

Penerapan REST API Laravel sebagai Fondasi *Back-end* Aplikasi G-MOOC 4D

Iman Nurjaman¹, Fandy Setyo Utomo², Nandang Hermanto³

^{1,2,3}Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto

surel: ¹imannj69@gmail.com, ²fandy_setyo_utomo@amikompurwokerto.ac.id, ³nandanghermanto@amikompurwokerto.ac.id

Info Artikel

Sejarah artikel:

Diserahkan
18 Desember 2023
Diterima
15 Februari 2024
Diterbitkan
20 April 2024

Kata kunci:

Pendidikan inklusi, Learning Management System, Backend, REST API, Black box testing

ABSTRAK

Revolusi industri keempat, mengalami kemajuan teknologi dan memengaruhi setiap aspek kehidupan. Pendidikan memiliki peran besar dalam mengembangkan individu secara fisik, intelektual, dan emosional. Perkembangan teknologi menjadi pendidikan daring atau Learning Management System (LMS). Pendidikan seperti yang disebutkan penting untuk pengembangan manusia. Pendidikan inklusi diatur oleh Permendiknas No.70/2009, mendukung kesetaraan dalam pembelajaran. Anak tunanetra memerlukan bahan ajar khusus, seperti tulisan Braille. Aplikasi MOOC, seperti G-MOOC 4D, dengan fitur inklusi, membuka akses belajar bagi penyandang. Backend, sebagai logika aplikasi di server, krusial dalam aplikasi berbasis web. REST API, perlu dalam pembangunan aplikasi modern. REST API mengelola pertukaran data efisien dan aman dengan format JSON. REST API diharapkan meningkatkan efisiensi dan keamanan aplikasi G-MOOC 4D. Pengembangan sistem atau sering disebut Software Development Life Cycle (SDLC) adalah suatu proses yang digunakan untuk mengembangkan atau mengubah sistem perangkat lunak dengan memanfaatkan model-model dan metodologi yang telah digunakan sebelumnya oleh para profesional dalam pengembangan sistem perangkat lunak. Peneliti berhasil membangun REST API menggunakan framework Laravel, yang menghasilkan sejumlah titik akses (API endpoints) yang digunakan oleh tim frontend untuk melakukan interaksi dan mengakses data dari backend sistem secara efisien dan aman. Pengujian menggunakan metode black box melalui aplikasi Postman dalam penelitian ini berhasil mencapai kesuksesan menyeluruh pada setiap titik akses (endpoint) API yang telah dibangun).

Penulis koresponden:

Iman Nurjaman
Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Purwokerto
Email: imannj69@gmail.com

1. PENDAHULUAN

Pada Revolusi industri keempat yang saat ini dialami melalui kemajuan teknologi membawa pengaruh pada setiap aspek kehidupan kita. Bahkan sebelum itu, kemajuan teknologi tidak diragukan lagi telah mengubah sejumlah aspek kehidupan kita, khususnya di bidang pendidikan. Pendidikan berpengaruh besar pada perkembangan individu dalam berbagai aspek kepribadian dan potensi, termasuk fisik, intelektual, emosional, sosial, dan spiritual, sesuai dengan konteks sosial budaya dan lingkungan fisiknya [1]. Pendidikan dapat diperoleh dari lembaga formal dan nonformal [2]. Pendidikan adalah untuk memanusiakan manusia, manusia perlu dibantu agar dia bisa menjadi manusia, karena manusia itu adalah sifat, banyak manusia yang bukan manusia, karena salah satu pertanda manusia adalah dengan dia berpikir [3]. Berdasarkan pendapat tersebut tidak hanya anak normal, anak yang berkebutuhan khusus juga mempunyai hak yang sama atas pendidikan.

Pendidikan inklusi bagi anak-anak berkebutuhan khusus, seperti anak tunanetra, yang sering kali mengalami kesulitan dalam memperoleh akses pendidikan [2]. Pendidikan inklusi, sebagaimana diatur dalam Permendiknas No.70 tahun 2009, bertujuan memberikan peluang pendidikan yang merata bagi individu dengan beragam kelainan fisik, emosional, mental, dan sosial, serta potensi kecerdasan dan bakat istimewa, sesuai dengan kebutuhan dan kemampuannya [4]. Anak tunanetra, yang mengalami gangguan fungsi penglihatan, memerlukan dukungan khusus dalam pembelajaran, seperti menggunakan tulisan braille untuk membantu orientasi mereka. Tulisan braille, berbasis titik timbul yang dapat dibaca melalui sentuhan, menjadi kunci dalam proses pembelajaran mereka [5]. Dalam konteks ini, bahan pembelajaran yang memadai dan penggunaan media pembelajaran yang sesuai sangat penting untuk mendukung pembelajaran anak tunanetra.

