Indonesian Journal of Multidisciplinary on Social and Technology
Homepage: https://journal.ilmudata.co.id/index.php/ijmst
Vol. 4 No. 3 (2026) pp: 6403-6410
P-ISSN: 2986-6790, e-ISSN: 2986-6782

Pengembangan Middleware PHP untuk Sinkronisasi Data Kamar Rumah Sakit

Achmad Fauzan, Gusmelia Testiana
Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, UIN Raden Fatah Palembang
0106fauzan@gmail.com, gusmeliatestiana_uin@radenfatah.ac.id

Abstrak

Pengelolaan data ketersediaan kamar rawat inap di rumah sakit pemerintah umumnya masih dilakukan secara manual sehingga informasi yang tersaji pada platform terintegrasi daerah sering kali tidak mencerminkan kondisi aktual. Penelitian ini memaparkan pengembangan middleware berbasis PHP yang menghubungkan basis data lokal rumah sakit dengan API Dashboard Satu Data Provinsi Sumatera Selatan selama Program Magang Berdampak Kominfo di Rumah Sakit Khusus Gigi dan Mulut (RSKGM) Provinsi Sumatera Selatan. Middleware dirancang dengan arsitektur tiga lapis, yaitu lapisan antarmuka pengguna, lapisan logika bisnis, dan lapisan integrasi API. Lapisan antarmuka menghadirkan dasbor editable berbasis Bootstrap 5 yang memungkinkan operator mengelola data kabupaten/kota, nama rumah sakit, kelas ruang inap, jumlah kamar, dan jumlah kamar kosong secara langsung. Lapisan logika bisnis menangani validasi input, sinkronisasi penuh dengan basis data lokal MariaDB melalui mekanisme truncate-and-reinsert, serta komputasi ringkasan statistik secara real-time. Lapisan integrasi API menggunakan cURL untuk mengirimkan data dalam format multipart/form-data ke endpoint dashboard.sumselprov.go.id dengan autentikasi JWT Bearer Token. Selain pengiriman manual, middleware menyediakan fitur auto-kirim dengan interval satu jam. Pengujian dilakukan terhadap 45 data kamar dari RSKGM Provinsi Sumatera Selatan dengan hasil HTTP 200 pada seluruh rekaman. Middleware ini terbukti mampu mengurangi keterlambatan pemutakhiran data, menjamin konsistensi antara basis data lokal dan dashboard provinsi, serta meningkatkan efisiensi pengelolaan data tanpa memerlukan instalasi tambahan selain PHP dan MariaDB.

Kata kunci: Middleware, PHP, API Integration, Satu Data Indonesia, Ketersediaan Kamar Rumah Sakit, Dashboard, cURL, JWT Authentication

Abstract

Managing inpatient room availability data in government hospitals is commonly performed manually, causing the information presented on regional integrated platforms to frequently fail in reflecting the actual condition. This paper describes the development of a PHP-based middleware that bridges the local hospital database with the API of the South Sumatra Provincial One Data Dashboard (Dashboard Satu Data Provinsi Sumatera Selatan) during the Kominfo Impactful Internship Program at Rumah Sakit Khusus Gigi dan Mulut (RSKGM) Provinsi Sumatera Selatan. The middleware is designed using a three-tier architecture comprising a user-interface layer, a business-logic layer, and an API-integration layer. The user-interface layer provides a Bootstrap 5-based editable dashboard that enables operators to directly manage data including regency/city name, hospital name, ward class, total rooms, and available rooms. The business-logic layer handles input validation, full synchronization with the local MariaDB database via a truncate-and-reinsert mechanism, and real-time summary statistics computation. The API-integration layer employs cURL to transmit data in multipart/form-data format to the dashboard.sumselprov.go.id endpoint with JWT Bearer Token authentication. In addition to manual submission, the middleware provides an auto-send feature at one-hour intervals. Testing was conducted on 45 room records from RSKGM Provinsi Sumatera Selatan, yielding HTTP 200 responses for all records. The middleware demonstrably reduces data update latency, ensures consistency between the local database and the provincial dashboard, and improves data management efficiency without requiring additional installation beyond PHP and MariaDB.

