JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 11 April Tahun 2026
ISSN 3025-0919 (media online)
Hal 2875-2878

Implementasi RESTful API sebagai Layanan Backend Aplikasi Web Menggunakan Laravel dan MySQL

Muhammad Alwan Khoiri[1], Novan Wisnu Pratama[2*], Muhammad Yunus Rangkuti[3]

[1-4]Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Pamulang, Indonesia
Email: [1]muhammadalwankhoiri42@gmail.com, [2*]novanmantab@gmail.com, [3]dosen03156@unpam.ac.id

Abstrak− Dalam era pengembangan aplikasi web terkini, penting untuk memisahkan bagian backend dari frontend guna mencapai tingkat fleksibilitas dan skalabilitas yang optimal. Studi ini fokus pada penerapan RESTful API sebagai solusi backend dengan memanfaatkan framework Laravel serta basis data MySQL, menggunakan aplikasi portofolio sebagai contoh kasus. Tantangan utama dalam pendekatan monolitik terletak pada tingginya interdependensi antar elemen, yang sering kali menghambat proses integrasi dengan platform eksternal seperti aplikasi seluler. Pendekatan metodologis yang diterapkan adalah model Waterfall dalam System Development Life Cycle (SDLC), mencakup tahapan analisis persyaratan, desain sistem, pengkodean, serta evaluasi. Output dari penelitian ini berupa layanan web service berbasis API yang dapat mengelola operasi CRUD (Create, Read, Update, Delete) dan penanganan file gambar, dengan respons dalam format JSON. Verifikasi dilakukan melalui alat Postman untuk memastikan performa setiap titik akhir yang memadai. Pada akhirnya, pemanfaatan Laravel memfasilitasi konstruksi arsitektur RESTful yang andal dan efektif, sekaligus menyederhanakan sinkronisasi data lintas berbagai klien platform.

Kata Kunci: API; Backend; Laravel; Restful.

Abstract−*In the realm of contemporary web application development, separating the backend from the frontend is essential to achieve superior flexibility and scalability. This research aims to deploy a RESTful API as a backend service utilizing the Laravel framework and MySQL database, with a portfolio application serving as the case study. A common issue in monolithic development lies in the high interdependence among components, which complicates integration with other platforms such as mobile applications. The methodology employed here follows the Waterfall model of the System Development Life Cycle (SDLC), encompassing requirements analysis, system design, code implementation, and testing. The outcome is a web service based on an API capable of managing CRUD operations (Create, Read, Update, Delete) along with image file handling, delivering responses in JSON format. Testing was conducted via Postman to verify the proper functionality of each endpoint. Ultimately, leveraging Laravel simplifies the creation of a secure and efficient RESTful architecture, while facilitating seamless data integration across diverse client platforms.*

Keywords*: API; Backend; Laravel; Restful.*

1. PENDAHULUAN

Perkembangan bidang teknologi informasi kini mendorong transisi arsitektur perangkat lunak dari pendekatan monolitik ke arah model berorientasi layanan atau microservices. Pada arsitektur konvensional, bagian backend dan frontend sering kali terintegrasi dalam satu kode dasar yang sama, sehingga menimbulkan hambatan dalam hal perawatan dan pengembangan selanjutnya, khususnya saat aplikasi perlu diakses melalui berbagai platform seperti web dan mobile (Riawan et al., 2024).

Alternatif penyelesaian untuk masalah tersebut melibatkan penerapan Representational State Transfer (REST) API. API berbasis REST memfasilitasi pertukaran informasi antar sistem yang berbeda dengan cara yang ringan dan cepat melalui format JSON (JavaScript Object Notation). Dengan mengisolasi logika bisnis pada sisi server (backend) dari antarmuka pengguna (frontend), para pengembang bisa beroperasi secara mandiri, dan sistem pun menjadi lebih mudah untuk diperluas skalanya (Fielding, 2000). Lebih lanjut, inovasi ini mendukung sinkronisasi data yang lebih efektif di antara aplikasi dari platform yang beragam (Simbulan & Aryanto, 2024).

Studi ini memanfaatkan Framework Laravel sebagai instrumen utama dalam proses pengembangan. Laravel dipilih berkat fitur bawaan yang komprehensif untuk pembuatan API, termasuk Eloquent ORM, routing yang intuitif, serta perlindungan data. Basis data MySQL diterapkan sebagai sarana penyimpanan informasi karena ketangguhan dan performanya yang konsisten dalam mengelola hubungan data. Sasaran dari penelitian ini adalah menciptakan layanan

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 11 April Tahun 2026
ISSN 3025-0919 (media online)
Hal 2875-2878

backend yang handal untuk pengelolaan data portofolio, yang bisa dijangkau oleh berbagai aplikasi klien melalui protokol HTTP (Devi et al., 2024).