Seiring berkembangnya teknologi, pendidikan kini tidak hanya dapat dilakukan secara tatap muka, namun juga dapat dilakukan secara *online* [6]. Saat ini sistem pembelajaran *online* menggunakan *Learning Management System* (LMS) telah diterapkan secara luas di berbagai wilayah di dunia. Pada dasarnya LMS merupakan perangkat lunak atau *software* yang digunakan untuk keperluan administrasi, dokumentasi, dan pelaporan yang berkaitan dengan berbagai aspek seperti proses belajar mengajar serta kegiatan lain yang berlangsung secara *online* atau dengan koneksi Internet [7]. LMS berisi materi-materi yang mencakup kompetensi pedagogik dan profesional, disusun dengan beragam media, seperti teks, animasi, video, audio, serta efek khusus, materi-materi ini disediakan sebagai tambahan (*supplement*) dan pengayaan (*enrichment*) untuk mendukung perkembangan kompetensi pembelajar [6].

Pembelajaran daring yang menghubungkan pelajar secara *online* telah menjadi bagian yang lumrah dalam dunia pendidikan. Namun, konsep *Massive Open Online Course* (MOOC) mengambil langkah lebih jauh dengan menyajikan pembelajaran dalam skala besar secara global [8]. MOOC menyediakan akses terbuka melalui situs web dengan tujuan menciptakan partisipasi interaktif dalam jumlah besar [9]. Meskipun demikian, MOOC menghadapi tantangan dalam memfasilitasi aksesibilitas bagi penyandang disabilitas. Kebanyakan platform MOOC belum sepenuhnya memperhatikan kebutuhan aksesibilitas untuk mendukung partisipasi penyandang disabilitas dalam pembelajaran *online*. Diperlukan pengembangan aplikasi MOOC yang dapat memberikan pengalaman belajar yang lebih inklusif bagi mereka.

Di Indonesia sudah terdapat *Massive Open Online Course* (MOOC) atau platform pembelajaran *online* yang menerapkan sistem gamifikasi di dalam proses pembelajarannya, yaitu *Gamification Massive Open Online Course* (G-MOOC). Platform G-MOOC merupakan platform MOOC yang dibangun dengan menggunakan pendekatan gamifikasi. Gamifikasi di perlukan karena faktor pentingnya tingkat interaksi antara siswa dengan guru atau dengan siswa lainnya [10]. Saat ini G-MOOC masih kurang memperhatikan aksesibilitas bagi penyandang disabilitas, terutama tunanetra dalam proses pembelajaran. Diperlukan pengembangan aplikasi yang dapat menyediakan pengalaman belajar yang lebih inklusif dan ramah bagi penyandang disabilitas. Hal ini akan memungkinkan mereka untuk mengakses materi pembelajaran dengan lebih mudah dan efektif, sesuai dengan kebutuhan mereka.

Gamification Massive Open Online Course for disability (G-MOOC 4D) menggunakan konsep MOOC berbasis *website* yang menerapkan *gamification* dalam proses pembelajarannya. *Gamification* atau gamifikasi merupakan teknik populer yang memanfaatkan aplikasi dalam interaksi manusia-komputer (HCI), psikologi, dan pengembangan *game* digital untuk merangsang keterlibatan dan motivasi manusia [11]. Dengan

menerapkan *for disability* (4D), Aplikasi ini akan diadaptasi secara khusus untuk menyediakan aksesibilitas bagi penyandang disabilitas khususnya tunanetra, sehingga menciptakan lingkungan pembelajaran yang inklusif. Melalui G-MOOC 4D, mereka dapat belajar secara mandiri dan fleksibel tanpa terbatas oleh keterbatasan fisik atau geografis. Mereka akan merasa lebih tertantang dan terlibat aktif dalam aktivitas pembelajaran yang disajikan secara interaktif dan menarik.

Dalam pengembangan aplikasi G-MOOC 4D berbasis *website*, prosesnya dibagi menjadi empat tim yang berperan masing-masing. Pertama, tim *User Interface* (UI) bertanggung jawab merancang antarmuka pengguna. Kedua, tim *front-end* bertugas mewujudkan desain antarmuka yang telah dirancang oleh tim UI. Ketiga, tim *Artificial Intelligence* (AI) memiliki peran penting dalam menyediakan aksesibilitas bagi pengguna, seperti *login* menggunakan pengenalan wajah dan interaksi dengan sistem menggunakan media suara. Terakhir, tim *back-end* memiliki tugas menghubungkan *front-end* dan AI dengan server. Sistem *back-end* bertujuan untuk mengembangkan bagian dari aplikasi yang tidak terlihat oleh pengguna, yaitu mengatur logika, *database*, dan fungsionalitas inti dari sebuah sistem.

