

Pengembangan *Role-Based Access Control* dan *Single-Page Application* pada Sistem Pelayanan Terpadu Satu Pintu Stasiun Meteorologi Radin Inten II

(Development of *Role-Based Access Control* and *Single-Page Application* in the One-Stop Integrated Service System of Radin Inten II Meteorological Station)

Alexandrio Ariel Rafidan^{1*}, Rio Ariestia P¹, Rudi Hartono²

¹Program Studi Teknik Informatika, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung, Jl. Prof. Dr. Ir. Sumantri Brojonegoro No. 1, Bandar Lampung 35145, Indonesia
²Jl. Alamsyah Ratu Prawira Negara Km. 28 Branti, Natar, Lampung Selatan

Riwayat artikel:

Submitted : 14/01/2026
Published : 02/04/2026

* Email Korespondensi:

2315061071@students.unila.ac.id



Abstrak: Proses permohonan data meteorologi di Stasiun Meteorologi Kelas I Radin Inten II Lampung saat ini masih dilakukan secara konvensional, sehingga menimbulkan kendala dalam transparansi status layanan serta adanya risiko keamanan pada dokumen sensitif pemohon. Laporan ini bertujuan untuk merancang dan membangun sistem informasi Pelayanan Terpadu Satu Pintu (PTSP) berbasis *web* dengan fokus pada efisiensi manajemen administrator dan keamanan data. Proses pengembangan dilakukan menggunakan metodologi Agile dengan mengintegrasikan *framework Laravel* dan *React.js* melalui *Inertia.js*, serta mengimplementasikan fitur *Role-Based Access Control* (RBAC), *Private File Streaming* untuk perlindungan berkas identitas, dan *Task Scheduling* berbasis *Cron Job* untuk otomatisasi sistem. Hasil pengembangan menunjukkan bahwa sistem mampu melakukan pengelolaan data secara aman dan efisien melalui dengan strategi *deployment* yang menjamin integritas kode sumber di lingkungan *shared hosting*. Sistem ini diharapkan dapat mendukung peningkatan kualitas layanan publik di lingkungan BMKG melalui proses administrasi yang lebih cepat, transparan, dan akuntabel bagi pengelola maupun masyarakat.

Kata kunci: keamanan data; *laravel*; otomatisasi; PTSP; *react.js*

Abstract: The meteorological data request process at the Radin Inten II Lampung Class I Meteorological Station is currently still carried out conventionally, thus creating obstacles in the transparency of service status and security risks to sensitive applicant documents. This report aims to design and build a web-based Integrated One-Stop Service (PTSP) information system with a focus on administrator management efficiency and data security. The development process was carried out using the Agile methodology by integrating the *Laravel* and *React.js* frameworks through *Inertia.js*, and implementing *Role-Based Access Control* (RBAC) features, *Private File Streaming* for identity file protection, and *Cron Job-based Task Scheduling* for system automation. The development results show that the system is able to manage data safely and efficiently through a deployment strategy that ensures the integrity of the source code in a shared hosting environment. This system is expected to support improving the quality of public services in the BMKG environment through a faster, more transparent, and accountable administrative process for both managers and the public.

Keywords: automation; data security; *laravel*; PTSP; *react.js*

1. PENDAHULUAN

Pelayanan Terpadu Satu Pintu (PTSP) merupakan inovasi birokrasi yang

mengintegrasikan layanan publik untuk meningkatkan efisiensi, transparansi, dan akuntabilitas [1]. Dengan membatasi interaksi

langsung antara petugas dan pemohon, sistem ini menyederhanakan birokrasi, memberikan kepastian biaya, serta mencegah pungutan liar[2]. Di era digital saat ini, implementasi PTSP sangat mengandalkan portal web (*e-Service*) agar pelacakan status permohonan dapat dilakukan secara mandiri [3]. Sistem informasi berbasis web itu sendiri mengkombinasikan komponen manusia, prosedur, dan teknologi internet untuk mengolah data menjadi informasi bermakna. Implementasinya mampu meningkatkan efektivitas pelayanan, efisiensi waktu, dan akurasi data guna meminimalisir *human error*, serta mengamankan dokumen dari risiko kehilangan [4].

Guna mendukung Stasiun Meteorologi Radin Inten II dalam mendigitalisasi pelayanan data yang masih berjalan secara konvensional, diperlukan pengembangan arsitektur modern web berupa *Single-Page Application* (SPA). SPA mampu menyajikan konten dinamis tanpa pemuatan ulang seluruh halaman dengan mengunduh satu *file* JavaScript di awal dan mengambil data asinkron via API [5]. Kinerja antarmuka yang interaktif ini diukur melalui metrik *Time To Interactive* (TTI) dan *Time To First Byte* (TTFB) [6]. Sebagai fondasi di sisi peladen (*backend*), sistem ini memanfaatkan *framework* Laravel, yakni *framework* PHP *open-source* berarsitektur *Model-View-Controller* (MVC) yang memisahkan pengelolaan data, tampilan, dan logika untuk menghasilkan struktur kode yang rapi [7]. Didukung fitur *Routing*, *Eloquent ORM*, dan *Artisan CLI*, Laravel menawarkan kecepatan pengembangan dan keamanan yang tinggi [8].