2. METODE

2.1 Tahapan Penelitian

Studi ini menerapkan pendekatan pengembangan perangkat lunak berdasarkan model Waterfall. Pemilihan model ini didasarkan pada karakteristiknya yang terstruktur dan bertahap, sehingga sangat sesuai untuk membangun sistem dengan spesifikasi yang telah ditetapkan secara eksplisit (Alif Fikri et al., 2024). Langkah-langkah yang dilaksanakan mencakup:

1. Analisis Kebutuhan: Mengidentifikasi kebutuhan data dan fungsi API yang diperlukan (manajemen portofolio dan pengunjung).
2. Perancangan Sistem: Membuat desain basis data menggunakan Entity Relationship Diagram (ERD) dan merancang endpoint API.
3. Implementasi: Penulisan kode program menggunakan PHP dengan Framework Laravel dan konfigurasi database MySQL (Sinlae et al., 2024).
4. Pengujian: Melakukan uji coba fungsionalitas API menggunakan aplikasi Postman.

2.2 Alat dan Bahan

Alat yang digunakan dalam penelitian ini meliputi:

1. Sistem Operasi: Windows 10/11.
2. Bahasa Pemrograman: PHP Versi 8.4.5.
3. Framework: Laravel Versi 12.
4. Database: MySQL (melalui LARAGON).
5. Editor: Visual Studio Code.
6. Tools Pengujian: Postman.

3. ANALISA DAN PEMBAHASAN

3.1 Perancangan Database

Sistem ini menggunakan database MySQL. Tabel utama yang dirancang adalah tabel portfolios untuk menyimpan data karya. Struktur tabel dapat dilihat pada Tabel 1.

Tabel 1. Struktur Tabel Portfolios

Field	Tipe Data	Keterangan
id	BigInt	Primary Key, Auto Increment
title	Varchar(255)	Judul Karya
client_name	Varchar(255)	Nama Klien
description	Text	Deskripsi Karya
date	Date	Tanggal Pengerjaan

Sumber: (Penulis, 2025)

3.2 Implementasi Kode

Pembuatan backend dilakukan melalui pengembangan controller yang diberi nama PortofolioController. Controller tersebut berperan dalam mengelola permintaan HTTP yang datang dari klien. Salah satu fitur penting adalah metode penyimpanan data (store), yang mengurus validasi masukan, penyimpanan informasi ke dalam basis data, serta unggah berkas gambar ke lokasi penyimpanan internal (storage).

Berikut adalah cuplikan kode implementasi metode store untuk menangani permintaan POST:

```
public function store(Request $request)
{
    // 1. Validasi Input
    $request->validate([
```

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 11 April Tahun 2026
ISSN 3025-0919 (media online)
Hal 2875-2878

```
        'title' => 'required|string',
        'client_name' => 'required|string',
        'images.*' => 'image|mimes:jpeg,png,jpg|max:2048'
    ]);

    // 2. Simpan Data Portofolio
    $portfolio = Portfolio::create([
        'title' => $request->title,
        'description' => $request->description,
        'client_name' => $request->client_name,
        'date' => $request->date,
    ]);

    // 3. Proses Upload Gambar
    if ($request->hasFile('images')) {
        foreach ($request->file('images') as $image) {
            $path = $image->store('portfolios', 'public');
            $portfolio->photos()->create(['photo_path' => $path]);
        }
    }

    return response()->json($portfolio->load('photos'), 201);
}
```

Selain metode penyimpanan, sistem juga mengintegrasikan mekanisme pencatatan pengunjung unik berdasarkan alamat IP dalam fungsi getPortfolioforIndex. Tujuannya adalah untuk mengawasi lalu lintas tanpa mengharuskan autentikasi pengguna, sebagaimana terlihat pada kode di bawah ini:

```
public function getPortfolioforIndex()
{
    $ip = request()->ip();
    // Cek duplikasi IP sebelum mencatat
    if (!PortfolioVisitor::where('visitor_ip', $ip)->exists()) {
        PortfolioVisitor::create(['visitor_ip' => $ip]);
    }
    // Mengembalikan data JSON
    $photos = Photo::with('portfolio')->get();
    return response()->json(['data' => $photos]);
}
```

3.3 API TESTING

Evaluasi dilakukan dengan pendekatan Black Box Testing menggunakan alat Postman untuk memeriksa respons HTTP dari server. Penekanan pengujian terletak pada memastikan format JSON yang dihasilkan selaras dengan skema basis data serta kode status HTTP yang tepat (seperti 200 OK untuk keberhasilan, atau 201 Created untuk entri baru) (Kartono et al., 2024).
Hasil pengujian *endpoint* dapat dilihat pada Tabel 2 di bawah ini:

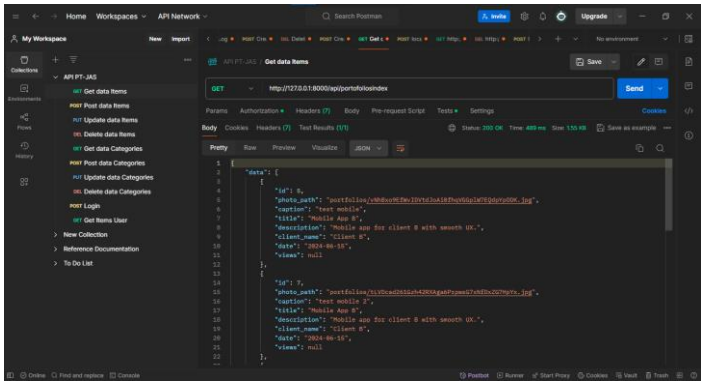
Tabel 2. Hasil Pengujian Endpoint API

No	Method	*Endpoint URL*	Fungsi	Hasil (Status Code)
1	GET	/api/portfolios/index	Menampilkan semua data	Berhasil (200 OK)
2	POST	/api/portfolios	Menambah data baru	Berhasil (201 *Created*)
3	GET	/api/portfolios/{id}	Menampilkan detail data	Berhasil (200 OK)
4	PUT	/api/portfolios/{id}	Mengupdate data	Berhasil (200 OK)
5	DELETE	/api/portfolios/{id}	Menghapus data	Berhasil (200 OK)

Sumber: (Penulis, 2025)

JRIIN : Jurnal Riset Informatika dan Inovasi
Volume 3, No. 11 April Tahun 2026
ISSN 3025-0919 (media online)
Hal 2875-2878

Gambar 1 menunjukkan contoh respon JSON yang berhasil diterima oleh Postman saat melakukan *request* data.



Gambar 1. Hasil *JSON GET API*

4. KESIMPULAN

Berdasarkan temuan dari proses implementasi dan evaluasi yang telah dilaksanakan, dapat ditarik kesimpulan bahwa Framework Laravel terbukti sangat efisien dalam membangun layanan RESTful API. Pemanfaatan elemen seperti Eloquent ORM dan API Resources di Laravel mempercepat manipulasi informasi ke dalam basis data MySQL. Layanan backend yang dihasilkan berhasil mengisolasi logika data dari antarmuka pengguna, sehingga informasi tersebut dapat digunakan oleh berbagai platform aplikasi, termasuk web dan mobile, melalui format JSON standar (Hadinata & Stianingsih, 2024).

Untuk kemajuan pengembangan di masa depan, direkomendasikan penerapan mekanisme keamanan autentikasi melalui JWT (JSON Web Token) atau Laravel Sanctum guna menjaga akses terhadap data rahasia. Lebih jauh, pengintegrasian sistem caching seperti Redis disarankan untuk mengoptimalkan kecepatan respons API ketika menghadapi volume trafik data yang besar.

REFERENCES

Alif Fikri, M., Rahma Saputri, D., Marcelino, M., Hoerudin, D., & Saifudin, A. (2024). Penerapan Model Waterfall Untuk Meningkatkan Kecepatan Dan Fleksibilitas Pengembangan Sistem Inventaris. *Buletin Ilmiah Ilmu Komputer Dan Multimedia (BIIKMA)*, *2*(1), 136–144. https://jurnalmahasiswa.com/index.php/biikma/article/view/976

Devi, N. K. A. S., Harsemadi, I. G., & Srinadi, N. L. P. (2024). Sistem Informasi Pengarsipan Surat Berbasis Website pada Kantor Perbekel Desa Pelapuan Menggunakan Framework Laravel. *Seminar Hasil Penelitian Informatika Dan Komputer (SPINTER)*, 496–501.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* [University of California, Irvine]. https://ics.uci.edu/~fielding/pubs/dissertation/top.htm

Hadinata, W., & Stianingsih, L. (2024). Analisis Perbandingan Performa RESTful API Antara Express.js Dengan Laravel Framework dengan JMeter. *Jurnal Informatika Dan Teknik Elektro Terapan*, *12*(1), 531–540. https://doi.org/10.23960/jitet.v12i1.3845

Kartono, F. K., Nursaadah, S., Nugroho, M. R., Tama, D. A., Mashudi, F. A., Wicaksono, A., & Nasir, M. (2024). Pengujian Black Box Testing Pada Sistem Website Osha Snack: Pendekatan Teknik Boundary Value Analysis. *Jurnal Kridatama Sains Dan Teknologi*, *6*(2), 754–766. https://doi.org/10.53863/kst.v6i02.1407

Riawan, F. A., Kusumo, D. S., & Selviandro, N. (2024). Implementation of REST API Architecture for Feelsquest Online Course Feature in Feelsbox Application Using Laravel Framework. *Jurnal Teknik Informatika (JUTIF)*, *5*(5), 1355–1364. https://doi.org/10.52436/1.jutif.2024.5.5.2493

Simbulan, R., & Aryanto, J. (2024). Implementasi REST API Web Services pada Aplikasi Sumber Daya Manusia. *Jurnal Indonesia: Manajemen Informatika Dan Komunikasi*, *5*(1), 552–560. https://doi.org/10.35870/jimik.v5i1.511