

RESEARCH ARTICLE

Penerapan Arsitektur REST API pada Aplikasi Backend Manajemen Informasi Fakultas Industri Kreatif (MI-FIK) Universitas Telkom

Rafly Farhandika, Mira Kania Sabariah* and Monterico Adrian

Fakultas Informatika, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

* Corresponding author: mirakania@telkomuniversity.ac.id

Received on 25 April 2024; accepted on 29 May 2024

Abstrak

MI-FIK merupakan proyek lanjutan dari perancangan purwarupa perangkat lunak pengelolaan informasi Fakultas Industri Kreatif. MI-FIK merupakan perangkat lunak yang dirancang untuk mempermudah dosen, staff, dan mahasiswa Fakultas Industri Kreatif untuk memperoleh informasi di lingkungan Fakultas Industri Kreatif. Perangkat lunak MI-FIK merupakan perangkat lunak yang berbasis mobile application yang pada pengembangannya dibagi menjadi 2 sisi, yaitu pengembangan dari sisi *mobile application* dan *web server*. Selaras dengan pengembangannya, terdapat permasalahan yang muncul yaitu, perangkat lunak MI-FIK membutuhkan teknologi yang mampu menghubungkan antara sisi *mobile application* dan *web server*. Dalam pengembangannya, penggunaan teknologi REST API merupakan jawaban atas permasalahan yang terjadi, yang dapat menghubungkan sisi *mobile application* dan *web server*. Tidak sampai disitu saja, masalah lain muncul yaitu, dibutuhkan metode untuk menguji REST API, agar fungsionalitas perangkat lunak dapat sesuai dengan requirement yang telah disepakati. Penggunaan metode black box testing merupakan solusi atas permasalahan untuk menguji REST API guna memastikan output dari setiap API yang ada sesuai dengan fungsionalitas dari perangkat lunak MI-FIK. Hasil dari pengujian yang dilakukan menggunakan metode black box dapat disimpulkan telah berhasil dan data yang ditampilkan sesuai dengan harapan berdasarkan dengan kasus uji yang telah dilakukan.

Key words: REST API, Black box testing, JSON

Pendahuluan

Pendahuluan

Perangkat lunak MI-FIK adalah proyek lanjutan dari perancangan purwarupa perangkat lunak pengelolaan informasi Fakultas Industri Kreatif, Telkom University yang telah dikelola oleh Ceka Elgy Dwi Putra, mahasiswa program studi S1 Desain Komunikasi Visual angkatan 2018. Perangkat lunak MI-FIK adalah perangkat lunak yang berguna untuk mengelola informasi untuk civitas akademika Fakultas Industri Kreatif yang dirancang untuk menyelesaikan permasalahan yang terjadi pada lingkungan Fakultas Industri Kreatif.

Berdasarkan hasil survei sebelumnya, terdapat beberapa masalah dalam penyebaran informasi di Fakultas Industri Kreatif. Mulai dari penumpukan informasi, informasi yang tidak terklasifikasi, informasi yang dikelola dengan buruk, penggunaan media informasi yang berlebihan, hingga informasi lama yang sulit untuk dilihat kembali. Informasi tersebut meliputi informasi akademik seperti jadwal kuliah, informasi tentang fakultas, informasi kurikulum, jadwal ujian, jadwal kuliah, jadwal mengajar, jadwal pertemuan, event, seminar, dan lainnya.

MI-FIK merupakan perangkat lunak yang dirancang untuk mempermudah dosen, staff, dan mahasiswa Fakultas Industri Kreatif untuk memperoleh informasi di lingkungan Fakultas Industri Kreatif. Perangkat lunak MI-FIK merupakan perangkat lunak yang berbasis mobile application yang saat ini direncanakan akan berjalan menggunakan sistem operasi Android. Pada TA Capstone yang saat ini dijalankan, pengembangan perangkat lunak MI-FIK akan dibagi menjadi 2 sisi pengembangan, yaitu pengembangan dari sisi mobile application dan pengembangan dari sisi API Web Server. Pada bagian pengembangan dari sisi mobile application akan berfokus untuk mengimplementasikan antarmuka pengguna (user interface), sedangkan pada bagian pengembangan dari sisi API Web Server akan berfokus pada pembuatan fungsionalitas dan logika yang bertugas menerima request berupa data informasi dari client dan mengirim response data informasi berdasarkan request tersebut [1]. Sehubungan dalam pengembangannya, terdapat permasalahan yang akan dihadapi karena pengembangan aplikasi dibagi menjadi 2 sisi, sehingga dibutuhkan teknologi yang mampu menghubungkan antara sisi mobile application dan web server. Terdapat juga permasalahan lain, yaitu pengujian yang dilakukan terhadap

teknologi yang digunakan sebagai penghubung antara mobile application dan web server agar data yang bertukar sesuai dengan yang seharusnya.

Dalam pengembangannya, diperlukan teknologi aplikasi yang dapat menjembatani pertukaran data informasi dari web server menuju mobile application ataupun sebaliknya. Untuk itu, penggunaan teknologi REST API yang dikembangkan pada bagian web server diperlukan agar mempermudah dalam melakukan pertukaran data informasi dengan mobile application ataupun sebaliknya. Terdapat empat metode umum HTTP yang ada di dalam sistem REST yaitu POST, GET, DELETE, PUT. Keempat metode HTTP tersebut berperan dalam melakukan proses CRUD dari database antara lain Create, Read, Update, Delete dengan menggunakan JSON sebagai format dalam melakukan pertukaran data [2]. Untuk memastikan bahwa data informasi yang dibutuhkan sesuai dengan request yang diminta oleh client, maka akan dilakukan pengujian terhadap REST API yang dibuat [3].

Untuk memastikan bahwa REST API berjalan dengan semestinya, maka akan dilakukan pengujian terhadap REST API yang dibuat. Pengujian REST API akan menggunakan metode pengujian Black box guna memastikan output dari setiap API yang ada sesuai dengan fungsionalitas dari perangkat lunak MI-FIK [4].

Tinjauan Pustaka

Aplikasi Backend

Aplikasi backend adalah aplikasi yang berfokus pada sisi bagaimana sebuah aplikasi bekerja. Kode pemrograman pada aplikasi backend berfokus pada fungsionalitas dan logika yang diperlukan oleh sebuah aplikasi. Aplikasi backend utamanya berfungsi sebagai penerima request dari client dan mengirim response berdasarkan request tersebut. Aplikasi backend juga merupakan perangkat lunak yang menghubungkan client dengan database sehingga proses pertukaran dan manipulasi data dapat berlangsung [1].