Keywords: Middleware, PHP, API Integration, One Data Indonesia, Hospital Room Availability, Dashboard, cURL, JWT Authentication

1. Pendahuluan

Implementasi kebijakan Satu Data Indonesia yang diatur dalam Peraturan Presiden Nomor 39 Tahun 2019 mendorong setiap Organisasi Perangkat Daerah (OPD) untuk mengintegrasikan data operasionalnya ke dalam platform data terpadu daerah (Kementerian Komunikasi dan Informatika Republik Indonesia, 2021). Di Provinsi Sumatera Selatan, integrasi ini dikelola melalui Dashboard Satu Data Provinsi Sumatera Selatan yang dikembangkan oleh Dinas Komunikasi dan Informatika (Diskominfo) Provinsi Sumatera Selatan (Diskominfo Provinsi Sumatera Selatan, 2024). Salah satu data strategis yang wajib tersaji secara akurat adalah data ketersediaan kamar rawat inap rumah sakit, mengingat informasi ini digunakan publik untuk mengetahui kapasitas layanan kesehatan serta dimanfaatkan instansi kesehatan lain sebagai acuan dalam koordinasi rujukan pasien.

Rumah Sakit Khusus Gigi dan Mulut (RSKGM) Provinsi Sumatera Selatan, sebagai salah satu OPD di lingkungan Pemerintah Provinsi Sumatera Selatan, menghadapi tantangan dalam pemutakhiran data kamar secara berkala. Berdasarkan hasil observasi dan wawancara selama Program Magang Berdampak Kominfo periode September–Desember 2025, ditemukan bahwa proses penginputan data masih dilakukan secara manual melalui antarmuka web dashboard, tanpa mekanisme otomatis yang menghubungkan sistem informasi manajemen rumah sakit (SIMRS) dengan platform Satu Data provinsi. Kondisi ini menyebabkan keterlambatan pemutakhiran data, inkonsistensi format, dan ketergantungan pada ketersediaan operator (Diskominfo Provinsi Sumatera Selatan, 2024).

Penelitian terdahulu menunjukkan bahwa pendekatan middleware berbasis REST API efektif dalam menjembatani sistem-sistem yang heterogen di lingkungan pemerintahan (Pressman & Maxim, 2020; Sommerville, 2016). Middleware berfungsi sebagai lapisan perantara yang menangani transformasi data, autentikasi, dan mekanisme pengiriman sehingga sistem sumber tidak perlu dimodifikasi (Laudon & Laudon, 2020). Dalam konteks integrasi data kesehatan, middleware juga telah diterapkan untuk memastikan akurasi dan konsistensi data antar sistem yang tidak dirancang untuk saling berkomunikasi secara native (O'Brien & Marakas, 2019).

Beberapa penelitian terdahulu telah membahas integrasi data lintas sistem pada konteks layanan publik. Penelitian mengenai optimasi layanan berbasis chatbot pada platform e-commerce menunjukkan bahwa lapisan perantara dapat meningkatkan responsivitas sistem tanpa mengubah basis data utama (Rizki et al., 2022). Studi lain mengenai analisis stabilitas tautan dalam protokol perutean jaringan zona dinamis menegaskan pentingnya parameter timeout dan mekanisme penanganan kegagalan pada komunikasi antar sistem yang melibatkan jaringan publik dengan latensi bervariasi (Fadhillah et al., 2020). Sementara itu, kajian sistem informasi akuntansi menggarisbawahi pentingnya kontrol integritas data, seperti validasi input dan penggunaan parameter terikat (bound parameters), saat data dipindahkan antar basis data yang berbeda kepemilikan (McLeod & Schell, 2019). Penelitian mengenai dampak penerapan e-commerce pada usaha mikro juga menunjukkan bahwa kesederhanaan antarmuka pengguna menjadi faktor penentu keberhasilan adopsi suatu sistem informasi oleh operator nonteknis (Rizki, 2022).

Namun demikian, sebagian besar penelitian tersebut berfokus pada konteks integrasi sistem bisnis, layanan pelanggan, atau jaringan komputer secara umum, dan belum membahas secara spesifik tantangan integrasi data layanan kesehatan milik pemerintah daerah dengan keterbatasan infrastruktur dan sumber daya manusia teknologi informasi yang umum dijumpai di instansi kesehatan tingkat provinsi. Kesenjangan (gap) yang teridentifikasi dari tinjauan literatur tersebut adalah belum tersedianya studi kasus implementasi middleware ringan, tanpa framework backend tambahan, yang secara khusus dirancang untuk mengatasi keterbatasan infrastruktur OPD kesehatan daerah, sekaligus mengakomodasi dua mekanisme autentikasi sekaligus, yaitu token berbasis JWT dan session cookie, sebagaimana dipersyaratkan oleh banyak sistem pemerintahan berbasis CodeIgniter di Indonesia.