Untuk memastikan kerahasiaan dokumen pemohon, sistem ini menerapkan konsep *Role-Based Access Control* (RBAC) yang membatasi hak akses sistem berdasarkan peran pengguna dalam organisasi melalui komponen *users*, *roles*, dan *permissions* [9]. Menggunakan prinsip *least privilege*, metode ini meminimalkan risiko penyalahgunaan data dan terbukti efektif mengefisienkan manajemen akses[10]. Seluruh rancangan arsitektur sistem dimodelkan menggunakan *Unified Modeling Language* (UML) untuk memvisualisasikan struktur dan perilaku perangkat lunak [11], mencakup *Use Case*, *Activity*, *Sequence*, hingga *Class Diagram* [12], [13].

Pengembangan sistem secara keseluruhan dijalankan menggunakan metodologi *Agile*, sebuah pendekatan iteratif dan inkremental yang berfokus pada kolaborasi intensif [14], [15]. Mengandalkan siklus pendek (*sprint*), metode ini sangat adaptif terhadap perubahan [16] dan lebih efektif dalam meminimalkan risiko kegagalan dibandingkan metode tradisional [17]. Oleh karena itu, penelitian ini bertujuan untuk mengintegrasikan teknologi-teknologi tersebut ke dalam sebuah sistem informasi PTSP yang efektif, aman, dan akuntabel, lengkap dengan fitur *Task Scheduling* (*Cron Job*) dan *Private File Storage* untuk mengotomatisasi

layanan meteorologi di Stasiun Meteorologi Radin Inten II.

2. BAHAN DAN METODE PENELITIAN

Penelitian ini menggunakan metode observasi langsung dan wawancara untuk memetakan kendala sistem. Berdasarkan data tersebut, sistem dikembangkan dengan metode *Agile* yang meliputi tahapan *Planning*, *Analysis*, *Design* (menggunakan UML seperti *Use Case* dan *Activity Diagram*), serta *Development* menggunakan *framework* Laravel dan React.js.

2.1. Bahan dan Alat

Tabel 1. Perangkat Keras yang Digunakan

No	Nama Alat	Spesifikasi	Kegunaan
1	Laptop	AMD Ryzen 5 5600H, RAM 16 GB, SSD 512 GB	Digunakan untuk perancangan, pengembangan, dan pengujian sistem

Tabel 2. Perangkat Lunak yang Digunakan

No	Nama Software	Versi	Kegunaan
1	Windows 11	64-bit	Sistem operasi utama pengembangan
2	Visual Studio Code	1.122.1	Editor kode program
3	Laravel	Laravel 12	Framework backend sistem
4	React.js	18.3.1	Framework frontend sistem
5	Inertia.js	2.3.24	Penghubung Laravel dan React.js
6	MySQL	8.4.3	Sistem manajemen basis data
7	Laragon	8.3.0	Web server lokal dan environment development
8	Google Chrome / Microsoft Edge	148.0.3967.96	Pengujian dan akses aplikasi web
9	Git	2.51.0	Version control pengembangan sistem

Tabel 3. Bahan Penelitian

No	Bahan	Sumber	Kegunaan
1	Data alur pelayanan PTSP	Observasi di Stasiun Meteorologi Radin Inten II	Digunakan untuk analisis kebutuhan sistem
2	Dokumen persyaratan layanan	Administrator PTSP	Digunakan untuk perancangan fitur pengajuan layanan
3	Data pengguna dummy	Data simulasi	Digunakan untuk pengujian sistem
4	Literatur jurnal dan dokumentasi teknis	Jurnal ilmiah dan dokumentasi resmi Laravel/React.js implementasi	Digunakan sebagai referensi teori dan implementasi
5	SOP pelayanan PTSP	Instansi terkait	Digunakan untuk memahami prosedur layanan yang berjalan

2.2. Metode

Metodologi *Agile* dipilih karena bersifat iteratif, adaptif, dan mengutamakan kolaborasi pengguna. Pendekatan ini memungkinkan penyesuaian sistem secara berkelanjutan sesuai kebutuhan sekaligus mempercepat proses pengembangan secara keseluruhan.

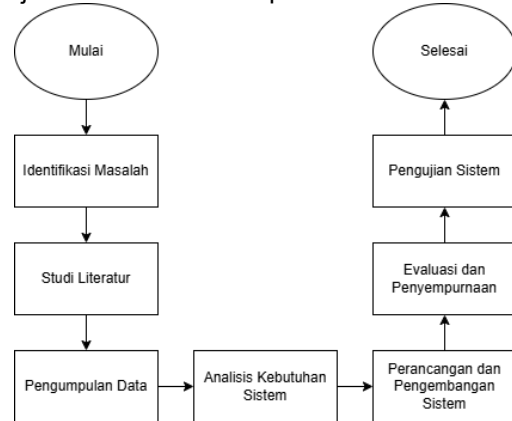
**Gambar 1.** SDLC Agile

Agile diterapkan melalui siklus pendek (*sprint*) yang mencakup enam tahapan terintegrasi: *Planning* untuk merencanakan fitur, *Analysis* untuk kebutuhan sistem, *Design* untuk diagram UML dan basis data, *Development* menggunakan Laravel-React.js-Inertia.js, *Testing* untuk uji fungsionalitas,

serta *Review* untuk evaluasi berdasarkan umpan balik pengguna.