PHP

PHP (Hypertext Preprocessor) merupakan Bahasa pemrograman terstruktur, yang berarti PHP menyediakan cara untuk menyelesaikan masalah komputasi dengan langkah-langkah diskrit yang dilakukan dengan cara "dari atas ke bawah" atau dengan cara terurut [5]. PHP adalah pemrograman berbasis web yang mampu mengolah dan memproses data secara dinamis. PHP merupakan server-side embedded script language, yaitu Bahasa pemrograman yang berjalan di bagian web server yang berarti semua perintah program dijalankan oleh server. Data yang didapat dengan cara client mengirim data, dan data tersebut akan tersimpan di dalam database web server dan dapat ditampilkan apabila client mengakses data tersebut. PHP pada awalnya dirancang untuk terintegrasi dengan web server Apache, namun seiring dengan perkembangannya, PHP juga dapat bekerja dengan web server PWS (Personal Web Server), web server IIS (Internet Information Server), dan web server Xitami [6, 7].

Framework Laravel

Laravel merupakan framework web untuk Bahasa pemrograman PHP yang dapat diunduh secara gratis di website resmi Laravel. Laravel dibuat dan dikelola oleh Taylor Otwell dan bersifat open-source [8], yang berarti siapapun dapat berpartisipasi dalam pengembangan framework Laravel. Laravel mampu membantu developer dalam mengurangi waktu pengembangan aplikasi web dan memuat metode PHP berorientasi objek modern. Sintaks yang ekspresif, mudah dipelajari dan digunakan oleh developer pemula, juga memiliki fungsi modern yang menarik bagi developer [9]. Laravel ditujukan untuk pengembangan aplikasi

web yang mengikuti pola arsitektur model-view-controller (MVC). Laravel menyediakan fitur canggih seperti database abstrak yang ekspresif dan dependency injection yang sangat terukur, sistem modular packaging dengan dependency manager khusus. Laravel menyediakan fitur authentication, routing, session manager, caching, IoC container dan banyak komponen umum yang dapat digunakan, juga tools database migration dan dukungan unit testing terintegrasi, semua alat ini memberi developer kemampuan untuk membangun aplikasi yang kompleks [8, 9, 10].

JSON

JSON (JavaScript Object Notation) merupakan standar format berbasis teks untuk merepresentasikan data terstruktur berdasarkan sintaks objek JavaScript. JSON digunakan untuk mentransfer data yang ada pada aplikasi web untuk mengirim data dari server ke client agar data tersebut dapat ditampilkan pada halaman web atau aplikasi atau sebaliknya. Sintaks format JSON diidentikkan dengan kode yang digunakan untuk membuat objek di JavaScript. Unit data yang direpresentasikan di dalam JSON adalah dokumen [2, 11].

REST API

REST merupakan singkatan dari Representational State Transfer adalah gaya arsitektur perangkat lunak yang sederhana, dapat diskalakan, efektif, aman, dan andal yang menggunakan subset HTTP. Aplikasi web yang terstruktur dengan baik dapat dengan mudah dibangun. Ketika menggunakan arsitektur REST, banyak developer berhasil membuat basis API yang simple dan kuat pada layanan web AJAX dan RESTful. Web-services yang menerapkan prinsip arsitektur REST disebut RESTful. Arsitektur REST pertama kali diperkenalkan pada tahun 2000 oleh Roy Fielding dalam disertasi doktoralnya. REST API berarti sebuah aplikasi API yang menerapkan arsitektur REST. REST API menyediakan metode HTTP seperti POST, GET, PUT, DELETE ke masing-masing endpoint sesuai dengan tugas dari tiap endpoint untuk mengoperasikan CRUD (Create, Read, Update, Delete). Arsitektur REST memiliki enam prinsip, yaitu [1, 8, 9, 12, 13]:

1. **Uniform Interface**
Merupakan penyederhanaan arsitektur dan pemisah antara client dan server, yang memungkinkan masing-masing bagian dikembangkan secara mandiri. Prinsip ini adalah dasar fundamental dari desain arsitektur RESTful. RESTful menggunakan metode HTTP seperti GET, POST, PUT, DELETE untuk mendeskripsikan metode request, URI (Uniform Resource Identifier) berguna untuk mengidentifikasi sumber request, dan HTTP response yang berisi informasi data yang dikembalikan dari server.
2. **Client-Server**
Prinsip Client-Server mendeskripsikan tentang pemisahan dalam pengembangan client dan server, yang memungkinkan masing-masing sisi dapat berkembang secara mandiri. Penerapan prinsip Client-Server, akan meningkatkan probabilitas dan skalabilitas di berbagai platform dengan menyederhanakan komponen server.
3. **Stateless**
Prinsip Stateless mewajibkan request dari client ke server harus berisi informasi yang lengkap dan utuh. Server tidak diperbolehkan menggunakan data atau konteks yang tersimpan pada server. Pada prinsip ini, semua state dan session harus tersimpan di client.
4. **Layered System**
Prinsip Layered System mengharuskan arsitektur yang terdiri dari lapisan hirarki membatasi perilaku setiap komponen.
5. **Code on Demand**
Prinsip Code on Demand yaitu REST memungkinkan server untuk

menyesuaikan fungsionalitas client secara sementara menggunakan transfer logic. Client mampu menjalankan code secara langsung yang dikirim dari server.

6. Cacheable

Prinsip Cacheable merupakan prinsip cache yang mengharuskan data pada response secara implisit atau eksplisit diberi label sebagai informasi bahwa data dapat disimpan dalam cache atau tidak dapat disimpan dalam cache. Jika response dapat disimpan, cache client memiliki hak untuk menggunakan informasi response tersebut untuk request yang sama berikutnya.

Web Server

Web server merupakan sebuah perangkat lunak yang memberikan layanan berupa akses data yang disimpan di repositorinya dan untuk menghasilkan konten dinamis. Web server berfungsi untuk menerima request HTTP atau HTTPS dari client dengan menggunakan internet browser seperti Firefox atau Chrome. Banyaknya client yang terhubung dengan web server, banyak developer dapat menggunakan resource yang sama untuk kebutuhan mereka. Web server dapat diakses menggunakan IP address dan port yang diberikan. Web server dapat meneruskan konten statis dan menghasilkan konten dinamis sesuai dengan permintaan user [14].

