST client* dan *REST Server*

REST Web service diciptakan oleh Roy Fielding pada tahun 2002 sebagai alternatif untuk pengembangan *web service*, dengan format output *JSON*. Salah satu manfaat *REST Web service* adalah mempermudah pengembang untuk melakukan integrasi data melalui operasi database menggunakan protokol *HTTP*, sementara dalam operasi database konvensional dikenal perintah *Create*, *Read*, *Update*, dan *Delete*[12].

2.5. Respon Status Code

Tabel 1. Respon Status Code

Kategori	Deskripsi
100	Mengkomunikasikan informasi pada level protokol transfer.
200	Menunjukkan bahwa permintaan dari klien telah diterima dengan sukses.
300	Menunjukkan bahwa klien harus melakukan tindakan tambahan untuk menyelesaikan permintaannya.
400	Kategori kesalahan ini (kode status) menunjukkan bahwa kesalahan berasal dari pihak klien.
500	Server bertanggung jawab atas kesalahan ini (kode status).

REST API menggunakan bagian *Status-Line* dari pesan respons *HTTP* untuk memberi tahu klien tentang hasil menyeluruh dari permintaan mereka. *RFC 2616* mendefinisikan sintaks *Status-Line* seperti yang ditunjukkan di bawah ini:

Status-Line = *HTTP-Version SP Status-Code SP Alasan-Frasa CRLF*

HTTP mendefinisikan empat puluh kode status standar yang dapat digunakan untuk menyampaikan hasil permintaan klien[13].

2.6. JSON

JSON menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C, seperti C++, C#, Java, JavaScript, Perl, Python, dll., format teks ini tidak bergantung pada bahasa pemrograman apa pun. *JSON* adalah bahasa pertukaran data yang sempurna karena karakteristiknya. *JSON* terdiri dari dua struktur yang berbeda: kumpulan pasangan nama/nilai. Beberapa bahasa menunjukkannya sebagai objek, rekaman, struktur, kamus, tabel hash, daftar berkunci, atau daftar nilai terurutkan. Dalam kebanyakan bahasa, ini ditunjukkan sebagai *array*, vektor, daftar, atau urutan[14].

2.7. Laravel

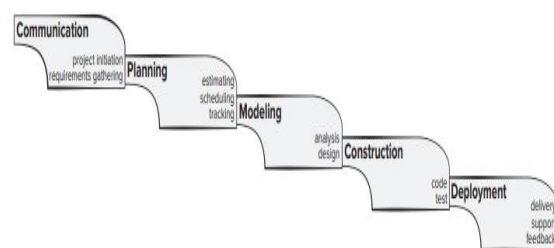
Laravel adalah sebuah *framework* aplikasi web berbasis *PHP* yang bertujuan untuk mempermudah pengembangan aplikasi web dengan menyediakan berbagai fitur dan alat yang kuat. Salah satu fitur utama *Laravel* adalah kemampuan untuk membuat pengelolaan basis data lebih mudah. *Laravel* menyediakan *ORM (Object-Relational Mapping)* yang disebut *Eloquent*, yang memungkinkan pengembang

berinteraksi dengan basis data menggunakan sintaks yang mudah dipahami tanpa harus menulis kueri *SQL* secara langsung. Selain itu, *Laravel* memiliki sistem migrasi yang membuat pengaturan basis data menjadi lebih mudah[15].

2.8. Nuxt.Js

Nuxt.js adalah pustaka *JavaScript* sumber terbuka yang dibangun di atas *Vue.js* dan menggunakan *Node.js*, *Webpack*, dan *Babel.js* di bawahnya. Ini mengambil inspirasi dari *Next.js*, sebuah kerangka kerja dengan tujuan yang sama yang didasarkan pada *React.js*. Tujuan *Nuxt.js* adalah untuk membantu pengembang membuat dan mengoptimalkan aplikasi *Vue.js*[16].

3. METODE PENELITIAN



Gambar 2. Model waterfall

Model *Waterfall* dipilih karena sifatnya yang sistematis, linear, dan terdokumentasi dengan baik, sehingga cocok untuk proyek dengan spesifikasi yang jelas sejak awal. Dalam penelitian ini, metode tersebut menjamin proses pengembangan integrasi *Laravel 12* dan *NuxtJS* berjalan terencana mulai dari perancangan hingga implementasi. Dengan menyelesaikan setiap tahapan secara berurutan, kualitas sistem administrasi yang dihasilkan tetap konsisten dan sesuai dengan tujuan penelitian.[17].