Pembaruan (novelty) yang ditawarkan penelitian ini adalah desain middleware satu berkas (single-file) yang menggabungkan antarmuka manajemen data, sinkronisasi basis data lokal, dan integrasi API multi-autentikasi dalam satu alur kerja yang dapat dioperasikan oleh staf nonteknis tanpa pelatihan mendalam. Penelitian ini bertujuan merancang dan mengimplementasikan middleware berbasis PHP yang mampu: (1) menyediakan antarmuka pengelolaan data yang ramah pengguna bagi operator rumah sakit, (2) menyinkronkan data dengan basis data lokal secara konsisten, dan (3) mengirimkan data ke API Dashboard Satu Data Provinsi Sumatera Selatan secara otomatis maupun manual. Kontribusi utama penelitian ini adalah tersedianya solusi integrasi ringan yang dapat dioperasikan tanpa instalasi tambahan selain PHP dan MariaDB, serta dapat diadaptasi oleh OPD lain dengan kebutuhan serupa.

DOI: https://doi.org/10.69693/ijmst.v4i3.10661
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

2. Metode Penelitian

Penelitian ini menggunakan pendekatan Research and Development (R&D) dengan model pengembangan perangkat lunak waterfall yang diadaptasi menjadi lima tahap, yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan evaluasi alur kerja (Jogiyanto, 2017). Pengembangan dilakukan selama Program Magang Berdampak Kominfo di RSKGM Provinsi Sumatera Selatan pada periode September–Desember 2025.

2.1. Analisis Kebutuhan

Analisis kebutuhan dilakukan melalui observasi langsung, wawancara terstruktur dengan Kepala Bagian Tata Usaha dan staf IT, serta studi dokumentasi terhadap panduan penggunaan Dashboard Satu Data Provinsi Sumatera Selatan yang diterbitkan oleh Diskominfo (Diskominfo Provinsi Sumatera Selatan, 2024). Dari hasil analisis diperoleh kebutuhan fungsional sebagai berikut: (1) sistem harus dapat membaca dan menampilkan data ketersediaan kamar dari basis data lokal, (2) operator dapat menambah, mengubah, dan menghapus data secara langsung melalui antarmuka tabel, (3) sistem menyimpan perubahan ke basis data lokal, dan (4) sistem mengirimkan data ke API eksternal dengan protokol autentikasi yang sesuai.

2.2. Perancangan Sistem

Sistem dirancang menggunakan arsitektur tiga lapis (three-tier architecture) yang terdiri dari lapisan presentasi (presentation layer), lapisan logika bisnis (business logic layer), dan lapisan integrasi API (API integration layer). Pemisahan lapisan ini bertujuan untuk meningkatkan keterpeliharaan (maintainability) dan memudahkan pengembangan lebih lanjut (Satzinger et al., 2016). Basis data dirancang dengan satu tabel utama data_ketersediaan_kamar_rumah_sakit pada MariaDB dengan struktur yang identik dengan skema basis data Dashboard Satu Data Provinsi Sumatera Selatan, sehingga meminimalkan transformasi data saat pengiriman.

2.3. Implementasi

Implementasi dilakukan dalam satu file PHP monolitik sesuai dengan kendala lingkungan produksi di RSKGM yang tidak mendukung framework PHP. Teknologi yang digunakan meliputi PHP 7.4+, MariaDB 10.4, Bootstrap 5.3, dan library cURL bawaan PHP (Kadir, 2018). Kode dikembangkan secara iteratif dengan pendekatan test-driven terhadap endpoint API aktual Dashboard Satu Data Provinsi Sumatera Selatan.