Black Box Testing

Black box testing merupakan pengujian yang berfokus pada pengujian fungsional dari perangkat lunak secara spesifik. Tester ditugaskan untuk menentukan sekumpulan kondisi input dan melakukan pengujian terhadap fungsional dari program perangkat lunak tersebut. Proses pengujian black box terdiri dari pengujian perangkat lunak yang dijalankan dengan mencoba memasukkan data pada setiap form yang tersedia. Pengujian black box diperlukan untuk mengetahui apakah program memenuhi persyaratan yang telah ditentukan [4].

Metodologi Penelitian

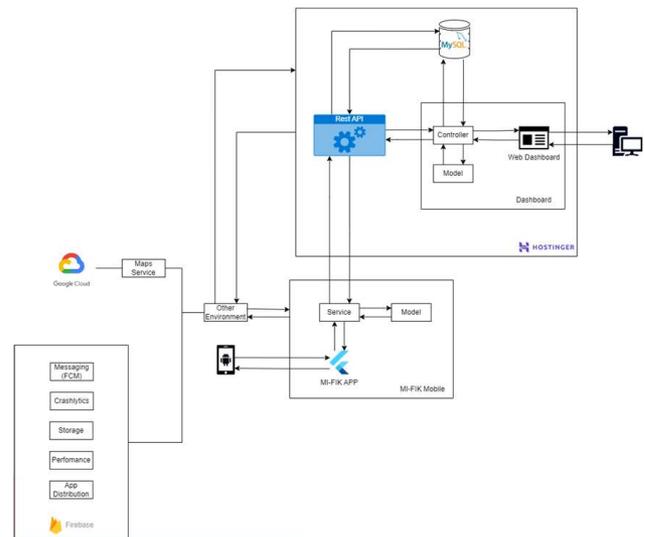
Studi Literatur

Tahap ini akan dilakukan guna memperoleh informasi seperti jurnal, ebook, dan dokumen pendukung lainnya yang dapat dijadikan sebagai acuan dalam pembahasan dan pembuatan tugas akhir.

Analisis Aplikasi

MI-FIK merupakan perangkat lunak yang berguna untuk mengelola informasi untuk civitas akademika Fakultas Industri Kreatif yang dirancang untuk menyelesaikan permasalahan yang terjadi pada lingkungan Fakultas Industri Kreatif. MI-FIK memudahkan dosen, staff, dan mahasiswa Fakultas Industri Kreatif untuk memperoleh informasi di lingkungan Fakultas Industri Kreatif. Perangkat lunak MI-FIK merupakan perangkat lunak yang berbasis mobile application yang saat ini direncanakan akan berjalan menggunakan sistem operasi Android. Proses analisis aplikasi merupakan tahap penting dalam pengembangan aplikasi, terutama pada bagian backend. Proses analisis aplikasi bertujuan agar tercapainya goal yang telah ditentukan. Beberapa langkah yang dilakukan dalam proses analisis aplikasi sebagai backend developer adalah [15]:

1. Memahami kebutuhan aplikasi
Memahami tujuan dan kebutuhan aplikasi atau perangkat lunak yang akan dibangun. Hal ini dapat dilakukan dengan berkomunikasi dengan stakeholder, mengumpulkan spesifikasi proyek dan melakukan riset terkait aplikasi yang sama.
2. Merancang arsitektur sistem
Setelah memahami kebutuhan aplikasi, selanjutnya developer



Gambar 1. Arsitektur Sistem

melakukan perancangan arsitektur sistem yang akan digunakan, termasuk menentukan teknologi yang akan digunakan, menentukan bagaimana komponen aplikasi akan terhubung, menentukan struktur database yang akan digunakan.

3. Membuat spesifikasi teknis
Setelah merancang arsitektur sistem, developer harus membuat spesifikasi teknis yang menjelaskan bagaimana aplikasi akan dibangun secara detail. Spesifikasi teknis mencakup seluruh fitur yang akan dibangun, serta proses dan alur kerja yang akan dilakukan.
4. Menentukan estimasi waktu dan anggaran
Setelah membuat spesifikasi teknis, developer dapat menentukan estimasi waktu yang diperlukan untuk menyelesaikan proyek, serta anggaran yang dibutuhkan.
5. Menyiapkan lingkungan pengembangan
Selanjutnya developer dapat menyiapkan lingkungan pengembangan yang dibutuhkan untuk memulai pengerjaan proyek, termasuk menyiapkan server, menginstall perangkat lunak yang dibutuhkan, menyiapkan kode sumber yang diperlukan.
6. Memulai pengerjaan proyek
Setelah kelima proses selesai, developer dapat memulai pengerjaan proyek sesuai dengan spesifikasi teknis yang telah ditentukan.

Rancangan Sistem

Arsitektur Sistem

Perancangan REST API akan menggunakan framework Laravel. Penggunaan framework Laravel bertujuan untuk memudahkan dalam pengembangan REST API, karena Laravel menyediakan tools untuk pembuatan REST API.

Pada perangkat Lunak MI-FIK, ketika pengguna membuka perangkat lunak MI-FIK, perangkat lunak akan melakukan request data ke web server yang nantinya web server akan memberikan response berupa data yang sebelumnya diminta dan akan ditampilkan pada suatu halaman. Web server dapat mengirimkan request dengan menggunakan method seperti Post, Get, Put, Delete, sesuai request yang dilakukan dari perangkat lunak MI-FIK. Selanjutnya REST API akan melakukan query pada database seperti Create, Read, Update, Delete dan lainnya. Hasil dari query tersebut kemudian akan dikirim kembali ke web server sebagai response dengan menggunakan salah satu method seperti Post, Get, Put, Delete dalam bentuk JSON yang nantinya akan