3.1. Metode Pengumpulan Data

a. Metode Observasi

Pengamatan langsung proses administrasi di GPdI Sahabat Allah Palembang memberikan sumber data. Tujuan observasi ini adalah untuk meningkatkan pemahaman tentang proses sistem, cara pengguna berkomunikasi dengan petugas administrasi, dan masalah yang mungkin muncul selama proses pengajuan administrasi.

b. Studi Literatur

Pada titik ini, penulis mencari dan mempelajari beberapa referensi dari buku, jurnal, dan skripsi sebelumnya yang membahas topik serupa, terutama yang berkaitan dengan sistem permohonan administrasi.

c. Wawancara

Pengumpulan data juga dilakukan dengan cara berkomunikasi secara langsung kepada pihak pengurus GPdI Sahabat Allah Palembang untuk kebutuhan informasi yang lebih mendalam.

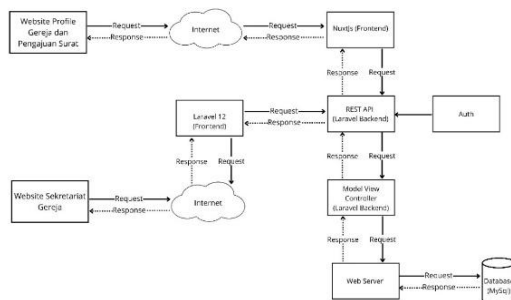
3.2. Metode Perancangan

Penulis menggunakan alat bantu *UML (Unified Modeling Language)* untuk melakukan analisis rancangan sistem informasi. *UML* adalah bahasa berbasis grafik yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mencatat perkembangan sistem piranti lunak berbasis pemrograman berorientasi objek. *Use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* adalah contoh tahapan yang dilalui.

3.3. Metode Pengujian

Pengujian *Black Box* adalah metode uji perangkat lunak yang berfokus sepenuhnya pada fungsionalitas eksternal dari sudut pandang pengguna. Pengujian ini dilakukan dengan memeriksa kecocokan antara *input* dan *output* berdasarkan skenario yang telah dirancang, tanpa perlu mengetahui logika atau struktur kode internal program. Tujuannya adalah memastikan sistem beroperasi sesuai dengan kebutuhan pengguna.

3.4. Arsitektur Sistem



Gambar 3. Arsitektur sistem

Dengan menggunakan arsitektur *REST API*, sistem ini mengintegrasikan *Website Sekretariat (Laravel 12)* sebagai *backend* dan *Website Profil & Pengajuan Surat (NuxtJS)* sebagai *frontend*. Pola desain *MVC* dan *database MySQL* membuat pengorganisasian data lebih efisien dan terstruktur. Autentikasi *JSON Web Token (JWT)* menjamin bahwa setiap akses terenkripsi, memastikan bahwa pertukaran data aman. Dengan implementasi ini, jemaat dan sekretariat dapat sinkronisasi data secara real-time. Ini juga membuat ekosistem administrasi digital yang tangguh dan responsif untuk pelayanan gereja.