Penelitian ini berfokus pada pengembangan sistem *back-end*. *Back-end* merujuk pada aplikasi dan skrip yang berjalan di server atau bagian belakang suatu sistem. Secara efektif, *back-end* berperan sebagai wadah logika fungsional dan fungsi penting dari aplikasi perangkat lunak atau sistem informasi. Fungsi dari sistem *back-end* adalah memastikan bahwa permintaan data atau layanan yang dikirimkan oleh antarmuka pengguna (*front-end*) diproses dan dihasilkan sesuai dengan standar yang telah ditetapkan. Komponen utama dalam pengembangan perangkat lunak, seperti aplikasi, basis data, integrasi data aplikasi, *Representational State Transfer Application Programming Interface* (REST API), serta operasi *back-end* lainnya, merupakan bagian integral dari sistem *back-end* [12].

Dewasa ini penggunaan REST API menjadi semakin populer dalam pengembangan aplikasi *online*. Pembuatan serta pengujian REST API server yang berfungsi untuk melakukan pertukaran data melalui protokol HTTP dengan hasil respons dalam format JSON [12]. Penggunaan REST API yang dapat meningkatkan efisiensi pertukaran data dan transaksi, berfokus pada keandalan aplikasi *back-end* [13]. Perancangan *back-end*, digunakan pendekatan interaktif dan inkremental untuk membangun API yang mengadopsi arsitektur REST. *Endpoint-nya* diberi nama dengan sederhana dan metode HTTP yang mudah diakses [14].

Berdasarkan penelitian yang disebutkan, REST API telah menjadi pendekatan penting dalam pengembangan perangkat lunak melakukan pertukaran data. Fokus utamanya adalah pada efisiensi, keandalan, keamanan data, dan dokumentasi terstruktur. REST API memungkinkan aplikasi berkomunikasi secara efektif melalui Internet dan telah menjadi fondasi banyak sistem saat ini. REST API dapat mengelola *front-end* dan *back-end* secara efisien. Sistem *back-end* akan dibangun dalam bentuk server *back-end* berbasis REST API. Dalam sistem REST, terdapat empat jenis permintaan HTTP yang umum digunakan: GET, POST, PUT, dan DELETE. Metode HTTP sederhana ini melakukan operasi CRUD *database* seperti *Create*, *Read*, *Update*, dan *Delete* dengan memanfaatkan format data JSON [12].

Berdasarkan latar belakang yang telah diuraikan, Penelitian ini akan memfokuskan pada pengembangan REST API sebagai bagian dari *back-end* dengan harapan dapat meningkatkan efisiensi serta keamanan dalam pertukaran data dalam aplikasi G-MOOC 4D.

2. METODE

2.1. Metode Pengumpulan Data

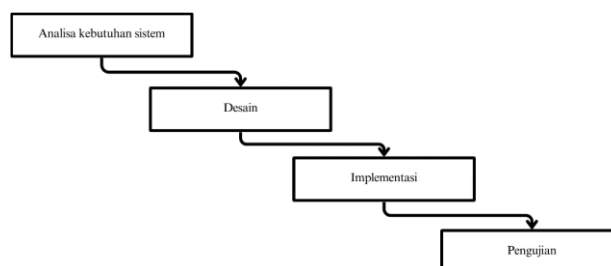
Metode pengumpulan data yang digunakan dalam penelitian ini yaitu dengan menggunakan metode wawancara dan metode studi pustaka.

2.2. Metode pengembangan aplikasi

Pengembangan sistem atau sering disebut *Software Development Life Cycle* (SDLC) adalah suatu proses yang digunakan untuk mengembangkan atau mengubah sistem perangkat lunak dengan memanfaatkan model-

model dan metodologi yang telah digunakan sebelumnya oleh para profesional dalam pengembangan sistem perangkat lunak [16].