Table 1. Functional Requirements Mobile

ID	Functional Requirement	Priority
FRM001	Dosen, Staff, dan Mahasiswa dapat melakukan registrasi	High
FRM002	Dosen, Staff, dan Mahasiswa dapat melakukan login	High
FRM003	Dosen, Staff, dan Mahasiswa dapat melakukan logout	Medium
FRM004	Dosen, Staff, dan Mahasiswa dapat melihat dan request tag baru	High
FRM005	Dosen, Staff, dan Mahasiswa dapat mengelola data profile User (Dosen, Staff, dan Mahasiswa)	High
FRM006	Dosen, Staff, dan Mahasiswa dapat melihat informasi event sesuai role yang dimiliki	High
FRM007	Dosen, Staff, dan Mahasiswa dapat melakukan pencarian informasi event	Medium
FRM008	Dosen, Staff, dan Mahasiswa dapat mengelola folder archive untuk menyimpan informasi	High
FRM009	Dosen, Staff, dan Mahasiswa dapat menyimpan dan mengelola event/informasi pada folder archive yang dipilih	High
FRM010	Dosen, Staff, dan Mahasiswa mendapat pengingat(reminder) ketika kegiatan akan berlangsung	High
FRM011	Dosen, Staff, dan Mahasiswa dapat mengelola task pribadi	High
FRM012	Dosen, Staff, dan Mahasiswa dapat melihat daftar event dan task dalam rentang satu bulan	Low
FRM013	Dosen, Staff, dan Mahasiswa dapat melihat kumpulan pengumuman sesuai role dan tag yang dimiliki	High
FRM014	Dosen dan Staff dapat membuat event baru	High
FRM015	Dosen, Staff, dan Mahasiswa dapat melihat riwayat kegiatan yang pernah dilakukan oleh masing-masing pengguna	Medium
FRM016	Dosen, Staff, dan Mahasiswa dapat melihat role yang dimiliki	High
FRM017	Dosen, Staff, dan Mahasiswa dapat melihat dan mengirimkan pertanyaan FAQ	Medium
FRM018	Dosen, Staff, dan Mahasiswa dapat mengirimkan feedback	Medium
FRM019	Dosen, Staff, dan Mahasiswa dapat melihat panduan penggunaan aplikasi	Medium
FRM020	Dosen, Staff, dan Mahasiswa dapat melihat informasi tentang aplikasi	Low
FRM021	Dosen, Staff, dan Mahasiswa dapat melihat agenda task atau event pada suatu hari selama satu minggu ke depan	High

diteruskan menuju perangkat lunak MI-FIK. Sebelum data sampai ke perangkat lunak MI-FIK, controller akan mengarahkan data ke model untuk diolah yang nantinya akan dikirim kembali oleh controller menuju perangkat lunak MI-FIK sebagai data jadi yang siap ditampilkan.

Pada sisi server-side, terdapat session yang menyimpan data-data pengguna ataupun admin yang bersifat pribadi yang nantinya data-data tersebut akan digunakan oleh sistem. Pada sistem, session akan menyimpan data pengguna ataupun admin yang penting seperti username, token, role, email dan data penting lainnya. Sementara pada sisi client-side, session digunakan untuk menyimpan data-data yang baru saja didapatkan dari skema pengambilan data menggunakan REST API.

Functional Requirements dan Non Functional Requirements

Functional Requirements Functional requirements merupakan daftar dari semua fitur, perilaku, dan fungsionalitas yang diharapkan dari sistem yang dibangun. Tujuannya adalah untuk menjelaskan apa yang diharapkan dari sistem tersebut dan bagaimana sistem tersebut dapat berinteraksi dengan pengguna. Functional requirements menjadi panduan dasar dalam perancangan dan pengembangan REST API sesuai dengan kebutuhan sistem. Functional requirements didapatkan dan dipetakan oleh System Analyst ke dalam table dan diberi ID dengan melakukan analisis kasus pengguna, wawancara dan diskusi, analisis kebutuhan bisnis, pengumpulan feedback dari pengguna.

Functional Requirements Mobile Functional requirements mobile berisi fitur-fitur yang dibutuhkan sistem aplikasi mobile MI-FIK. Terdapat 21 functional requirement untuk sistem aplikasi mobile MIFIK yang dipetakan kedalam table dan diberi ID pada tabel 1.

Functional Requirements Web Dashboard Functional requirements web dashboard berisi fitur-fitur yang dibutuhkan sistem aplikasi web

dashboard MIFIK. Terdapat 29 functional requirement untuk sistem aplikasi web dashboard MI-FIK yang dipetakan kedalam table dan diberi ID pada tabel 2

Non Functional Requirements Non functional requirements merupakan aspek yang penting untuk dipertimbangkan saat merancang REST API guna membantu memastikan bahwa API memenuhi kebutuhan dari client dalam hal kinerja, keamanan, keandalan dan faktor penting lainnya. Non functional requirements didapatkan dan dipetakan oleh System Analyst ke dalam bentuk tabel dan menggunakan ID dengan melakukan wawancara dan diskusi dengan pengguna, analisis kebutuhan pengguna.

Non Functional Requirements Mobile Non functional requirements mobile berisi aspek-aspek yang perlu diterapkan oleh sistem aplikasi mobile MI-FIK. Dalam pembuatan REST API, terdapat 7 Non Functional Requirement untuk sistem aplikasi mobile MI-FIK yang dipetakan kedalam table dan diberi ID pada tabel 3.

Non Functional Requirements Web Dashboard Non functional requirements web dashboard berisi aspek-aspek yang perlu diterapkan oleh sistem aplikasi web dashboard MI-FIK. Dalam pembuatan REST API, terdapat 8 Non Functional Requirement untuk sistem aplikasi web dashboard MI-FIK yang dipetakan kedalam table dan diberi ID pada tabel 4.

Konfigurasi Laravel untuk Pembuatan REST API

Sebelum melakukan pembuatan REST API, akan dilakukan beberapa konfigurasi Laravel:

1. Melakukan instalasi xampp dengan versi php 8.1.6

Table 2. Functional Requirements Web Dashboard

ID	Functional Requirement	Priority
FRW001	Dosen dan Staf dapat melakukan registrasi	High
FRW002	Admin, Dosen, dan Staf dapat melakukan login untuk mengakses keseluruhan web dashboard	High
FRW003	Admin, Dosen, dan Staf dapat melakukan logout	Medium
FRW004	Admin, Dosen, dan Staf dapat melihat daftar event yang akan berlangsung, sedang berlangsung, dan sudah selesai	High
FRW005	Admin, Dosen, dan Staf dapat melihat detail event	High
FRW006	Admin, Dosen, dan Staf dapat melakukan pencarian event dan melakukan filter event yang dicari	Medium
FRW007	Admin, Dosen, dan Staf dapat membuat dan mengelola event	High
FRW008	Admin dapat mengelola Tag	High
FRW009	Dosen dan Staf dapat melihat dan melakukan pencarian Tag	Low
FRW010	Admin, Dosen, dan Staf dapat melihat event pada rentang hari dan bulan tertentu	Medium
FRW011	Admin, Dosen, dan Staf dapat melihat event yang telah selesai	Medium
FRW012	Admin dapat membuat dan mengelola data Info, serta dapat mencari dan filter berdasarkan type Info	Medium
FRW013	Admin dapat melihat data dictionary dan mengelola, serta melakukan pencarian data	High
FRW014	Admin dapat melihat pengumuman dan mengelola, serta melakukan pencarian data	High
FRW015	Admin dapat melihat daftar feedback dan destroy feedback, serta melakukan filter berdasarkan Suggestion	Medium
FRW016	Admin dapat melihat dan menjawab pertanyaan	High
FRW017	Admin dapat melihat statistik data	Medium
FRW018	Admin dapat mengatur range data	Low
FRW019	Seluruh pengunjung dapat melihat informasi sekilas tentang MI-FIK, random FAQ dan mengirim Feedback	Medium
FRW020	Admin, Dosen, dan Staff dapat mengelola profile User (Dosen, Staff, dan Mahasiswa)	High
FRW021	Admin dapat mengelola about us, help editor, dan contact us	Medium
FRW022	Dosen dan Staf dapat melihat about us, help editor, dan contact us	Medium
FRW023	Admin dapat melihat login history seluruh user (Dosen, Staff, dan Mahasiswa) dan admin	High
FRW024	Dosen dan Staff dapat melakukan request add dan remove tag User (Dosen, Staff, dan Mahasiswa)	High
FRW025	Admin dapat mengelola Group dan User Group	Medium
FRW026	Admin dapat mengelola role Pengguna User (Dosen, Staff, dan Mahasiswa)	High
FRW027	Admin, Dosen, dan Staf dapat melihat dan mengelola Trash	High
FRW028	Admin, Dosen, dan Staf dapat melihat seluruh lokasi event	Medium
FRW029	Dosen dan Staf dapat melihat dan bertanya FAQ	High

- Melakukan cloning repository project tim dari GitHub dan letakkan project pada folder htdocs yang ada di dalam folder xampp yang telah diinstall sebelumnya
- Melakukan instalasi dependencies yang ada pada project dengan mengetik perintah pada cmd (lokasi direktori project) "composer install"
- Import database pendukung dalam pembuatan REST API
- Melakukan konfigurasi terhadap file .env yang ada dalam folder project menyesuaikan dengan environment yang digunakan
- Melakukan instalasi package Laravel Sanctum sebagai metode authentication
 - Ketikkan pada cmd (lokasi direktori project) "composer require laravel/sanctum"
 - Selanjutnya, mempublikasikan konfigurasi sanctum dan file migration menggunakan perintah artisan vendor:publish.

File konfigurasi sanctum akan ditempatkan pada direktori application config: "php artisan vendor:publish --provider="Laravel/Sanctum/SanctumServiceProvider"

- Selanjutnya jalankan perintah database migration. Sanctum akan membuat satu table database untuk menyimpan token API: "php artisan migrate"
- Pastikan laravel/sanctum sudah include di dalam file composer.json.

Perancangan REST API

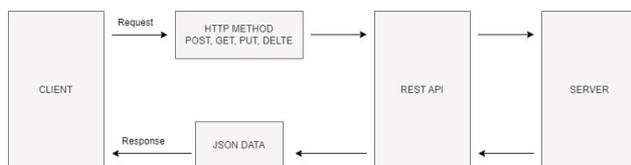
Pada tahap ini, akan dilakukan perancangan desain REST untuk API yang akan dibuat. API yang telah dibuat akan digunakan untuk menghubungkan client dengan REST service dengan mengirim, mengambil, dan merubah data yang terdapat pada database menggunakan konsep request – response yang diakses menggunakan protokol HTTP (HyperText Transfer Protocol) antara client dengan server dengan

Table 3. Non Functional Requirements Mobile

ID	Non Functional Requirement	Quality
NFRG-PR001	Seorang pengguna dapat melakukan login kedalam aplikasi dengan latensi rata-rata 2-5 detik	Performance Requirements
NFRG-SR002	Authentication user (Dosen, Staff, dan Mahasiswa) dengan mengenkripsi password menggunakan hash pada database	Security Requirement
NFRG-SR003	Password harus terdiri minimal 6 karakter yang terdiri dari kombinasi huruf dan angka	Security Requirement
NFRMSR004	User (Dosen, Staff, dan Mahasiswa) dapat menggunakan semua fitur pada aplikasi jika telah terdaftar pada database dan telah melewati authentication (login)	Security Requirement
NFRG-SR005	Audit trail: server akan mencatat aktivitas login pengguna (token, hak akses, dan waktu server)	Security Requirement
NFRM-SF006	Data pribadi tidak dapat dilihat oleh user (Dosen, Staff, dan Mahasiswa) lain, data yang hanya bisa dilihat oleh user (Dosen, Staff, dan Mahasiswa) lain yaitu nama pengguna dan foto profil	Safety Requirement
NFRG-RR013	Aplikasi tersedia selama 24 jam	Reliability

Table 4. Non Functional Requirements Web Dashboard

ID	Non Functional Requirement	Quality
NFRG-PR001	Seorang pengguna dapat melakukan login kedalam aplikasi dengan latensi rata-rata 2-5 detik	Performance Requirements
NFRG-SR002	Authentication user (Dosen, Staff, dan Mahasiswa) dengan mengenkripsi password menggunakan hash pada database	Security Requirement
NFRG-SR003	Password harus terdiri minimal 6 karakter yang terdiri dari kombinasi huruf dan angka	Security Requirement
NFRWSR004	User (Dosen dan Staff) dan Admin dapat menggunakan semua fitur pada aplikasi jika telah terdaftar pada database dan telah melewati authentication (login)	Security Requirement
NFRG-SR005	Audit trail: server akan mencatat aktivitas login pengguna (token, hak akses, dan waktu server)	Security Requirement
NFRWUR007	Pengguna akan diberikan dokumentasi dan menu "Help" untuk membantu mempelajari cara mengoperasikannya	Usability
NFRG-RR013	Web tersedia hingga 24 jam	Reliability
NFRWTR014	Semua fungsionalitas fitur yang dikembangkan akan dicatat tanggal pembuatan, pengujian, hingga perilsan	Traceability

**Gambar 2.** standar format komunikasi JSON (Javascript Object Notation)

menggunakan standar format komunikasi JSON (JavaScript Object Notation).