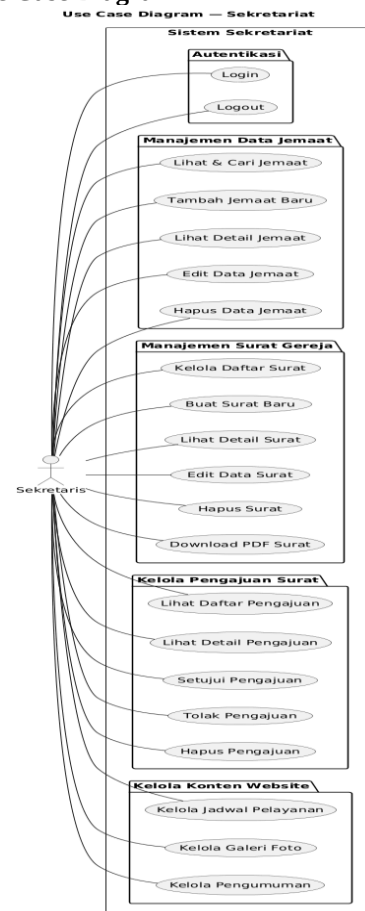
4. HASIL DAN PEMBAHASAN

4.1. Rancangan Program

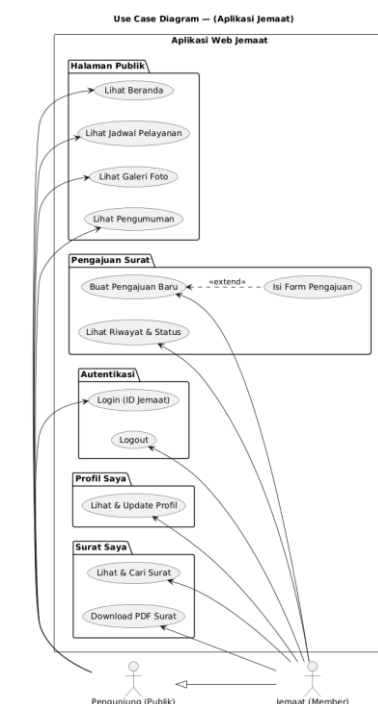
Rekomendasi prosedur dibuat dengan tujuan meningkatkan efisiensi dan meningkatkan berbagai aspek fungsionalitas sistem yang sedang berjalan saat ini. Untuk menggambarkan alur kerja dan struktur data secara sistematis, perangkat lunak *draw.io* digunakan untuk memodelkan sistem yang direkomendasikan secara visual menggunakan diagram *UML*. Perangkat lunak ini dikembangkan pada tahap implementasi dengan menggabungkan bahasa pemrograman *PHP* dan *JavaScript*. Selain itu, ekosistem pengembangan

lokal *Laragon* digunakan sebagai pengelola sistem manajemen *database* untuk memastikan kinerja dan stabilitas aplikasi.

3.5. Use Case Diagram



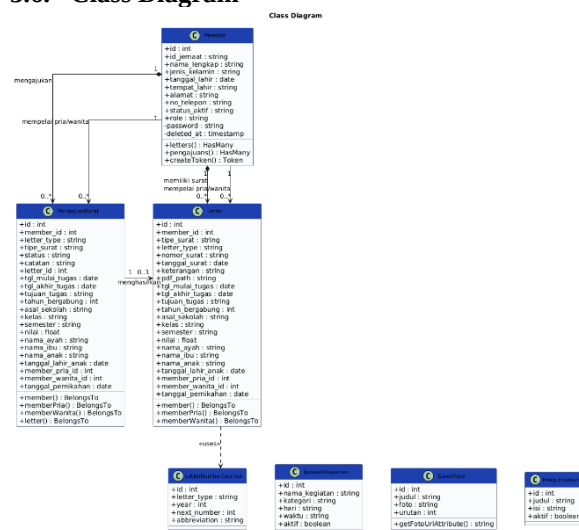
Gambar 4. Use case diagram sekretariat gereja



Gambar 5. Use case diagram jemaat

Diagram use case pada sistem portal administrasi digital berbasis *Laravel* ini melibatkan dua aktor utama, yaitu Jemaat dan Sekretariat. Aktor Jemaat berfokus pada layanan mandiri yang mencakup login, pendaftaran, pengajuan berbagai jenis surat, serta pemantauan status dokumen secara *real-time*. Sementara itu, aktor Sekretariat memiliki kendali manajerial untuk mengelola data jemaat, menginput surat masuk, dan memberikan persetujuan (*approval*) atas pengajuan surat. Melalui integrasi berbagai fitur ini, sistem diharapkan mampu menyatukan data yang sebelumnya terfragmentasi ke dalam satu platform yang terpusat, aman, dan efisien, sehingga secara signifikan meningkatkan pengalaman pengguna dalam mengelola urusan administratif.