Salah satu dari berbagai metode pengembangan perangkat lunak adalah pendekatan metode *waterfall*. Metode yang digunakan dalam penelitian ini menggunakan model *waterfall*, di mana Model ini adalah paradigma tradisional yang digunakan dalam pengembangan perangkat lunak. (H. Kurniawan dkk., 2020), yang terbagi atas empat tahap, yaitu:



Gambar 1. Metode pengembangan aplikasi

1) Analisa kebutuhan sistem (*requirements*)

Analisa kebutuhan sistem adalah tahap awal yang sangat penting dalam pengembangan perangkat lunak. Tujuannya adalah untuk memahami dan mendefinisikan kebutuhan bisnis dan fungsional yang harus dipenuhi oleh sistem yang akan dikembangkan. Data hasil wawancara digunakan sebagai sumber utama untuk menganalisis kebutuhan sistem. Melalui proses ini, informasi yang diperoleh diharapkan dapat mengurangi risiko, mencegah potensi masalah di masa mendatang, serta memastikan bahwa sistem yang dikembangkan sesuai dengan harapan dari perancang G-MOOC 4D.

2) Desain

Dalam proses ini, sistem dirancang menggunakan Data Flow Diagram (DFD) dan Entity Relationship Diagram (ERD). DFD, juga dikenal sebagai Diagram Arus Data (DAD), adalah model logika data yang menggambarkan asal-usul, arah, dan interaksi data dalam sistem, termasuk proses, penyimpanan data, dan interaksi antara data yang tersimpan dengan proses yang diterapkan pada data tersebut [17]. ERD atau Entity Relationship Diagram adalah representasi struktural yang digunakan untuk merancang database. ERD menggambarkan data yang disimpan dalam sistem dan batasan-batasannya. Komponen utama ERD melibatkan entitas (*entity set*), relasi (*relationship set*), dan kendala-kendala (*constraints*). [18].

3) Implementasi

Setelah analisis kebutuhan sistem dan tahap desain, selanjutnya adalah implementasi, ketika desain yang dikembangkan selama tahap desain harus diubah menjadi bentuk aplikasi [19]. Pada tahap ini, dilakukan implementasi dari Data Flow Diagram (DFD) dan Entity Relationship Diagram (ERD) yang telah dibuat sebelumnya. Proses implementasi melibatkan penelitian instruksi program berdasarkan logika dan struktur yang telah diatur dalam DFD dan ERD. Tahap ini merupakan titik krusial dalam konversi desain menjadi sebuah aplikasi yang berjalan secara efektif sesuai dengan spesifikasi yang telah ditetapkan sebelumnya.

4) Pengujian

Pada tahap ini, pengujian dilakukan menggunakan metode Black Box, di mana fokus utamanya adalah pada fungsionalitas eksternal perangkat lunak tanpa memperhatikan struktur internal atau logika program, tetapi lebih menekankan pada *input* dan *output* yang dihasilkan [20].

Dalam konteks ini, pengujian menggunakan *software* Postman sebagai sarana untuk mengirimkan permintaan HTTP ke API yang relevan dan mengevaluasi respons yang diterima. Tujuannya adalah untuk memeriksa apakah fungsi-fungsi dalam program beroperasi sesuai dengan persyaratan yang diperlukan oleh *front-end*. Hasil dari pengujian Black Box ini digunakan untuk

memastikan bahwa interaksi antara *front-end* dan *back-end* berjalan dengan lancar, sebagaimana yang diharapkan tanpa memerlukan pengetahuan internal tentang struktur kode.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Kebutuhan Sistem

Aplikasi G-MOOC 4D memiliki fitur kelas dengan materi pembelajaran berupa video dan kuis. Sistem ini memiliki penilaian berupa poin dari keaktifan menyelesaikan materi dan nilai dari kuis yang dikerjakan. Terdapat dua akses: admin dengan hak akses tertinggi yang dapat mengakses hampir semua fitur, dan pengguna biasa dengan hak akses terbatas. Spesifikasi kebutuhan sistem meliputi:

1) Kebutuhan Admin

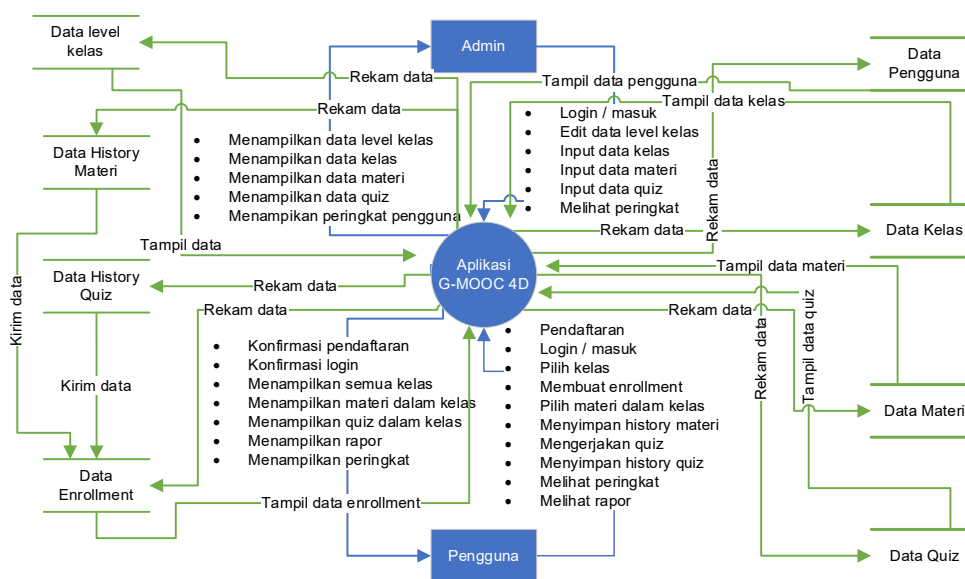
Admin memiliki beberapa kebutuhan sistem yaitu: (1) *Login*. (2) Mencari, dan mengubah level kelas. (3) Membuat, mencari, mengubah dan menghapus kelas. (4) Membuat, mencari, mengubah dan menghapus materi. (5) Membuat, mencari, mengubah dan menghapus mengelola kuis. (6) Melihat *leaderboard* atau peringkat pengguna.

2) Kebutuhan Pengguna

Pengguna memiliki beberapa kebutuhan sistem yaitu: (1) Register. (2) *Login*. (3) Verifikasi email. (4) Reset *password*. (5) Melihat kelas. (6) Melihat materi. (7) Mengerjakan *quiz*. (8) Membuat histori kelas, materi, dan *quiz*. (9) Melihat *leaderboard* atau peringkat. (10) Melihat nilai yang diperoleh berdasarkan kelas yang sudah diikuti (rapor).

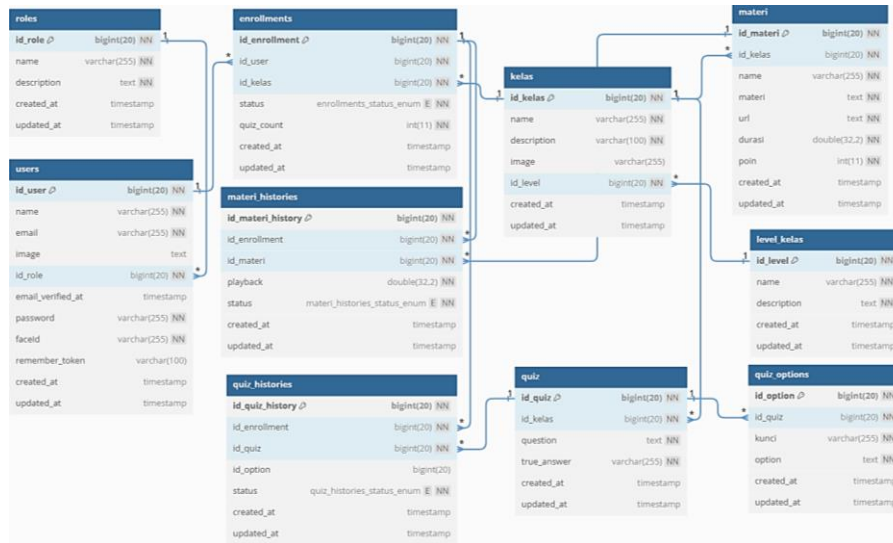
3.2 Desain

1) Data Flow Diagram (DFD)



Gambar 2. Data Flow Diagram

2) Entity Relationship Diagram (ERD)

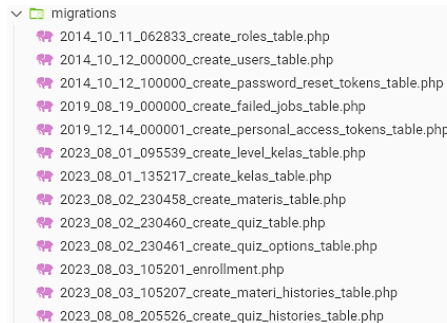


Gambar 3. Entity Relationship Diagram

3.3 Implementasi

Laravel mempunyai konsep MVC (*Model-View-Controller*). dalam penelitian ini, peneliti hanya menggunakan model, *controller*, dan beberapa fitur yang tersedia dari Laravel, yaitu *migration*, *seeder*, *Notification*, *middleware* dan *routes* dalam membangun REST API. Berikut ini adalah beberapa struktur *project* Laravel.