REST API memiliki method HTTP yang umum digunakan seperti berikut:

1. Method POST digunakan untuk memasukkan data ke dalam database
2. Method GET digunakan untuk mendapatkan atau mengambil data yang terdapat pada database

3. Method PUT digunakan untuk mengubah data yang terdapat pada database
4. Method DELETE digunakan untuk menghapus data yang terdapat pada database

Terdapat klasifikasi pengolahan data yang terdapat pada database berdasarkan path endpoint yang akan di terapkan sebagai berikut:

1. Endpoint dengan path /data akan digunakan untuk mendapatkan atau mengambil data dari database dan memasukkan data ke dalam database.
2. Endpoint dengan path /data/id atau /data/slug_name akan digunakan untuk mendapatkan detail dari data, mengubah data, dan menghapus data berdasarkan id atau slug_name.

Implementasi

Pada tahap ini, pembangunan REST API dengan menggunakan bahasa pemrograman PHP (Hypertext Preprocessor) dengan menggunakan framework Laravel 9 dengan settingan yang telah dirancang.

Pengujian Sistem

Pada tahap ini, pengujian sistem akan dilakukan menggunakan metode Black box testing dengan pengujian yang bersifat Alpha testing yang merupakan pengujian yang dilakukan di lingkungan pengembang, guna memastikan REST API berjalan sesuai dengan persyaratan yang telah disepakati. Black box testing merupakan salah satu metode pengujian sistem yang memfokuskan tes terhadap sistem tanpa memperhatikan bagaimana sistem tersebut dibuat atau bagaimana sistem tersebut bekerja di dalamnya.

Secara umum, pengujian REST API mempunyai tujuan yaitu:

1. Nilai yang dikeluarkan sesuai dengan kondisi yang dimasukkan
2. Memastikan API tidak merubah struktur data yang ada
3. Memastikan API tidak mengeluarkan nilai apapun tanpa kondisi yang telah dimasukkan
4. Memastikan API berhasil mengoutputkan data

Hasil dan Pembahasan

Justifikasi Penggunaan Komponen

Dalam pengembangan REST API, terdapat beberapa komponen utama yang dipilih untuk mencapai tujuan pembuatan REST API. Justifikasi penggunaan masing-masing komponen sebagai berikut:

1. Framework Laravel untuk REST API Development: Laravel merupakan salah satu framework PHP yang populer dan terkemuka dalam pengembangan web, khususnya dalam konteks pengembangan API. Framework Laravel memiliki struktur yang konsisten, termasuk sistem routing yang fleksibel, migrasi database yang terintegrasi, dan model eloquent yang memungkinkan interaksi dengan database menggunakan sintaks yang sederhana.
2. Penggunaan Model Eloquent untuk Interaksi dengan Database: Model Eloquent pada Laravel digunakan untuk mempermudah berinteraksi dengan database. Penggunaan model eloquent akan meminimalkan kompleksitas dalam operasi CRUD (Create, Read, Update, Delete) pada database.
3. Middleware untuk Keamanan dan Validasi: Middleware pada Laravel digunakan untuk menerapkan langkah-langkah keamanan dan validasi sebelum permintaan pengaksesan endpoint tertentu. Penggunaan middleware akan memastikan bahwa data yang diterima oleh API adalah valid sebelum data tersebut diproses lebih lanjut.
4. Metode Pengujian Black Box dengan Postman: Metode pengujian black box digunakan untuk memastikan bahwa pengujian dilakukan dari perspektif pengguna akhir, tanpa memperhatikan implementasi internal dari sistem. Postman digunakan sebagai alat pengujian karena menyediakan antarmuka yang mudah dipahami untuk mengirim permintaan HTTP ke API dan mengevaluasi tanggapan yang diterima.
5. Penggunaan Layanan Hostinger untuk Hosting API: Penggunaan layanan Hostinger sebagai platform untuk hosting API dikhususkan untuk memastikan aksesibilitas dan ketersediaan API kepada pengguna secara online. Layanan Hostinger menawarkan lingkungan hosting yang memiliki skalabilitas dan jaminan ketersediaan yang tinggi.

Perancangan REST API

Hasil dari API yang telah selesai dirancang akan dilakukan pengkategorian ke dalam API ID. Request method merupakan metode HTTP yang digunakan oleh URI, versi API merupakan pemetaan versi yang dilakukan guna mempermudah jika nantinya terjadi penambahan API dengan nama proses yang sama atau hampir sama, nama proses

Table 5. Penerapan 1

Nama	Add content
Proses	
FR ID	FR-M-014
Route	/api/v1/content/create
Deskripsi	Dalam proses pengembangan REST API, proses Add content bertanggung jawab untuk menambahkan konten baru ke dalam sistem melalui endpoint /api/v1/content/create. Proses ini mencakup validasi input, penyimpanan konten ke dalam database, dan pengembalian respons yang sesuai kepada pengguna.

Table 6. Penerapan 2

Nama	Update user image
Proses	
FR ID	FR-M-005
Route	/api/v1/user/update/image
Deskripsi	Dalam proses pengembangan REST API, proses Update user image bertanggung jawab untuk mengubah gambar profil pengguna melalui endpoint /api/v1/user/update/image. Proses ini mencakup validasi input, pembaruan gambar profil di database, dan pengembalian respons yang sesuai kepada pengguna.

adalah nama dari proses yang dapat digunakan dan dijalankan dengan request method dan URI yang telah tersedia, route merupakan pengaturan yang menghubungkan URI dengan tindakan yang harus dijalankan oleh aplikasi, FR ID merupakan functional requirement fitur yang diterapkan pada REST API, pemetaan FR ID ke dalam REST API menunjukkan bahwa REST API dibuat dengan tujuan untuk memenuhi kebutuhan functional requirement yang ada. Terdapat 4 request method yang akan dipakai pada REST API diantaranya GET, POST, PUT, DELETE.

Pengembangan REST API didasarkan pada iterasi yang sudah ditentukan sebelumnya, berikut merupakan beberapa REST API yang diimplementasikan.

Penerapan 1 pada tabel 5. Dalam proses pengembangan REST API Add content, pengembangan berjalan pada iterasi 5 dan iterasi 6, alur berjalannya API mengikuti Sequence Diagram yang telah di buat oleh System Analyst, serta response yang di outputkan menyesuaikan dengan mockup API yang telah di buat System Analyst.

Penerapan 2 pada tabel 6. Dalam proses pengembangan REST API Update user image, pengembangan berjalan pada iterasi 15, alur berjalannya API mengikuti Sequence Diagram yang telah di buat oleh System Analyst, serta response yang di outputkan menyesuaikan dengan mockup API yang telah di buat System Analyst.