3.6. Class Diagram



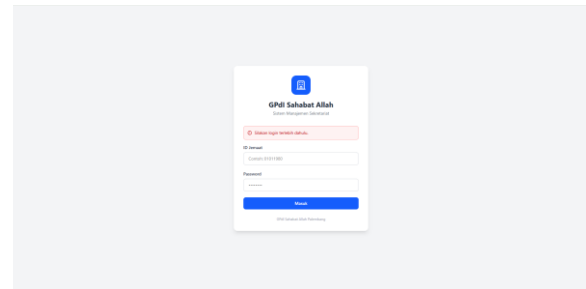
Gambar 6. Class diagram

Sistem portal administrasi berbasis *Laravel* ini memiliki diagram kelas yang menggambarkan struktur data melalui enam kelas utama yang saling terintegrasi. Sebagai aktor utama sistem, kelas Jemaat memiliki hubungan *one-to-one* dengan kelas User. Ini menunjukkan hubungan antar entitas. Selanjutnya, kelas Jemaat berhubungan dengan kelas Pengajuan_Surat untuk menangani permohonan dokumen; kelas Surat_Masuk mengawasi arsip surat yang diterima. Selain itu, pada halaman publik terdapat kelas berita dan galeri yang berfungsi untuk menyediakan konten informasi dan dokumentasi. Pemetaan hubungan yang jelas antara kelas ini memungkinkan tim pengembang untuk memahami secara menyeluruh logika program, yang memungkinkan implementasi sistem dilakukan dengan lebih sistematis, efisien, dan aman untuk meningkatkan kualitas pelayanan informasi secara keseluruhan.

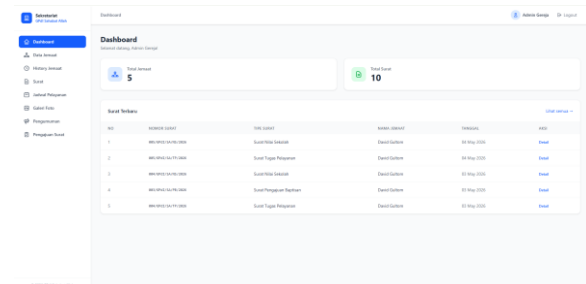
3.7. Tampilan Website

Halaman *login* ini berfungsi sebagai gerbang autentikasi utama yang mewajibkan setiap pengguna untuk memasukkan ID dan kata sandi yang valid guna

memverifikasi identitas, menjaga kerahasiaan data pribadi, serta memastikan hak akses terbatas hanya diberikan kepada pihak yang berwenang untuk masuk ke dalam sistem.

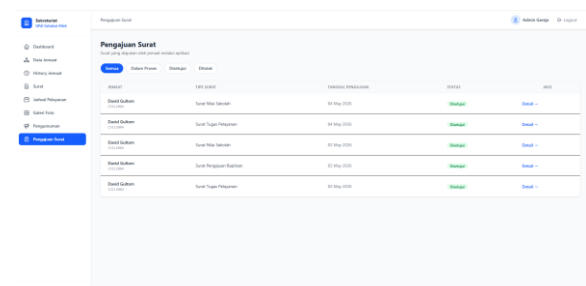


Gambar 7. Halaman *login* sekretariat

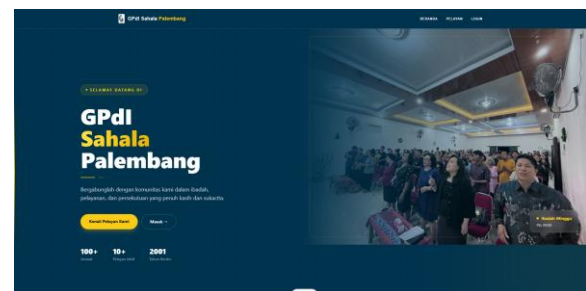


Gambar 8. Halaman *dashboard* sekretariat

Halaman *dashboard* sekretariat berfungsi sebagai pusat kendali utama yang menyajikan ringkasan data secara komprehensif, mulai dari total keseluruhan jemaat yang terdaftar, jumlah akumulasi surat yang dikelola, hingga daftar riwayat surat terbaru untuk memudahkan pemantauan aktivitas administrasi secara *real-time*.



Gambar 9. Halaman pengajuan surat sekretariat

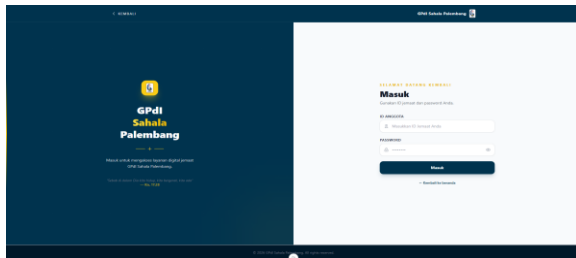


Gambar 10. Halaman beranda jemaat

Halaman Pengajuan Surat pada sisi Sekretariat berfungsi sebagai pusat validasi administratif yang

memungkinkan staf untuk meninjau secara saksama, kemudian mengambil keputusan untuk menyetujui atau menolak setiap permohonan surat yang diajukan oleh jemaat berdasarkan kelengkapan data yang tersedia.