2.4. Pengujian

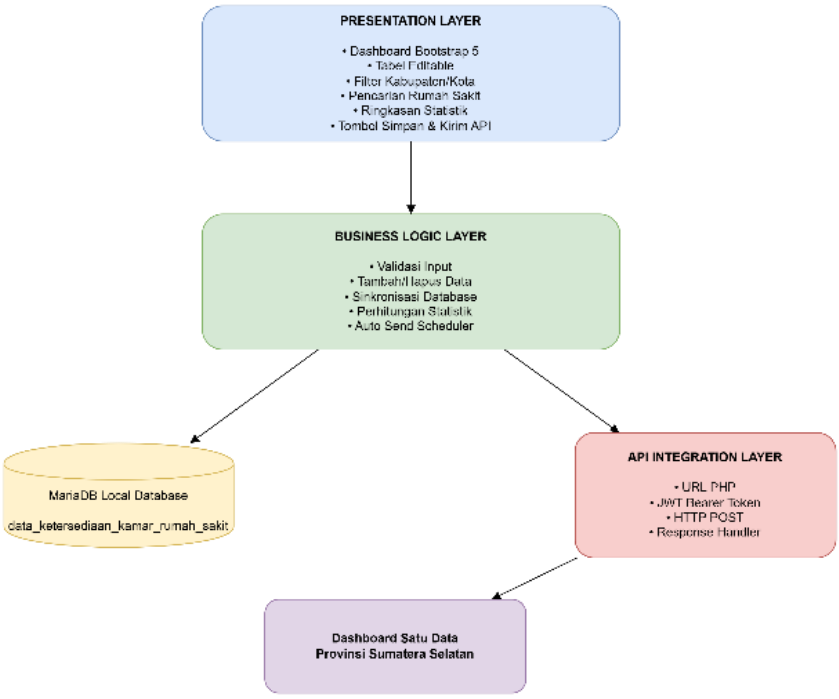
Pengujian dilakukan menggunakan pendekatan black-box testing dengan dua skenario: (1) pengujian fungsional terhadap seluruh fitur antarmuka dan logika bisnis, dan (2) pengujian integrasi terhadap API eksternal menggunakan data aktual RSKGM Provinsi Sumatera Selatan. Parameter keberhasilan integrasi API adalah respons HTTP 200 dari server Dashboard Satu Data Provinsi Sumatera Selatan.

3. Hasil dan Pembahasan

3.1. Arsitektur Middleware

Middleware yang dikembangkan mengimplementasikan arsitektur tiga lapis sebagaimana ditunjukkan pada Gambar 1. Lapisan presentasi (presentation layer) bertanggung jawab atas rendering antarmuka HTML menggunakan Bootstrap 5.3, pengelolaan form interaktif dengan JavaScript vanilla, serta fitur filter dan pencarian berbasis DOM. Lapisan logika bisnis (business logic layer) mencakup pemrosesan input POST, validasi dan pembersihan data, komputasi ringkasan statistik, dan orkestrasi alur kerja simpan/kirim. Lapisan integrasi API (API integration layer) mengenkapsulasi seluruh komunikasi dengan API eksternal menggunakan library cURL PHP (Laudon & Laudon, 2020).

DOI: https://doi.org/10.69693/ijmst.v4i3.10661
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)



Gambar 1. Arsitektur Tiga Lapis Middleware

3.2. Struktur Basis Data

Tabel 1 menampilkan struktur tabel data_ketersediaan_kamar_rumah_sakit yang digunakan oleh middleware. Desain tabel mengadopsi skema yang identik dengan basis data Dashboard Satu Data Provinsi Sumatera Selatan untuk memastikan kompatibilitas data tanpa konversi (Diskominfo Provinsi Sumatera Selatan, 2024). Indeks komposit unik pada kolom kabkota, rumah_sakit, dan kelas_ruang_inap menjamin tidak ada duplikasi data untuk kombinasi ketiga atribut tersebut.

Tabel 1. Struktur Tabel Basis Data Middleware

Kolom	Tipe Data	Keterangan
kabkota	VARCHAR(100)	Nama Kabupaten/Kota
rumah_sakit	VARCHAR(150)	Nama Rumah Sakit
kelas_ruang_inap	VARCHAR(150)	Kelas Ruang Inap
jumlah_kamar	INT(11)	Total kamar tersedia
jumlah_kamar_kosong	INT(11)	Kamar belum terisi

3.3. Lapisan Logika Bisnis

Lapisan logika bisnis mengimplementasikan mekanisme sinkronisasi penuh (full synchronization) dengan basis data lokal. Ketika operator menekan tombol "Simpan ke Database" atau "Kirim Data ke API", sistem terlebih dahulu melakukan truncate pada tabel data_ketersediaan_kamar_rumah_sakit, kemudian menyisipkan ulang seluruh rekaman dari tampilan form menggunakan prepared statement dengan parameter binding (bind_param) untuk mencegah SQL injection (McLeod & Schell, 2019). Pendekatan truncate-and-reinsert dipilih dibanding upsert (INSERT ... ON DUPLICATE KEY UPDATE) karena lebih sederhana untuk dikelola dan memastikan konsistensi penuh antara tampilan dan isi basis data.

Fungsi ambilDataDariTabel() membaca data dari tabel dengan urutan kabkota, rumah_sakit, dan kelas_ruang_inap, memastikan tampilan konsisten setiap kali halaman dimuat. Jika koneksi basis data gagal,

DOI: https://doi.org/10.69693/ijmst.v4i3.10661
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

sistem menggunakan data fallback yang telah didefinisikan, sehingga antarmuka tetap dapat dioperasikan meskipun basis data tidak tersedia. Mekanisme graceful degradation ini penting untuk lingkungan produksi yang infrastruktur TI-nya tidak sepenuhnya stabil (Romney & Steinbart, 2021).