Penerapan 3 pada tabel 7. Dalam proses pengembangan REST API Get Content by Slug, pengembangan berjalan pada iterasi 13 dan iterasi 14, alur berjalannya API mengikuti Sequence Diagram yang telah di buat oleh System Analyst, serta response yang di outputkan menyesuaikan dengan mockup API yang telah di buat System Analyst. Berikut sample REST API yang dikembangkan dan dipetakan ke dalam tabel 8.

Table 7. Penerapan 3

Nama Proses	Get Content by Slug
FR ID	FR-M-007
Route	/api/v1/content/slug/{slug}/order/{order}/date/{date}/{utc}/find/{search}
Deskripsi	Dalam proses pengembangan REST API, proses Get Content by Slug bertanggung jawab untuk mendapatkan konten berdasarkan slug tertentu melalui endpoint /api/v1/content/slug/{slug}/order/{order}/date/{date}/{utc}/find/{search}. Proses ini mencakup pencarian konten sesuai dengan parameter slug, order, tanggal, zona waktu, dan pencarian teks tertentu.

Table 8. sample REST API

ID	FR ID	Nama Proses	Request Method	Route
RA-12	FRW007, FRM-014	Add content	POST	V1 /api/v1/content/create
RA-40	FRM-005	Update user image	PUT	V1 /api/v1/user/update/image
RA-06	FRW004, FRW006, FRW007, FRM-007	Get Content by Slug	GET	V1 /api/v1/content/slug/slug/order/order/date/date/utc/find/search

Table 9. Detail Pengujian

Tanggal Pengujian	31 July 2023
Penguji	Rafly Farhandika (Backend API) Alya Ghaitsa Rizky Pertiwi (Quality Assurance)
Aplikasi Pengujian	Postman

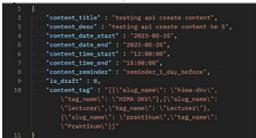
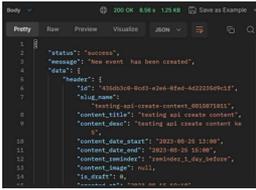
Pengujian Sistem

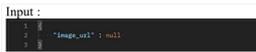
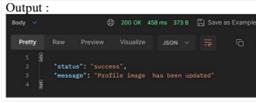
Pengujian REST API

Pengujian REST API dilakukan guna memastikan output data yang dihasilkan oleh REST API sesuai dengan mockup yang telah dibuat oleh System Analyst dan sesuai dengan kebutuhan untuk diolah oleh aplikasi. Pengujian dilakukan menggunakan metode Black Box Testing untuk menguji input dan output yang dihasilkan dari REST API tanpa mengetahui bagaimana kode di dalamnya diimplementasikan. Pengujian REST API dilakukan sebelum kegiatan UAT dan setelah kegiatan UAT. Pengujian dilakukan oleh Backend REST API dan Quality Assurance.

Pengujian Non Functional Requirements

Pengujian Non Functional Requirement dilakukan dengan menguji aspek-aspek non-fungsional yang ditentukan pada saat requirement analysis pada tahap awal pengembangan aplikasi. Pengujian Non functional requirement REST API dilakukan oleh Backend REST API dan Quality Assurance.

REST API ID	RA-12
Route	/api/v1/content/create
Kasus Uji	/api/v1/content/create
Fungsi	Menambahkan content baru / Menginputkan content baru
Hasil yang Diharapkan	Menginput content baru dan mendapatkan response data penginputan data berhasil
Hasil Uji	<p>Input:</p>  <p>Output:</p> 
	Berhasil

REST API ID	RA-40
Route	/api/v1/user/update/image
Kasus Uji	/api/v1/user/update/image
Fungsi	Melakukan update user profile image
Hasil yang Diharapkan	Mendapatkan response success melakukan update user profile image
Hasil Uji	<p>Input:</p>  <p>Output:</p> 
	Berhasil

Kesimpulan
Kesimpulan

Berdasarkan pada pembuatan dan pengujian REST API yang telah dilakukan menggunakan metode black box, dapat disimpulkan bahwa:

1. Perancangan REST API mengacu pada sequence diagram yang telah dibuat oleh System Analyst dan juga struktur response yang dihasilkan ketika REST API dijalankan mengacu pada mockup API yang telah dibuat oleh System Analyst.
2. Komponen yang digunakan dalam pembuatan REST API menggunakan komponen yang mudah dipelajari, memiliki dokumentasi yang lengkap, serta memiliki komunitas yang besar, sehingga memudahkan menemukan solusi jika terkena error saat pembuatan REST API.

REST API ID	RA-06
Route	/api/v1/content/slug/{slug}/order/{order} ³ . date/{date}/{utc}/find/{search}
Kasus Uji	/api/v1/content/slug/all/order/DESC/date/ all/8/find/%20?page=1
Fungsi	Mendapatkan semua data content berdasarkan slug
Hasil yang Diharapkan	Mendapatkan response semua data content berdasarkan slug

Hasil Uji
Output:

```

Output :
Body
  1
  2
  3
  4
  5
  6
  7
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34
  35
  36
  37
  38
  39
  40
  41
  42
  43
  44
  45
  46
  47
  48
  49
  50
  51
  52
  53
  54
  55
  56
  57
  58
  59
  60
  61
  62
  63
  64
  65
  66
  67
  68
  69
  70
  71
  72
  73
  74
  75
  76
  77
  78
  79
  80
  81
  82
  83
  84
  85
  86
  87
  88
  89
  90
  91
  92
  93
  94
  95
  96
  97
  98
  99
  100
  101
  102
  103
  104
  105
  106
  107
  108
  109
  110
  111
  112
  113
  114
  115
  116
  117
  118
  119
  120
  121
  122
  123
  124
  125
  126
  127
  128
  129
  130
  131
  132
  133
  134
  135
  136
  137
  138
  139
  140
  141
  142
  143
  144
  145
  146
  147
  148
  149
  150
  151
  152
  153
  154
  155
  156
  157
  158
  159
  160
  161
  162
  163
  164
  165
  166
  167
  168
  169
  170
  171
  172
  173
  174
  175
  176
  177
  178
  179
  180
  181
  182
  183
  184
  185
  186
  187
  188
  189
  190
  191
  192
  193
  194
  195
  196
  197
  198
  199
  200
  201
  202
  203
  204
  205
  206
  207
  208
  209
  210
  211
  212
  213
  214
  215
  216
  217
  218
  219
  220
  221
  222
  223
  224
  225
  226
  227
  228
  229
  230
  231
  232
  233
  234
  235
  236
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  1000
  
```

Berhasil

3. Pengujian menggunakan metode black box dilakukan dari perspektif pengguna akhir tanpa memperhatikan implementasi internal dari sistem. Metode pengujian ini membantu mengidentifikasi kesalahan fungsionalitas yang dapat dialami oleh pengguna saat berinteraksi dengan API.