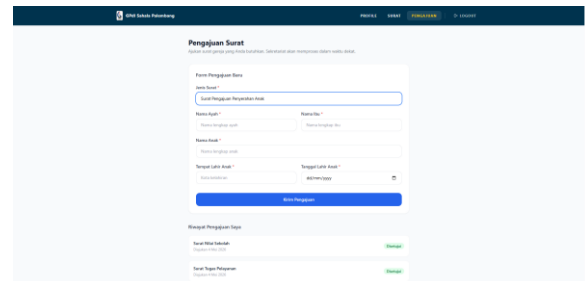
Halaman beranda ini dirancang sebagai pusat informasi publik yang menyajikan beragam konten esensial secara informatif, meliputi jadwal pelayanan rutin, pengumuman penting terkini, galeri dokumentasi kegiatan, hingga profil mendalam mengenai visi misi, struktur majelis, serta titik lokasi strategis gereja.



Gambar 11. Halaman login jemaat

Halaman login jemaat ini berfungsi sebagai sistem autentikasi independen, memaksa anggota untuk memasukkan ID dan kata sandi yang mereka miliki untuk memastikan bahwa mereka adalah pengguna yang sebenarnya. Sistem ini tidak hanya

berfungsi sebagai pintu utama untuk mengakses berbagai fitur unggulan, seperti pengajuan surat digital, pemantauan status permohonan secara real-time, dan penyebaran informasi pelayanan pada halaman utama secara terintegrasi dan personal, tetapi juga berperan penting dalam melindungi privasi dan kerahasiaan data jemaat dari pihak yang tidak diinginkan.



Gambar 12. Halaman pengajuan surat jemaat

Halaman Pengajuan Surat merupakan layanan mandiri bagi jemaat untuk memohon dokumen secara digital dengan mengisi formulir secara akurat. Sistem secara otomatis akan mengubah status menjadi "sedang diproses" agar jemaat dapat memantau progresnya secara real-time. Setelah mendapat persetujuan sekretariat, jemaat dapat langsung mengunduh dokumen tersebut melalui platform.

4.2. Modul Autentikasi & Akun

Tabel 2. Autentikasi & akun

Method	Endpoint	Auth	Keterangan
POST	/api/auth/login	-	Proses login jemaat/admin
DEL	/api/me/logout	member/admin	Proses keluar dan hapus token
GET	/api/me	member/admin	Menampilkan profil pengguna saat ini
PUT	/api/me	member/admin	Memperbarui data profil mandiri

Dokumentasi Autentikasi & Akun berfungsi mendapatkan token yang dipakai untuk mengakses sistem lainnya

4.3. Modul Layanan Mandiri Jemaat

Tabel 3. Layanan mandiri jemaat

Method	Endpoint	Auth	Keterangan
GET	/api/me/letters	member/admin	Melihat daftar surat yang sudah terbit
GET	/api/me/letters/{id}/download	member/admin	Mengunduh file PDF surat pribadi
GET	/api/me/pengajuan	member/admin	Melihat riwayat pengajuan surat
POST	/api/me/pengajuan	member/admin	Mengirim formulir pengajuan surat baru

Modul ini dikhususkan bagi jemaat untuk melakukan pengajuan administratif dan mengakses dokumen pribadi.

4.4. Modul Manajemen Administrasi

Tabel 4. Manajemen administrasi

Method	Endpoint	Auth	Keterangan
GET/POST	/api/admin/members	admin	Lihat daftar dan tambah data jemaat
GET/PUT/DEL	/api/admin/members/{id}	admin	Detail, update, dan hapus data jemaat
GET/POST	/api/admin/letters	admin	Kelola database arsip surat
GET/PUT/DEL	/api/admin/letters/{id}	admin	Detail, update, dan hapus arsip surat
GET	/api/admin/letters/{id}/pdf	admin	Cetak atau generate file PDF surat

Method	Endpoint	Auth	Keterangan
GET/PUT/DEL	/api/admin/pengajuan/{id}	admin	Review, update, dan hapus data pengajuan
PUT	/api/admin/pengajuan/{id}/setujui	admin	Validasi persetujuan pengajuan surat
PUT	/api/admin/pengajuan/{id}/tolak	admin	Validasi penolakan pengajuan surat

Modul ini memberikan kendali penuh kepada admin untuk mengelola database jemaat dan validasi dokumen.