Validasi input pada lapisan ini mencakup dua tingkat. Pada tingkat pertama, setiap nilai string dibersihkan menggunakan fungsi trim() untuk menghilangkan spasi berlebih sebelum disimpan, sedangkan nilai numerik dikonversi paksa menjadi tipe integer menggunakan casting (int) untuk mencegah penyisipan nilai non-numerik. Pada tingkat kedua, baris yang seluruh kolomnya kosong secara otomatis diabaikan sebelum proses simpan maupun kirim dijalankan, sehingga mengurangi risiko pengiriman data kosong ke API eksternal yang dapat menimbulkan respons galat dari server tujuan (Rainer & Prince, 2021).

3.3.1. Aspek Keamanan dan Keterbatasan

Dari sisi keamanan, penggunaan prepared statement dengan bind_param pada seluruh operasi penyisipan data secara efektif menutup celah SQL injection konvensional (McLeod & Schell, 2019). Namun demikian, ditemukan satu kelemahan arsitektural yang perlu menjadi perhatian: token JWT Bearer dan session cookie API eksternal tertanam langsung (hardcoded) di dalam berkas kode sumber. Praktik ini berisiko terhadap kebocoran kredensial apabila berkas tidak dikelola dengan kontrol akses yang memadai pada server produksi (Rizki et al., 2022). Sebagai mitigasi pada tahap pengembangan lanjutan, kredensial sebaiknya dipindahkan ke berkas konfigurasi lingkungan (environment variable) yang terpisah dari kode sumber dan tidak disertakan dalam sistem kontrol versi.

Keterbatasan lain yang teridentifikasi adalah ketiadaan mekanisme refresh token otomatis ketika token JWT kedaluwarsa, sehingga operator harus memperbarui token secara manual pada kode sumber apabila masa berlaku token berakhir. Pengiriman data yang bersifat sekuensial (satu rekaman per permintaan HTTP) juga membatasi skalabilitas middleware apabila diterapkan pada institusi dengan volume data kamar yang jauh lebih besar dibandingkan RSKGM Provinsi Sumatera Selatan (Fadhillah et al., 2020).

3.4. Lapisan Integrasi API

Pengiriman data ke Dashboard Satu Data Provinsi Sumatera Selatan dilakukan melalui HTTP POST ke endpoint https://dashboard.sumselprov.go.id/index.php/api/submit-data-entri menggunakan library cURL PHP. Fungsi kirimDataKamar() mengemas setiap rekaman sebagai payload multipart/form-data dengan enam parameter: tabel, kabkota, rumah_sakit, kelas_ruang_inap, jumlah_kamar, dan jumlah_kamar_kosong. Autentikasi dilakukan dengan dua mekanisme: JWT Bearer Token pada header Authorization, dan session cookie pada header Cookie. Penggunaan dua mekanisme autentikasi ini sesuai dengan persyaratan endpoint API Dashboard Satu Data Provinsi Sumatera Selatan yang menggunakan CodeIgniter sebagai framework backend.

Tabel 2 merangkum parameter konfigurasi cURL yang digunakan. Timeout ditetapkan 30 detik untuk mengakomodasi latensi jaringan internet di lingkungan rumah sakit. CURLOPT_FOLLOWLOCATION diaktifkan untuk menangani kemungkinan redirect dari server. Setiap respons API direkam beserta HTTP status code dan pesan error cURL (jika ada), sehingga operator dapat mengetahui status pengiriman tiap rekaman secara granular.

Tabel 2. Parameter Konfigurasi cURL

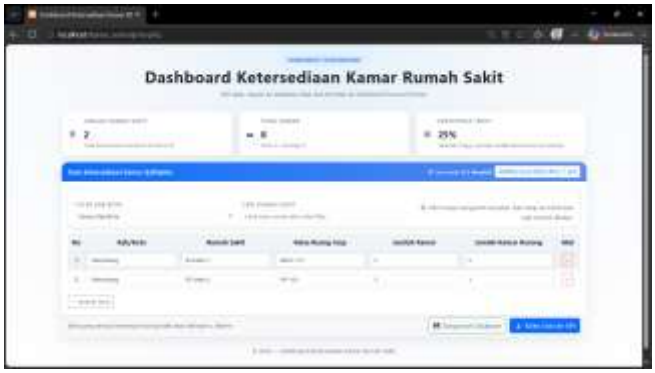
Parameter cURL	Nilai
CURLOPT_TIMEOUT	30 detik
CURLOPT_FOLLOWLOCATION	true
CURLOPT_HTTP_VERSION	HTTP/1.1
CURLOPT_CUSTOMREQUEST	POST
CURLOPT_POSTFIELDS	multipart/form-data
Header Authorization	Bearer JWT Token
Header Cookie	ci_session + sl-session