1) *Migrations*

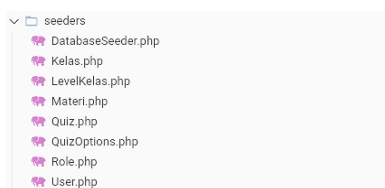
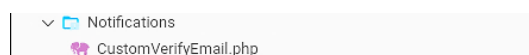


Gambar 4. Struktur *Migrations*

Gambar 4 Struktur migrasi yang dibuat berdasarkan ERD adalah untuk tabel *roles*, *level_kelas*, *kelas*, *materi*, *quiz*, *quiz_option*, *enrollment*, *materi_histories*, *quiz_histories*, dan ada perubahan pada atribut dalam tabel *user*. Sisanya adalah entitas *default* yang sudah ada dalam Laravel. Migrasi ini bertujuan untuk membuat tabel-tabel dalam basis data sesuai dengan struktur entitas yang terdefinisi dalam ERD.

2) *Seeders*

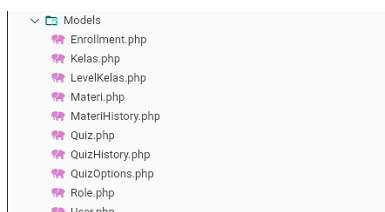
Gambar 5 menjelaskan tentang sejumlah *seeder* yang telah dibuat. *Seeder-seeder* ini bertujuan untuk mengisi data *dummy* pada setiap tabel dalam basis data.

Gambar 5. Struktur *Seeders*3) *Notifications*Gambar 6. Struktur *Notifications*

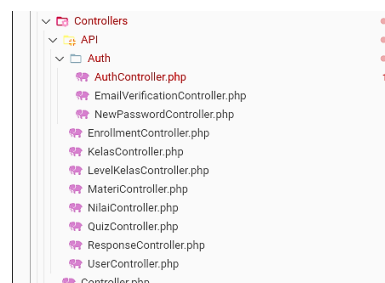
Gambar 6 menggambarkan adanya notifikasi khusus (*custom notification*) yang dapat disesuaikan sesuai kebutuhan. Notifikasi ini diatur untuk dikirim sebagai email verifikasi kepada alamat email pengguna.

4) *Mail*Gambar 7. Struktur *Mail*

Gambar 7 menjelaskan adanya *custom* Mail yang telah disesuaikan agar berfungsi dalam mengirimkan kode reset *password*, yang memungkinkan pengguna untuk melakukan pengaturan ulang *password* ketika lupa.

5) *Models*Gambar 8. Struktur *Models*

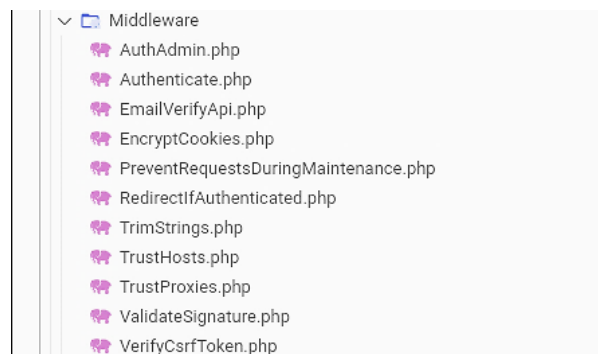
Gambar 8 menjelaskan bahwa beberapa model berfungsi untuk mendefinisikan atribut dan relasi dari suatu tabel. Melalui penggunaan *Eloquent* model, memungkinkan untuk dengan mudah melakukan relasi antara model-model yang berbeda.

6) *Controllers*Gambar 9. Struktur *Controllers*

Bagian ini merupakan bagian di mana manipulasi *database* dilakukan dengan memanggil model. *Controller* ini terlibat dalam proses *Create*, *Read*, *Update*, dan *Delete* (CRUD), di mana setiap langkah

melibatkan pemanggilan model yang sesuai dengan fungsionalitas yang diinginkan dalam *controller* tersebut.

7) *Middleware*



Gambar 10. Struktur *Middleware*

Gambar 10 menggambarkan Struktur *Middleware* yang telah dibuat. Beberapa *middleware* yang disertakan yaitu: (1) *AuthAdmin Middleware* ini membatasi akses hanya untuk pengguna dengan peran sebagai admin. (2) *EmailVerifyApi Middleware* ini mengatur akses untuk pengguna yang belum melakukan verifikasi email. Sisanya adalah *middleware default* yang disediakan oleh Laravel. Setiap *middleware* memiliki peran spesifik dalam mengontrol akses dan memberikan respons dalam format JSON.