2.3. Alur Pelaksanaan

Prosedur atau tahapan dalam penelitian ini disusun secara sistematis untuk memastikan pengembangan sistem berjalan terstruktur dari awal hingga akhir. Alur pelaksanaan penelitian tersebut ditunjukkan secara visual pada Gambar 2. berikut.

**Gambar 2.** Flowchart Alur Pengerjaan

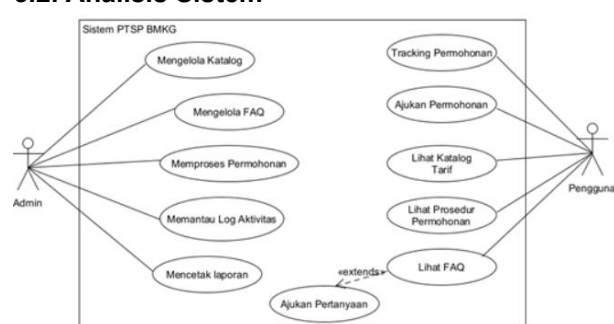
Alur penelitian dimulai dari identifikasi masalah sistem manual dan studi literatur[18]. Data dikumpulkan via observasi dan wawancara, lalu dianalisis untuk merancang diagram UML serta basis data [13]. Setelah itu, sistem diimplementasikan, diuji, dan dievaluasi hingga optimal, dengan rencana hasil berupa *Dashboard Admin*, keamanan *RBAC Private File Streaming*, dan otomatisasi status permohonan.

3. HASIL DAN PEMBAHASAN

3.1. Deskripsi Proyek Web PTSP BMKG

Web PTSP mengotomatisasi pelayanan administrasi konvensional menjadi terpusat dan efisien di Stasiun Meteorologi Radin Inten II. Pengembangan dilakukan secara kolaboratif, dengan fokus penulis pada implementasi fitur *role Admin* untuk manajemen data dan pemantauan pengajuan. Sistem ini memiliki sembilan fitur terintegrasi: *Dashboard Admin*, *Daftar Permintaan*, *FAQ Admin*, *FAQ User*, *Laporan*, *Katalog Layanan*, *Log Aktivitas*, *Panduan*, dan *Profile*.

3.2. Analisis Sistem

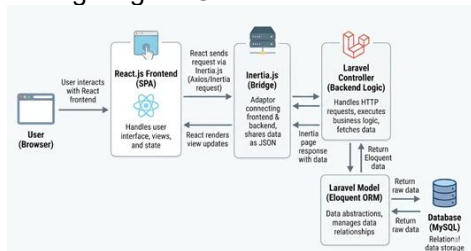
**Gambar 3.** Use Case Diagram

Use case ini mendefinisikan dua peran utama dalam sistem, yaitu Admin dan Pengguna. Peran Admin bertanggung jawab atas pengelolaan katalog layanan meteorologi, moderasi dan publikasi FAQ, pemrosesan verifikasi dokumen permohonan hingga unggah hasil, pemantauan *log* aktivitas (*audit trail*), serta pencetakan laporan pendapatan berkala. Sementara itu, peran Pengguna dapat berinteraksi dengan sistem untuk melihat katalog tarif, mengakses prosedur permohonan, mengajukan permohonan dengan mengunggah dokumen persyaratan, melakukan *tracking* progres permohonan menggunakan NIK dan kata sandi, serta membaca maupun mengajukan pertanyaan baru pada halaman FAQ.

3.3. Rancangan Sistem

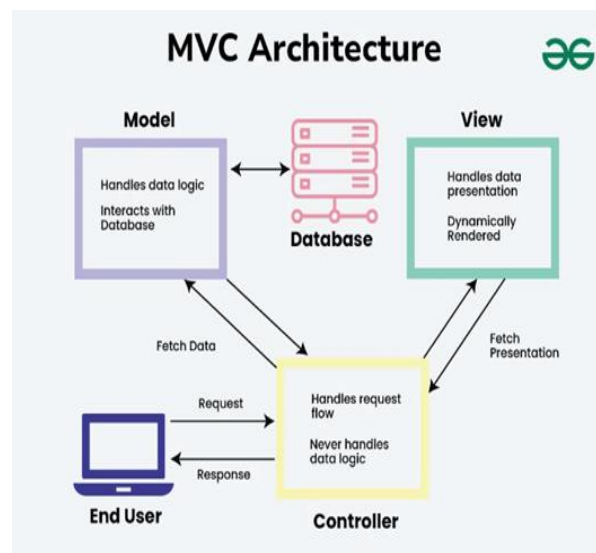
Perancangan basis data MySQL menggunakan lima tabel utama: *data_catalogs* (master layanan) berelasi *One-to-Many* dengan *requests* (transaksi permohonan) untuk menjaga integritas data; *faq_messages* untuk moderasi tanya jawab; serta *users* (kredensial admin) berelasi *One-to-Many* dengan *activity_logs* (*audit trail*) sebagai perekam riwayat tindakan demi keamanan.