4.5. Modul Konten Publik

Tabel 5. Konten publik

Method	Endpoint	Auth	Keterangan
GET	/api/public/jadwal	-	Menampilkan jadwal ibadah aktif
GET	/api/public/galeri	-	Menampilkan koleksi foto kegiatan
GET	/api/public/pengumuman	-	Menampilkan informasi pengumuman aktif
GET	/api/health	-	Pengecekan status operasional server

Modul ini menyediakan data informasi umum untuk halaman beranda dan pengecekan teknis sistem.

4.6. Modul Manajemen Konten Admin

Tabel 6. Manajemen konten admin

Method	Endpoint	Auth	Keterangan
GET/POST	/api/admin/jadwal	admin	Kelola data jadwal pelayanan
PUT/DEL	/api/admin/jadwal/{id}	admin	Update dan hapus jadwal
GET/POST	/api/admin/galeri	admin	Lihat daftar dan upload foto galeri
DEL	/api/admin/galeri/{id}	admin	Menghapus dokumentasi foto
GET/POST	/api/admin/pengumuman	admin	Kelola informasi pengumuman
PUT/DEL	/api/admin/pengumuman/{id}	admin	Update dan hapus pengumuman

Modul ini digunakan admin untuk memperbarui informasi yang ditampilkan pada halaman publik.

4.7. Pengujian

Tabel 7. Login jemaat

URL	http://localhost:8000/api/auth/login
Method	POST
Parameter	Id jemaat, password
Success Response	<pre>{ "status": "success", "message": "Login berhasil", "data": { "member": { "id": 6, "id_jemaat": "23112004", "nama_lengkap": "David Gultom", "jenis_kelamin": "Laki-laki", "tanggal_lahir": "2004-11-23", "tempat_lahir": "Palembang", "alamat": "Jl. Soak Simpung Lr. Sahabat ndol No. 16", "no_telepon": "082188252205", "status_aktif": "Aktif", "role": "member", "created_at": "2026-04- BT15:28:49+00:00", "updated_at": "2026-05- T12:56:55+00:00" }, "token": 0 irkMdZdbpDn4lpqgHkhHgS27zVMImSQ yS88Yryd2e2dc8d" } }</pre>

Tabel menunjukkan bahwa modul layanan jemaat untuk akses profil dan autentikasi mengembalikan respons JSON secara akurat. Identitas jemaat seperti nama, email, dan peran pengguna serta status keberhasilan (isSuccess, statusCode, dan responseMessage) disajikan. Hasilnya menunjukkan bahwa menambahkan web service REST ke aplikasi telah memenuhi kebutuhan jemaat yang terdaftar di GPdI Sahabat Allah Palembang.

Tabel 7. Pengajuan surat

URL	http://localhost:8000/api/me/pengajuan
Method	POST
Parameter	token
Success Response	<pre>"status": "success", "message": "Surat berhasil dibuat", "data": { "id": 9, "member_id": "10", "member": { "id": 10, "id_jemaat": "16071995", "nama_lengkap": "Rohana Updated", "tempat_lahir": "Surabaya", "tanggal_lahir": "1995-07-16", "alamat": "Jl. Merdeka No. 5, alembang", "no_telepon": "082233445566", "jenis_kelamin": "Laki-laki" } }</pre>

<pre> }, "tipe_surat": "Surat Tugas Pelayanan", "letter_type": "surat_tugas_pelayanan", "nomor_surat": 03/GPdI/SA/TP/2026", "tanggal_surat": "2026-04-30", "keterangan": null, "tgl_mulai_tugas": "2026-05-01", "tgl_akhir_tugas": "2026-05-07", "tujuan_tugas": "Pelayanan kebangunan hani di Jl. Merdeka Palembang", "has_pdf": false, "pdf_url": null, "created_at": "2026-04- 0T03:38:59+00:00", "updated_at": "2026-04- 0T03:38:59+00:00" } </pre>

Berdasarkan tabel tersebut, modul administrasi sekretariat untuk fitur manajemen data jemaat telah berfungsi dengan baik. Pengujian menunjukkan bahwa web service REST berhasil menyajikan respons JSON yang akurat, mencakup status operasional (isSuccess, statusCode, responseMessage) serta detail data jemaat lengkap seperti ID, Nama, Email, NIK, dan Alamat untuk keperluan pengelolaan administratif.