DOI: https://doi.org/10.69693/ijmst.v4i3.10661
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

3.5. Antarmuka Pengguna

Antarmuka pengguna dikembangkan menggunakan Bootstrap 5.3 dengan desain responsif yang dapat dioperasikan pada perangkat desktop maupun tablet. Terdapat tiga komponen utama antarmuka: (1) Kartu Ringkasan yang menampilkan jumlah rumah sakit unik, total kamar, serta persentase kamar terisi secara real-time berdasarkan data di tabel; (2) Tabel Data Editable dengan fitur filter berdasarkan kabupaten/kota dan pencarian nama rumah sakit; dan (3) Panel Hasil Pengiriman API yang menampilkan status HTTP dan detail respons raw per rekaman menggunakan komponen accordion Bootstrap.

Fitur auto-kirim diimplementasikan menggunakan JavaScript setInterval() dengan interval 3.600.000 milidetik (1 jam) yang secara programatik mengklik tombol "Kirim Data ke API". Pendekatan client-side ini dipilih karena lingkungan server produksi tidak mendukung konfigurasi cron job. Gambar 2 memperlihatkan tampilan antarmuka utama middleware.



Gambar 2. Antarmuka Dasbor Middleware

3.6. Hasil Pengujian

Pengujian black-box dilakukan terhadap seluruh fitur fungsional middleware. Tabel 3 merangkum hasil pengujian fungsional. Semua skenario uji menghasilkan output yang sesuai dengan ekspektasi.

Tabel 3. Hasil Pengujian Fungsional Middleware

No	Skenario Uji	Ekspektasi	Hasil
1	Tambah baris data baru	Baris muncul di tabel	Berhasil
2	Hapus baris data	Baris dihapus, nomor urut diperbarui	Berhasil
3	Filter kabupaten/kota	Baris yang tidak sesuai tersembunyi	Berhasil
4	Cari nama rumah sakit	Baris yang relevan ditampilkan	Berhasil
5	Simpan ke Database	Data tersimpan di MariaDB, DB sesuai tampilan	Berhasil
6	Kirim Data ke API (45 rekaman)	HTTP 200 pada semua rekaman	Berhasil
7	Akses tanpa koneksi DB	Fallback data tampil, tidak error fatal	Berhasil
8	Auto kirim aktif (simulasi interval)	Tombol kirim terpicu secara periodik	Berhasil

Pengujian integrasi API menggunakan 45 rekaman data kamar RSKGM Provinsi Sumatera Selatan menghasilkan HTTP 200 pada seluruh rekaman. Rata-rata waktu respons per rekaman adalah 1,2–2,8 detik, bergantung pada kondisi jaringan internet. Total waktu pengiriman 45 rekaman berkisar antara 54–126 detik. Nilai ini masih dalam

DOI: https://doi.org/10.69693/ijmst.v4i3.10661
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

batas yang dapat diterima mengingat pengiriman dilakukan secara sekuensial (satu rekaman per request) untuk mematuhi keterbatasan endpoint API yang tidak mendukung pengiriman batch (Satzinger et al., 2016).

3.7. Perbandingan Kondisi Sebelum dan Sesudah

Sebelum middleware dikembangkan, pemutakhiran data dilakukan secara manual oleh operator melalui antarmuka web Dashboard Satu Data, memerlukan rata-rata 15–20 menit per sesi pemutakhiran untuk seluruh data kamar RSKGM. Frekuensi pemutakhiran rata-rata adalah seminggu sekali, jauh dari idealnya setiap hari. Setelah middleware diimplementasikan, operator hanya perlu memeriksa dan menyesuaikan data pada tabel, lalu menekan satu tombol. Dengan fitur auto-kirim, pemutakhiran dapat dilakukan setiap jam tanpa intervensi manual. Waktu yang diperlukan operator berkurang dari 15–20 menit menjadi kurang dari 3 menit per sesi, dan konsistensi antara basis data lokal dengan dashboard provinsi terjamin (Rizki, 2022).