8) *Routes*

Routes berupa API Laravel dengan *protocol* HTTP menggunakan *method* HTTP. API diolah di bagian *file* *api.php*, bagian ini menerapkan *middleware* sebagai hak akses yang ditentukan sebelumnya. Berikut kode program dalam *file* *api.php*:

```
Route::controller(KelasController::class)->group(function () {
    Route::get('kelas', 'index');
    Route::get('search/kelas', 'search');
    Route::get('kelas/name/{name}', 'showByName');

    Route::middleware('auth.admin')->group(function () {
        Route::post('kelas/create', 'store');
        Route::get('kelas/{id_kelas}', 'show');
        Route::post('kelas/update/{id_kelas}', 'update');
        Route::delete('kelas/delete/{id_kelas}', 'destroy');
    });
});
```

Gambar 11. *Source code file* *api.php*

3.4 Pengujian

Pengujian ini menggunakan metode Black Box, yang berfokus pada pengujian fungsional dari sistem tanpa memerhatikan struktur internal atau kode program. Fokus utamanya adalah memastikan kesesuaian antara sistem yang telah diimplementasikan.

Tabel 1. Pengujian CRUD kelas

No.	<i>Endpoint</i>	Skenario pengujian	Hasil yang diharapkan	Hasil pengamatan	keterangan
1.	<i>/api/kelas/create</i>	Mengirimkan permintaan POST dengan <i>bearer token</i> , <i>name</i> , <i>image</i> ,	Mendapatkan respons 201 <i>Created</i> dengan pesan "Kelas berhasil dibuat".	Sesuai dengan yang diharapkan	Diterima

		<i>id_level</i> , dan <i>description</i> .			
2.	<code>/api/kelas/{id_kelas}</code>	Mengirimkan permintaan GET dengan <i>bearer token</i> dan <i>id_kelas</i> .	Mendapatkan respons 200 OK dengan data kelas yang sesuai dengan ID yang diminta.	Sesuai dengan yang diharapkan	Diterima
3.	<code>/api/kelas/update/{id_kelas}</code>	Mengirimkan permintaan POST dengan <i>bearer token</i> , <i>id_kelas</i> , <i>name</i> , <i>image</i> , <i>id_level</i> , dan <i>description</i> .	Mendapatkan respons 201 <i>Created</i> dengan pesan "Kelas berhasil <i>diupdate</i> ".	Sesuai dengan yang diharapkan	Diterima
4.	<code>/api/kelas/delete/{id_kelas}</code>	Mengirimkan permintaan DELETE dengan <i>bearer token</i> dan <i>id_kelas</i> .	Mendapatkan respons 200 OK dengan pesan "Kelas berhasil dihapus".	Sesuai dengan yang diharapkan	Diterima

4. KESIMPULAN

Untuk menarik kesimpulan dari seluruh uraian yang telah dibahas pada bab sebelumnya, berikut adalah rangkuman kesimpulan yang dapat diambil: (1) Penelitian ini berhasil membangun REST API menggunakan *framework* Laravel, yang menghasilkan sejumlah titik akses (*API endpoints*) yang digunakan oleh tim *front-end* untuk melakukan interaksi dan mengakses data dari *back-end* sistem secara efisien dan aman. (2) Pengujian menggunakan metode Black Box melalui aplikasi Postman dalam penelitian ini berhasil mencapai kesuksesan menyeluruh pada setiap titik akses (*API endpoint*) yang telah dibangun.

REFERENSI

- [1] I. Kurniawan, "Implementasi pendidikan bagi siswa tunanetra di Sekolah Dasar Inklusi," *Edukasi Islami: Jurnal Pendidikan Islam*, vol. 4, no. 08, pp. 1044–1060, Oct. 2017, doi: 10.30868/EI.V4I08.77.
- [2] J. C. Pratiwi, "Sekolah inklusi untuk anak berkebutuhan khusus: tanggapan terhadap tantangan kedepannya," in *PROSIDING SEMINAR NASIONAL PENDIDIKAN*, Surakarta, Nov. 2015.
- [3] M. R. Abdul, T. Rostitawati, R. Podungge, and M. Arif, "Pembentukan akhlak dalam memanusiaikan manusia: perspektif Buya Hamka," *Pekerti: Jurnal Pendidikan Islam dan Budi Pekerti*, vol. 2, no. 1, pp. 79–99, Feb. 2020.
- [4] R. Amrulloh, I. Rahadi, R. R. Yuliatin, Y. A. Hadi, and S. Ramdhani, "Implementation of e-learning based on learning sanagement system using discovery learning method for disabilities students," *Jurnal Pendidikan Bahasa dan Sastra Indonesia*, vol. 4, no. 1, pp. 1–11, May 2021.
- [5] K. Nisa, S. Mambela, and L. I. Badiah, "Karakteristik dan kebutuhan anak berkebutuhan khusus," *Jurnal Abadimas Adi Buana*, vol. 2, no. 1, pp. 33–40, Jul. 2018, doi: 10.36456/abadimas.v2.i1.a1632.
- [6] D. N. Andriani and A. D. Daroin, "Analisis faktor keberhasilan pembelajaran menggunakan Learning Management System (LMS)," *JPE (Jurnal Pendidikan Edutama)*, vol. 9, no. 1, pp. 1–10, 2022.
- [7] F. S. Anggriawan, "Pengembangan Learning Management System (LMS) sebagai media pembelajaran untuk sekolah menengah sederajat," *Jurnal Tata Rias*, vol. 9, no. 2, pp. 1–10, 2019.
- [8] H. Praherdhiono, E. P. Adi, and Y. Prihatmoko, "Konstruksi demokrasi belajar berbasis kehidupan pada implementasi LMS Dan MOOC," *Edcomtech: Jurnal Kajian Teknologi Pendidikan*, vol. 3, no. 1, pp. 21–28, May 2018.