Sementara itu, arsitektur sistem menggunakan Inertia.js sebagai jembatan yang menghubungkan Laravel dan React.js tanpa REST API konvensional, menghasilkan interaksi SPA yang cepat tanpa *full-page reload*. Untuk keamanan pada *shared hosting*, diterapkan strategi *folder splitting* dengan mengisolasi direktori inti (*core system* dan file *.env*) di luar direktori publik agar tidak dapat diakses langsung via URL browser.



Gambar 4. Arsitektur Sistem Berbasis Laravel, Inertia.js, dan React.js

Gambar 4. menunjukkan arsitektur sistem yang mengintegrasikan *backend* Laravel dan *frontend* React.js via Inertia.js. Saat pengguna berinteraksi dengan antarmuka React (SPA), *request* dikirim melalui Inertia.js ke Laravel Controller untuk diproses melalui Model (Eloquent ORM) ke database. Data hasil proses dikembalikan dalam bentuk *response* JSON via Inertia.js untuk langsung dirender oleh React tanpa *reload* halaman, sehingga menghasilkan performa sistem yang lebih cepat dan responsif.



Gambar 5. Arsitektur MVC Laravel

Program ini mengadopsi arsitektur MVC berbasis SPA melalui integrasi Laravel, Inertia.js, dan React.js. Model memanfaatkan Eloquent ORM Laravel untuk mengelola CRUD dan relasi database (seperti tabel *requests* dan *data_catalogs*). Controller memproses logika bisnis—seperti verifikasi berkas dan validasi data—lalu mengirimkan data via Inertia.js tanpa view Blade konvensional. Sementara View menggunakan komponen React.js yang merender antarmuka secara dinamis dengan *client-side routing*, menghasilkan pengalaman SPA yang cepat tanpa *reload* halaman.

```

route::middleware(['auth', 'verified'])->prefix('admin')->group(function () {
    Route::get('/dashboard', [DashboardController::class, 'index'])->name('dashboard');
    Route::get('/requests', [RequestManagementController::class, 'index'])->name('requests.index');
    Route::get('/requests/{id}', [RequestManagementController::class, 'show'])->name('requests.show');
    Route::patch('/requests/{id}', [RequestManagementController::class, 'update'])->name('requests.update');
    Route::get('/requests/{id}/file/{type}', [FileStreamController::class, 'streamFile'])->name('file.stream');
    Route::resource('catalogs', DataCatalogController::class)->except(['create', 'edit']);
    Route::resource('faqs', FaqManagementController::class)->only(['index', 'update', 'destroy']);
    Route::get('/logs', [ActivityLogController::class, 'index'])->name('logs.index');
    Route::get('/reports', [ReportController::class, 'index'])->name('reports.index');
    Route::get('/reports/download', [ReportController::class, 'download'])->name('reports.download');
    Route::get('/panduan', function () { return Inertia::render('Admin/Panduan'); })->name('panduan');
});

```

Gambar 6. Implementasi Middleware pada Rute Administratif

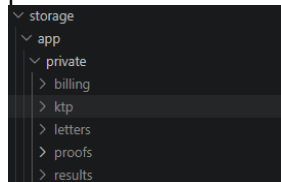
Sistem mengimplementasikan *middleware* *auth* dan *verified* pada kelompok rute ber-prefix */admin* untuk memastikan bahwa hanya administrator terautentikasi dan terverifikasi yang dapat mengakses rute administratif seperti *dashboard*, manajemen permohonan, katalog, dan *file streaming* demi membatasi hak akses serta meningkatkan keamanan aplikasi.

3.4. Pengembangan Fitur Keamanan dan Otomasi Sistem

Implementasi keamanan sistem memanfaatkan *middleware* sebagai lapisan proteksi yang mengintersepsi setiap *request* sebelum diproses oleh *controller*. Melalui mekanisme *Role-Based Access Control* (RBAC) sederhana, *middleware* mendeteksi sesi *login* dan memeriksa nilai *role* pengguna pada database; jika pengguna non-admin mencoba mengakses rute

administratif (seperti /admin/dashboard), sistem otomatis memblokir akses dengan respons *403 Forbidden* atau mengalihkan halaman ke menu *login* demi menjamin integritas data fungsional pengelola.

Untuk melindungi dokumen sensitif pengguna (seperti KTP), berkas disimpan di direktori non-publik `storage/app/private` tanpa akses URL langsung. Dokumen hanya dapat dibuka oleh administrator sah melalui *endpoint* khusus terautentikasi yang mengimplementasikan metode *file streaming* via fungsi `response()->file()` atau `StreamedResponse`.