3.8. Perbandingan dengan Pendekatan Middleware Lain

Dibandingkan dengan pendekatan middleware berbasis framework seperti Laravel atau CodeIgniter yang umum digunakan dalam integrasi sistem informasi pemerintahan (Republik Indonesia, 2019), pendekatan single-file PHP yang digunakan dalam penelitian ini memiliki kelebihan dari sisi kemudahan deployment, karena tidak memerlukan proses instalasi dependency tambahan melalui Composer maupun konfigurasi routing yang kompleks. Kelebihan ini relevan khususnya bagi OPD dengan keterbatasan akses internet untuk mengunduh paket pustaka eksternal, maupun keterbatasan hak akses administratif pada server hosting bersama (shared hosting). Akan tetapi, pendekatan ini juga memiliki kekurangan dari sisi struktur kode, yaitu bercampurnya logika presentasi dan logika bisnis dalam satu berkas, yang secara jangka panjang dapat menyulitkan pemeliharaan apabila kompleksitas fitur middleware terus bertambah. Trade-off antara kemudahan deployment dan keterpeliharaan kode jangka panjang ini menjadi pertimbangan utama bagi institusi lain yang hendak mengadopsi pendekatan serupa.

3.9 Alur Logis Pemrosesan Data (Flowchart)

Pembagian tahapan di dalam alur logis pemrosesan data ini secara umum dapat dikelompokkan menjadi tiga fase utama, yaitu fase pra-pemrosesan (*pre-processing*), fase sinkronisasi lokal (*local synchronization*), dan fase transmisi eksternal (*external transmission*).

Pada fase pra-pemrosesan, koordinasi dimulai ketika operator memicu interaksi pada halaman komponen tabel dinamis. Skrip JavaScript yang berjalan di sisi klien (*client-side*) akan menangkap seluruh larik (*array*) data dari elemen input *editable*. Sebelum data dikirimkan ke server lokal melalui metode HTTP POST, fungsi skrip memastikan bahwa tidak ada manipulasi struktur elemen yang merusak integritas skema. Begitu data diterima oleh berkas monolitik PHP, mesin logika bisnis segera melakukan penyaringan berlapis. String nama daerah dan nama rumah sakit dibersihkan dari spasi tak terlihat di awal dan akhir karakter, sementara nilai numerik kapasitas kamar dipaksa melalui fungsi pencetakan tipe (*type casting*) untuk mengeliminasi potensi error akibat ketidakcocokan tipe data data pada skema basis data tujuan.

Masuk ke fase sinkronisasi lokal, sistem menerapkan pendekatan atomik melalui instruksi TRUNCATE. Pemilihan instruksi ini alih-alih menggunakan klausa DELETE standar didasarkan pada efisiensi performa, di mana operasi TRUNCATE akan langsung menyetel ulang indeks dan mengosongkan tabel pada tingkat penyimpanan tanpa mencatat log transaksi baris demi baris. Langkah radikal ini menjamin bahwa kondisi basis data lokal MariaDB akan selalu mencerminkan apa yang sedang dilihat oleh operator di layar dasbor saat itu juga. Setelah status tabel dipastikan kosong, blok perulangan foreach mengeksekusi parameter terikat (*bound parameters*) secara berulang. Teknik pengikatan parameter dengan penanda tipe "sssii" (*three strings and two integers*) menjadi fondasi utama dalam memblokade upaya injeksi kode SQL (*SQL Injection attack*) yang memanfaatkan celah masukan teks bebas oleh pengguna.

Akhirnya, pada fase transmisi eksternal, fokus dialihkan pada kesiapan modul cURL dalam menangani skenario integrasi multi-autentikasi. Karena API Dashboard Satu Data Provinsi Sumatera Selatan memberlakukan pengamanan ganda, middleware harus menyusun struktur header HTTP secara manual dan presisi. Token JWT dilekatkan pada sub-atribut *Authorization Bearer*, sedangkan parameter sesi pelacak CodeIgniter (ci_session) disuntikkan ke dalam sub-atribut *Cookie*. Pengiriman dilakukan secara sekuensial dengan memanfaatkan penanganan pengecualian (*exception handling*) berbasis status kode HTTP. Jika salah satu rekaman dari total 45 data kamar mengalami kegagalan pengiriman akibat *timeout* jaringan, sistem tidak akan menghentikan seluruh

DOI: https://doi.org/10.69693/ijmst.v4i3.10661
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)

proses (*graceful handling*), melainkan tetap melanjutkan pengiriman rekaman berikutnya sembari mencatat log kegagalan spesifik pada panel *accordion* antarmuka, sehingga operator dapat melakukan tindakan mitigasi atau pengiriman ulang secara parsial.