- [9] J. Husna, "Implementasi MOOCs di pendidikan ilmu Perpustakaan dan informasi (sebuah peluang dan tantangan di Indonesia)," *Anuva: Jurnal Kajian Budaya, Perpustakaan, dan Informasi*, vol. 3, no. 3, pp. 247–256, Nov. 2019.
- [10] R. eko Saputro, D. Komarasary, and R. F. Sari, "Pengaruh elemen rewards terhadap keterlibatan mahasiswa dalam mengikuti pembelajaran online pada platform G-MOOC," *CogITo Smart Journal*, vol. 8, no. 2, pp. 308–320, Dec. 2022, doi: 10.31154/COGITO.V8I2.408.308-320.
- [11] S. Hafizi, M. Hanafiah, A. Hakim, A. Majid, K. Shukri, and M. Teh, "Gamifikasi dalam pendidikan: satu kajian literatur," *Asian People Journal (APJ)*, vol. 2, no. 2, pp. 31–41, Dec. 2019.
- [12] H. F. Herdiyatomoko, "Desain sistem backend berbasis REST API menggunakan Framework Laravel 7," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 2, pp. 136–144, 2022.
- [13] R. E. P. Rizaldy, U. A. Ahmad, and B. Dirgantoro, "Implementasi REST API pada pengembangan aplikasi backend untuk platform kursus online (Growup)," *e-Proceeding of Engineering*, vol. 10, no. 1, pp. 404–409, 2023.
- [14] I. A. Faruqi, S. F. S. Gumilang, and M. A. Hasibuan, "Perancangan back-end aplikasi Rumantara dengan gaya arsitektur REST menggunakan metode Iterative Incremental," *e-Proceeding of Engineering*, vol. 5, no. 1, pp. 1411–1417, 2018.
- [15] M. N. Adlini, A. H. Dinda, S. Yulinda, O. Chotimah, and S. J. Merliyana, "Metode penelitian Kualitatif studi pustaka," *Edumaspul - Jurnal Pendidikan*, vol. 6, no. 1, pp. 974–980, 2022.
- [16] H. Larasati and S. Masripah, "Analisa dan perancangan sistem informasi pembelian GRC dengan metode Waterfall," *Jurnal Pilar Nusa Mandiri*, vol. 13, no. 2, pp. 193–198, Sep. 2017.
- [17] F. Soufitri, "Perancangan data flow diagram untuk sistem informasi sekolah (studi kasus pada SMP Plus Terpadu)," *Ready Star*, vol. 2, no. 1, pp. 240–246, Nov. 2019.
- [18] M. L. A. Latukolan, A. Arwan, and M. T. Ananta, "Pengembangan Sistem Pemetaan Otomatis Entity Relationship Diagram Ke Dalam Database," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 4, pp. 4058–4065, Feb. 2019.
- [19] H. Kurniawan, W. Apriliah, I. Kurniawan, and D. Firmansyah, "Penerapan metode Waterfall dalam perancangan sistem informasi penggajian pada SMK Bina Karya Karawang," *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, vol. 14, no. 4, pp. 159–169, Jan. 2020, doi: 10.35969/interkom.v14i4.58.
- [20] A. Saifudin, A. Fauzi, B. A. Prihantoro, D. E. Putra, and R. Ashiddiqia, "Pre pengujian menggunakan Black Box dengan teknik Equivalence Partitioning pada sistem penerbitan buku," *JRIIN: Jurnal Riset Informatika dan Inovasi*, vol. 1, no. 1, pp. 282–286, Jul. 2023.