Gambar 7. Penyimpanan Berkas pada Direktori Private

Berdasarkan Gambar 7., seluruh berkas pengguna disimpan di direktori non-publik `storage/app/private` tanpa akses URL langsung demi meningkatkan keamanan data pribadi.

```
private function streamFile($path)
{
    if (!$path || !Storage::disk('local')->exists($path)) {
        abort(404, 'File tidak ditemukan di server.');
```

Gambar 8. Implementasi File Streaming pada Controller

Berdasarkan Gambar 8., fungsi `streamFile()` mengamankan berkas dengan memeriksa keberadaannya via `Storage::disk('local')->exists()` sebelum mengembalikan respons *error 404* jika absen. Berkas yang ditemukan diambil menggunakan `Storage::disk('local')->get()`, diidentifikasi tipe datanya via `mimeType()`, lalu dikirim ke browser dengan *header Content-Type* yang sesuai melalui kontrol sistem terautentikasi tanpa pernah mengekspos URL publiknya.

Otomatisasi sistem menggunakan perintah php artisan `schedule:run` yang didaftarkan pada Cron Job cPanel dengan interval eksekusi setiap 15 menit. Secara berkala, sistem akan menjalankan *query* pada tabel `requests` untuk memeriksa permohonan berstatus menunggu pembayaran yang telah melewati batas waktu `expired_at`, lalu otomatis mengubah statusnya menjadi 'invalid' demi menjaga kebersihan basis data dari transaksi kedaluwarsa.

```
public function handle()
{
    $invalidated = DataRequest::where('status', 'waiting_payment')
        ->where('va_expired_at', '<', now())
        ->update(['status' => 'invalid']);

    $expired = DataRequest::whereIn('status', ['paid', 'done'])
        ->where('download_expired_at', '<', now())
        ->update(['status' => 'expired']);

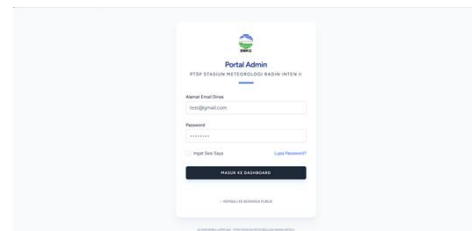
    $this->info("Berhasil: {$invalidated} billing jadi invalid, {$expired} link jadi expired.");
}
```

Gambar 9. Implementasi Logika Otomatisasi Status Permohonan

Berdasarkan Gambar 9., otomatisasi via *command* Laravel berkala memperbarui status pada tabel `requests`: data `waiting payment` yang melewati batas `va_expired_at` diubah menjadi `invalid`, sedangkan data `paid` atau `done` yang melewati batas `download_expired_at` diubah menjadi `expired` guna menjaga konsistensi data secara mandiri.

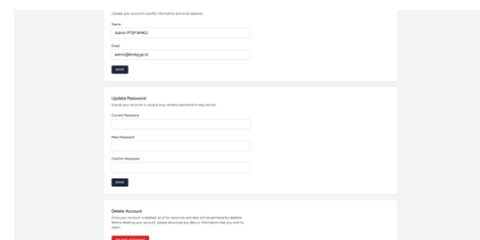
3.5 Hasil Pengembangan Website PTSP

Sebagai hasil akhir dari tahap implementasi, sistem Pelayanan Terpadu Satu Pintu (PTSP) ini telah berhasil diunggah ke *server* produksi dan sepenuhnya beroperasi secara daring. Sistem kini dapat diakses dengan baik oleh publik maupun pemohon layanan data kapan saja melalui tautan <https://ptsp.stametlampung.com>. Ketersediaan akses daring ini membuktikan bahwa website telah berfungsi sesuai dengan rancangan dan siap diimplementasikan untuk mendukung proses digitalisasi layanan secara nyata.



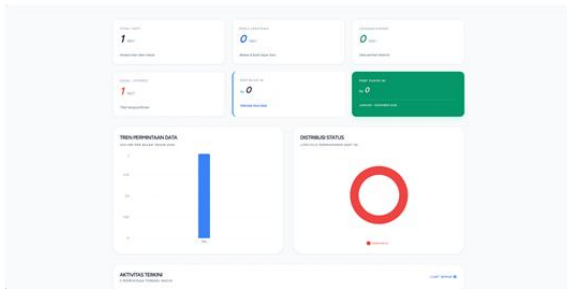
Gambar 10. Tampilan Halaman login

Halaman *login* merupakan pintu masuk administrator ke sistem PTSP BMKG melalui autentikasi email dinas dan kata sandi terdaftar. Halaman ini dilengkapi fitur "ingat sesi", pemulihan akun via "lupa password", serta navigasi kembali ke halaman publik. Setelah kredensial dinyatakan valid, sistem otomatis mengarahkan administrator ke halaman *dashboard*.



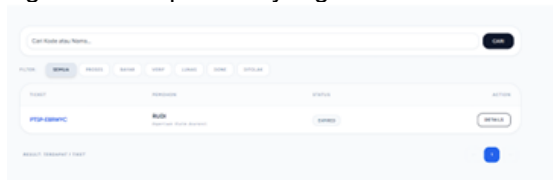
Gambar 11. Tampilan Halaman Profile

Halaman profil admin berfungsi bagi administrator untuk mengelola informasi akun dan keamanan akses. Fitur yang tersedia meliputi pembaruan nama dan email, perubahan kata sandi melalui verifikasi password lama, serta opsi penghapusan akun secara permanen dari sistem untuk memberikan kontrol penuh terhadap manajemen kredensial admin.