4. Kesimpulan

Middleware berbasis PHP yang dikembangkan dalam penelitian ini berhasil menjembatani sistem pengelolaan data lokal RSKGM Provinsi Sumatera Selatan dengan API Dashboard Satu Data Provinsi Sumatera Selatan. Arsitektur tiga lapis yang diterapkan memisahkan kepentingan presentasi, logika bisnis, dan integrasi API secara jelas sehingga meningkatkan keterpeliharaan kode. Mekanisme truncate-and-reinsert menjamin konsistensi penuh antara tampilan antarmuka dan isi basis data lokal, sedangkan penggunaan prepared statement melindungi sistem dari serangan SQL injection. Hasil pengujian menunjukkan bahwa middleware mampu mengirimkan 45 rekaman data kamar ke API eksternal dengan tingkat keberhasilan 100% (HTTP 200). Fitur auto-kirim berbasis JavaScript interval memungkinkan pemutakhiran periodik tanpa konfigurasi server tambahan. Penelitian ini membuktikan bahwa solusi integrasi ringan berbasis PHP tanpa framework eksternal dapat diterapkan secara efektif di lingkungan pemerintahan dengan infrastruktur TI terbatas, meskipun masih memiliki keterbatasan pada aspek keamanan penyimpanan kredensial dan skalabilitas pengiriman data. Untuk penelitian selanjutnya, disarankan untuk mengembangkan mekanisme pengiriman batch guna mengurangi total waktu pengiriman, mengintegrasikan middleware langsung dengan SIMRS menggunakan trigger basis data untuk pemutakhiran real-time, memindahkan kredensial autentikasi ke environment variable terpisah, serta menambahkan mekanisme refresh token otomatis untuk meningkatkan keamanan. Pengembangan antarmuka berbasis Progressive Web App (PWA) juga dapat dipertimbangkan untuk meningkatkan aksesibilitas pada perangkat mobile.

Reference

1. Diskominfo Provinsi Sumatera Selatan. (2024). *Panduan Penggunaan Dashboard Satu Data Provinsi Sumatera Selatan*. Palembang: Diskominfo Provinsi Sumatera Selatan.
2. Fadhillah, M. B. A., Anggoro, R., & Priyatno, A. M. (2020). Analisis Performa Link Stability dari Faktor Kecepatan untuk Dinamisasi Zona pada Zone Routing Protocol. *Jurnal Rekayasa Elektrika, 16*(3). https://doi.org/10.17529/jre.v16i3.16502
3. Jogiyanto, H. M. (2017). *Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Offset.
4. Kadir, A. (2018). *Pengenalan Sistem Informasi Edisi Revisi*. Yogyakarta: Andi Offset.
5. Kementerian Komunikasi dan Informatika Republik Indonesia. (2021). *Pedoman Umum Satu Data Indonesia*. Jakarta: Kementerian Komunikasi dan Informatika.
6. Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems: Managing the Digital Firm* (16th ed.). New Jersey: Pearson Education.
7. McLeod, R., & Schell, G. P. (2019). *Sistem Informasi Manajemen*. Jakarta: Salemba Empat.
8. O'Brien, J. A., & Marakas, G. M. (2019). *Management Information Systems* (11th ed.). New York: McGraw-Hill Education.
9. Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). New York: McGraw-Hill Education.
10. Rainer, R. K., & Prince, B. (2021). *Introduction to Information Systems* (8th ed.). New Jersey: John Wiley & Sons.
11. Republik Indonesia. (2019). Peraturan Presiden Nomor 39 Tahun 2019 tentang Satu Data Indonesia.
12. Rizki, R. M. (2022). Pengaruh e-commerce pada penjualan makanan khas Kampar. *Journal of Artificial Intelligence and Digital Business (RIGGS), 1*(1), 1–6. https://doi.org/10.31004/riggs.v1i1.3
13. Rizki, R. M., Saputra, L., & Sari, R. M. (2022). Optimasi pelayanan e-commerce menggunakan chatbot. *Journal of Artificial Intelligence and Digital Business (RIGGS), 1*(1), 7–12. https://doi.org/10.31004/riggs.v1i1.6
14. Romney, M. B., & Steinbart, P. J. (2021). *Accounting Information Systems* (15th ed.). New Jersey: Pearson Education.
15. Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2016). *Systems Analysis and Design in a Changing World* (7th ed.). Boston: Cengage Learning.
16. Sommerville, I. (2016). *Software Engineering* (10th ed.). Harlow: Pearson Education.

DOI: https://doi.org/10.69693/ijmst.v4i3.10661
Lisensi: Creative Commons Attribution 4.0 International (CC BY 4.0)