5. KESIMPULAN DAN SARAN

Dengan mengintegrasikan Laravel 12 sebagai backend dan NuxtJS sebagai frontend, penelitian ini berhasil membangun sistem administrasi digital untuk GPdI Sahabat Allah Palembang. Semua ini terhubung melalui REST API berformat JSON dan diamankan menggunakan JWT. Ini memungkinkan jemaat untuk mengajukan surat digital kapan saja tanpa harus pergi ke gereja, dan sekretariat dapat dengan mudah mengawasi dan memvalidasi pengajuan. Pengembangan sistem selanjutnya disarankan memiliki fitur notifikasi real-time, pengujian keamanan yang lebih menyeluruh, dan mungkin juga mengembangkan aplikasi mobile agar layanan dapat diakses oleh lebih banyak anggota jemaat.

DAFTAR PUSTAKA

- [1] A. Sarjito, "Dampak Digitalisasi Administrasi Perdesaan di Negara Berkembang," *Jurnal Ilmiah Ilmu Administrasi*, vol. 13, no. 2, pp. 106–124, Dec. 2023, doi: 10.33592/jiia.v13i2.3814.
- [2] D. R. R. R. Adriana Eka Prayudha, "perancangan sistem informasi pelayanan publik menggunakan framework laravel & mysql di kecamatan coblong kota bandung," 2020.
- [3] Glovenia Sanca Frisca, Sandy Kosasi, Tony Wijaya, Robertus Laipaka, and David, "Pemanfaatan web service dalam sistem layanan gereja katolik paroki mprd," *Jurnal Ilmiah Nasional Riset Aplikasi dan Teknik Informatika*, vol. 05, Jun. 2023, doi: <https://doi.org/10.53580/naratif.v5i1.194>.
- [4] D. R. Salsabilla, "implementasi rest-api untuk sistem informasi administrasi di redha wedding jakarta," *Journal of Information Technology and Computer Science (INTECOMS)*, vol. 7, no. 2, pp. 1–7, 2024.
- [5] Dicoding, "Apa itu Web service? Beserta Pengertian dan Contohnya." Accessed: Jun. 04, 2025. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-web-service/>
- [6] B. Aries. P. I. V. Wicaksono, "Pengembangan REST API untuk Sistem Pelaporan dan Pengaduan dengan Laravel (Studi Kasus: Balai Besar Karantina Pertanian Belawan)," *Journal uii*, vol. 3, no. 2, Oct. 2022.
- [7] Oky Dwi Arianto and Yerymia Alfa Susetyo, "penerapan restful web service dengan framework laravel untuk pembangunan sistem informasi manajemen sumber daya manusia," *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, pp. 522–532, 2022.
- [8] H. Fery Herdiytmoko, "Desain Sistem Backend Berbasis REST API Menggunakan Framework Laravel 7," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 2, pp. 136–144, 2022, [Online]. Available: <http://127.0.0.1:8000/api/siswa>,
- [9] Baharuddin, Hamka Wakkang, and Bambang Irianto, "implementasi web service dengan metode rest api untuk integrasi data covid 19 di sulawesi selatan," *Jurnal Sintaks Logika*, vol. 2, pp. 12–22, Jan. 2022, doi: 10.31850/jsilog.v2i1.
- [10] Annisa Rahmadani, Rizka Rahman Tanjung, and Windi Melisa, "Konsep Administrasi Pendidikan," *Dewantara : Jurnal Pendidikan Sosial Humaniora*, vol. 3, no. 3, pp. 79–86, Aug. 2024, doi: 10.30640/dewantara.v3i3.2848.
- [11] Sriyadi et al., *Modul Teknologi Web service*. 2019.
- [12] Ph. D. Irwan Alnarus Kautsar, *Pengembangan Layanan Web service dengan Microframework*, 1st ed. sidoarjo: UMSIDA Press, 2019.
- [13] M. Massé, *REST API Design Rulebook*, 1st ed. O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2011.
- [14] JSON.org, "Pengenalan JSON," <https://www.json.org/json-en.html>.
- [15] E. Siswanto and M. Kom, *BELAJAR LARAVEL*. semarang: Yayasan Prima Agus Teknik, 2023.
- [16] E. Freitas, *Nuxt.js Succinctly*. Syncfusion, Inc., 2022. [Online]. Available: www.syncfusion.com.
- [17] R. S. Pressman and B. R. Maxim, *SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH*. McGraw-Hill Education, 2 Penn Plaza, New York, NY 10121., 2019.