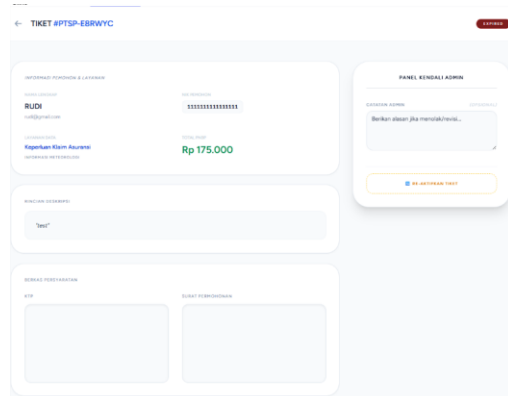
Gambar 12. Tampilan Halaman Dashboard Admin

Halaman *dashboard* merupakan pusat kontrol utama bagi administrator untuk memantau aktivitas sistem secara keseluruhan. Halaman ini menyajikan ringkasan informasi penting seperti total permohonan, status verifikasi, layanan selesai, transaksi gagal/kedaluwarsa, statistik pendapatan, log aktivitas terbaru, serta visualisasi grafik tren data secara *real-time* guna mendukung pengambilan keputusan yang efektif dan efisien.



Gambar 13. Tampilan Halaman Daftar Permohonan

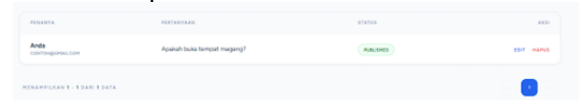
Halaman daftar permohonan berfungsi menampilkan seluruh data masuk secara ringkas, meliputi kode tiket, nama pemohon, dan status permohonan. Halaman ini dilengkapi fitur pencarian kata kunci, filter status untuk penyaringan data, serta tombol aksi untuk mengakses detail setiap permohonan guna mempermudah administrator dalam memantau dan mengelola transaksi secara efisien.



Gambar 14. Tampilan Halaman Detail Permohonan

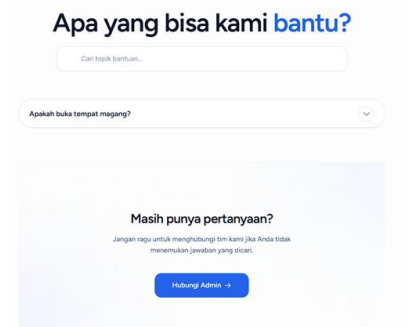
Halaman detail permohonan menyajikan informasi lengkap meliputi data pemohon, jenis layanan, total biaya, deskripsi, serta berkas persyaratan (KTP dan surat permohonan). Halaman ini juga menyediakan panel kendali bagi administrator untuk memberikan catatan, memperbarui status, dan melakukan tindakan

lanjutan seperti mengaktifkan kembali tiket sebagai pusat utama proses verifikasi.



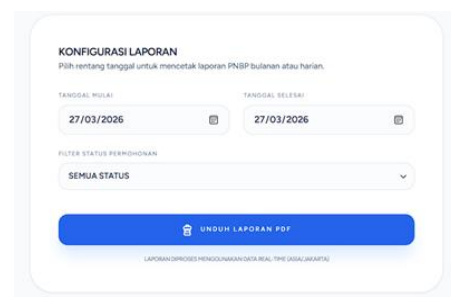
Gambar 15. Tampilan Halaman FAQ Admin

Halaman FAQ Admin digunakan untuk mengelola pertanyaan pengguna, menampilkan daftar pertanyaan, informasi penanya, dan status publikasi. Administrator dapat mengedit jawaban, menghapus pertanyaan, serta menentukan status publikasi sebagai bentuk moderasi konten agar informasi yang disajikan tetap relevan dan terkontrol.



Gambar 16. Tampilan Halaman FAQ User

Halaman FAQ Pengguna merupakan akses publik yang menyajikan daftar pertanyaan dan jawaban terstruktur hasil publikasi administrator. Halaman ini dilengkapi fitur pencarian untuk memudahkan penemuan informasi serta opsi menghubungi administrator jika pengguna membutuhkan bantuan lebih lanjut, berfungsi sebagai media penyampaian informasi yang meningkatkan kemudahan akses layanan.



Gambar 17. Tampilan Halaman Laporan

Halaman laporan memungkinkan administrator menghasilkan data layanan dan pendapatan berdasarkan rentang tanggal serta filter status permohonan tertentu. Setelah parameter ditentukan, sistem memproses data secara *real-time* dan menyediakan opsi unduh format PDF untuk mendukung kebutuhan evaluasi serta pelaporan operasional yang terstruktur.

Kategori	Satuan	Tarif PNBP	Status
Kegiatan Olah Raga	PER LOKASI - PER HARI	Rp 100.000	AKTIF
Kegiatan Komersial Outdoor/Indoor	PER LOKASI - PER HARI	Rp 100.000	AKTIF
Data Raster Cuaca (per 10 menit)	PER LOKASI - PER HARI	Rp 70.000	AKTIF
Kepentingan Klien Asuransi	PER LOKASI - PER HARI	Rp 175.000	AKTIF
Pendukung Kegiatan Proyek, Survey, dan Penelitian Komersial	PER LOKASI	Rp 3.750.000	AKTIF

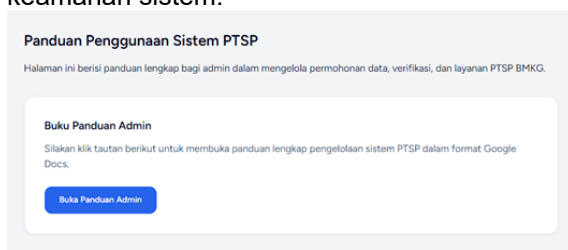
Gambar 18. Tampilan Halaman Katalog Layanan

Halaman katalog layanan digunakan administrator untuk mengelola daftar layanan meteorologi, menampilkan jenis, kategori, satuan, serta tarif PNBP yang berlaku. Halaman ini menyediakan fitur tambah, ubah, dan hapus data layanan, serta berfungsi sebagai pusat pengaturan yang menjadi acuan utama bagi pengguna dalam mengajukan permohonan.

Username	Aksi	Detail
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru
admin	Menyimpan	Menyimpan data layanan baru

Gambar 19. Tampilan Halaman Log Aktivitas

Halaman log aktivitas menampilkan riwayat aksi sistem untuk kebutuhan audit, mencakup waktu, pelaku, jenis aktivitas, dan detail perubahan data. Pencatatan terstruktur ini membantu administrator memantau setiap perubahan, meningkatkan transparansi, serta mendukung keamanan sistem.



Gambar 20. Tampilan Halaman Panduan

Halaman panduan menyediakan informasi penggunaan sistem PTSP bagi administrator melalui tautan dokumentasi lengkap mengenai pengelolaan permohonan, verifikasi data, dan pengoperasian fitur lainnya guna membantu memahami alur sistem secara mudah dan sistematis.

4. KESIMPULAN

Kesimpulan dari laporan ini menunjukkan bahwa pengembangan sistem PTSP menggunakan arsitektur *Modern Monolith* (Laravel dan React.js) berhasil menghasilkan aplikasi SPA berkinerja tinggi yang bebas dari *full-page reload*. Aspek keamanan sistem diperkuat melalui mekanisme RBAC dan *Private File Streaming* untuk melindungi dokumen sensitif (KTP), serta strategi *folder*

splitting pada *shared hosting* guna mengamankan kode sumber dan file konfigurasi dari akses ilegal. Selain itu, pemanfaatan *Task Scheduling* via Cron Job mampu mengotomatisasi manajemen status transaksi kadaluarsa secara mandiri tanpa intervensi manual petugas. Implementasi di Stasiun Meteorologi Kelas I Radin Inten II ini tidak hanya memberikan solusi operasional yang efisien bagi instansi, tetapi juga memperluas wawasan nyata mengenai seluruh siklus pengembangan perangkat lunak hingga tahap *deployment* di server produksi.

UCAPAN TERIMA KASIH

Penulis menyampaikan apresiasi dan terima kasih yang sebesar-besarnya kepada Stasiun Meteorologi Kelas I Radin Inten II Lampung atas dukungan fasilitas dan kesempatan yang diberikan selama masa kerja praktik. Terima kasih secara khusus juga ditujukan kepada Bapak Rudi Harianto, S.Kom., M.Si., selaku mentor dan pembimbing lapangan, yang senantiasa memberikan arahan, bimbingan, serta masukan yang membangun selama proses pengembangan sistem dan penyusunan penelitian ini

DAFTAR PUSTAKA

- [1] Suhartoyo. Fungsi Pelayanan Publik dalam Pelayanan Terpadu Satu Pintu Suhartoyo, "Implementasi Fungsi Pelayanan Publik dalam Pelayanan Terpadu Satu Pintu (PTSP)," *Administrative Law and Governance Journal*, vol. 2, no. 1, pp. 143–154, Jun. 2019, doi: 10.14710/ALJ.V2I1.143-154.
- [2] N. Rahmatullah and A. Asmarini, "Public Service Transformation: The Implementation Of Ptsp In Class 1a Of The Palu Religious Court," *JPASDEV: Journal of Public Administration and Sociology of Development*, vol. 5, no. 1, pp. 685–697, Jul. 2024, doi: 10.26418/jpasdev.vxx.
- [3] S. I. Pelayanan, A. Kependudukan, B. Website, K. Banaran, M. V. Al Hasri, and E. Sudarmilah, "Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis Website Kelurahan Banaran," *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 2, pp. 249–260, May 2021, doi: 10.30812/matrik.v20i2.1056.
- [4] I. Widiastuti and U. Krisnadwipayana, "Sistem Informasi Pelayanan Desa Berbasis Web di Desa Wanajaya Jawa Barat," *Dikmas: Jurnal Pendidikan Masyarakat dan Pengabdian*, vol. 2, no. 3, pp. 877–886, Sep. 2022, doi: 10.37905/DIKMAS.2.3.877-886.2022.
- [5] P. L. Belluano and P. L. L. Belluano, "Pengembangan Single Page Application Pada Sistem Informasi Akademik," *ILKOM Jurnal Ilmiah*, vol. 10, no. 1, pp. 38–43, Apr. 2018, doi: 10.33096/ilkom.v10i1.204.38-43.
- [6] D. V. Kornienko, S. V. Mishina, and M. O. Melnikov, "The Single Page Application

- architecture when developing secure Web services," *J. Phys. Conf. Ser.*, vol. 2091, no. 1, p. 012065, Nov. 2021, doi: 10.1088/1742-6596/2091/1/012065.
- [7] M. H. Tinambunan, M. Y. Simanjuntak, N. Marbun, and G. K. M. Baali, "Membangun Backend & Application Programming Interface (Api) Dengan Laravel," *Penerbit Tahta Media*, Jul. 2025, Accessed: Apr. 15, 2026. [Online]. Available: <http://tahtamedia.co.id/index.php/issj/article/view/1539>
- [8] G. R. U. Sinaga and S. Samsudin, "Implementasi Framework Laravel dalam Sistem Reservasi pada Restoran Cindelaras Kota Medan," *Jurnal Janitra Informatika dan Sistem Informasi*, vol. 1, no. 2, pp. 73–84, Oct. 2021, doi: 10.25008/JANITRA.V1I2.131.
- [9] M. Firmansyah, Mustika, and Pujiyanto, "Implementasi Role-Based Access Control Pada Sistem Informasi Layanan Administrasi Pada Lembaga Bahasa Um Metro," *Jurnal Mahasiswa Ilmu Komputer*, vol. 6, no. 2, pp. 113–122, Oct. 2025, doi: 10.24127/ILMUKOMPUTER.V6I2.9875.
- [10] V. G. Penggalih and E. P. Silmina, "Implementasi Dashboard Proyek Dengan Sistem Rbac Pada Pt.Xyz," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, pp. 2135–2142, Mar. 2025, doi: 10.36040/JATI.V9I2.12978.
- [11] M. Nugraha and M. Rosmeida, "Perancangan Sistem Informasi Beban Kerja Dosen Berbasis Web dengan UML," *Jurnal Algoritma*, vol. 18, no. 1, pp. 141–150, Aug. 2021, doi: 10.33364/ALGORITMA/V.18-1.866.
- [12] D. D. Randa, Y. M. Putra, and R. Noviardi, "Perancangan Sistem Informasi Penjualan Berbasis Online Menggunakan Pemrograman PHP dan Pemodelan UML (Studi Kasus Koningshooes Padang)," *JRTI (Jurnal Riset Tindakan Indonesia)*, vol. 7, no. 4, pp. 756–762, 2022, doi: 10.29210/30032656000.
- [13] H. N. Putra, "Implementasi Diagram UML (Unified Modelling Language) dalam Perancangan Aplikasi Data Pasien Rawat Inap pada Puskesmas Lubuk Buaya," *Sinkron: jurnal dan penelitian teknik informatika*, vol. 2, no. 2, pp. 67–77, Mar. 2018, Accessed: Apr. 13, 2026. [Online]. Available: <https://jurnal.polgan.ac.id/index.php/sinkron/article/view/130>
- [14] I. Burhani, J. Juwari, A. Soderi, and K. Diantoro, "Perbandingan Metodologi SDLC Waterfall dan Agile Dalam Rencana Pengembangan Sistem Informasi Kepatuhan," *Journal of Informatics Management and Information Technology*, vol. 5, no. 2, pp. 147–154, Apr. 2025, doi: 10.47065/JIMAT.V5I2.489.
- [15] A. Sinha and P. Das, "Agile Methodology Vs. Traditional Waterfall SDLC: A case study on Quality Assurance process in Software Industry," *2021 5th International Conference on Electronics, Materials Engineering and Nano-Technology, IEMENTech 2021*, 2021, doi: 10.1109/IEMENTECH53263.2021.9614779
- [16] D. S. B. Ginting, S. N. Salsabila, D. Lorenza, D. P. Akbar, T. B. D. Sitepu, and R. F. T. Lubis, "Implementation of Software Development Life Cycle (SDLC) and Agile Methods in Mobile-Based Learning to Improve Maternal Understanding of Stunting," pp. 1–6, Jan. 2026, doi: 10.1109/ICCAI65301.2025.11278988.
- [17] S. Pargaonkar, "A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering," *International Journal of Scientific and Research Publications*, vol. 13, no. 8, pp. 120–124, Aug. 2023, doi: 10.29322/IJSRP.13.08.2023.P14015.
- [18] F. Mahardika, S. G. Merani, A. T. Suseno, and D. Redaksi, "Penerapan Metode Extreme Programming pada Perancangan UML Sistem Informasi Penggajian Karyawan," *Blend Sains Jurnal Teknik*, vol. 2, no. 3, pp. 204–217, Dec. 2024, doi: 10.56211/BLENDSAINS.V2I3.313.