Rekomendasi yang dapat diberikan yaitu:

1. Perlu dilakukan pengujian menggunakan white box testing guna memeriksa aspek-aspek yang lebih mendalam seperti efisiensi dan optimasi kerja dari query yang digunakan dalam pembuatan REST API.
2. Pengujian struktur data secara keseluruhan guna mengidentifikasi cacat dan kekurangan yang tidak terlihat pada REST API.
3. Meningkatkan kualitas kode yang digunakan dalam pembuatan REST API.

Daftar Pustaka

1. Rizaldy REP, Ahmad UA, Dirgantoro B. IMPLEMENTASI REST API PADA PENGEMBANGAN APLIKASI BACKEND UNTUK PLATFORM KURSUS ONLINE (GROWUP). e-Proceeding of Engineering. 2022 Oct;9(5):4604-9. Accessed: Nov. 23, 2022. Available from: https://openlibrary.telkomuniversity.ac.id/pustaka/files/181715/jurnal_eproc/implementasi-rest-api-padapengembangan-aplikasi-backend-untuk-platform-kursusonline-growup-.pdf.
2. Herdiyatomoko HF. Desain Sistem Backend Berbasis REST API Menggunakan Framework Laravel 7. SKANIKA: Sistem Komputer

dan Teknik Informatika. 2022 Jul;5(2):136-44.

3. Rindri YA, Hartanto R, Ferdiana R. API Design and Development for Developer Payroll in Startups Based on Agile Productivity Factors. 2019.
4. Iskhak M, Rizkika S. Implementasi metode pengujian equivalence partitioning pada pengembangan RESTful API Sistem Informasi Klinik Pratama UPN 'Veteran' Yogyakarta. 2021:13-2021.
5. Brooks DR. Programming in HTML and PHP. Undergraduate Topics in Computer Science. Springer International Publishing; 2017.
6. Mubarak A. RANCANG BANGUN APLIKASI WEB SEKOLAH MENGGUNAKAN UML (UNIFIED MODELING LANGUAGE) DAN BAHASA PEMROGRAMAN PHP (PHP HYPERTEXT PREPROCESSOR) BERORIENTASI OBJEK. Jurnal Informatika dan Komputer Ternate. 2019 Apr;02(1).
7. Fadila RR, Aprison W, Musril HA. Perancangan Perizinan Santri Menggunakan Bahasa Pemrograman PHP/MySQL Di SMP Nurul Ikhlas. CSRID (Computer Science Research and Its Development Journal). 2021 Mar;11(2):84-95.
8. Chen X, Ji Z, Fan Y, Zhan Y. Restful API Architecture Based on Laravel Framework. 2017 Nov.
9. Subecz Z. Web-development with Laravel framework. Gradus. 2021;8(1):211-8.
10. Faruqi IA, Gumilang SFS, Hasibuan MA. PERANCANGAN BACK-END APLIKASI RUMANTARA DENGAN GAYA ARSITEKTUR REST MENGGUNAKAN METODE ITERATIVE INCREMENTAL. e-Proceeding of Engineering. 2018 Mar;5(1):1411-7.
11. Frozza AA, dos Santos Mello R, de Souza da Costa F. An approach for schema extraction of json and extended JSON document collections. In: Proceedings - 2018 IEEE 19th International Conference on Information Reuse and Integration for Data Science, IRI 2018. Institute of Electrical and Electronics Engineers Inc.; 2018. p. 356-63.
12. Gede I, Ariantara M, Arwani I, Hayuhardhika W, Putra N. Penerapan REST API dalam Pengembangan Aplikasi Pemesanan Rental Mobil berbasis Web dan Mobile (Studi Kasus: CV. Dwi Cipta Rent Car). 2020. Available from: <http://j-ptiik.ub.ac.id>.
13. Adam BM, Besari ARA, Bachtiar MM. Backend Server System Design Based on REST API for Cashless Payment System on Retail Community. In: International Electronics Symposium (IES); 2019. .
14. Horvat Z, Ilić V, Nikolić M. Web Server and QR Decoder Applications for Xilinx FPGA Boards. In: 2018 Zooming Innovation in Consumer Technologies Conference (ZINC); 2018. .
15. Subhiyaktto ER, Utomo DW. ANALISIS DAN PERANCANGAN APLIKASI PEMODELAN KEBUTUHAN PERANGKAT LUNAK MENGGUNAKAN METODE PROTOTYPING. In: PROSIDING SEMINAR NASIONAL MULTI DISIPLIN ILMU & CALL FOR PAPER UNISBANK KE3(SENDI.U3) 2017; 2017.

Table 13. Pengujian Non Functional Requirements 1

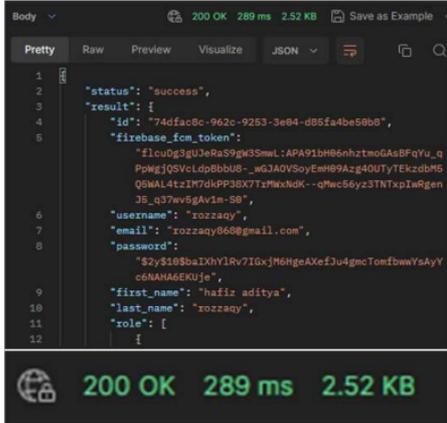
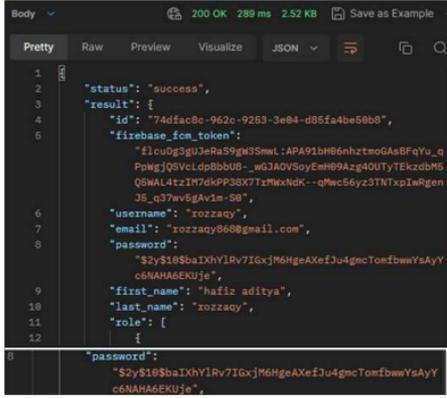
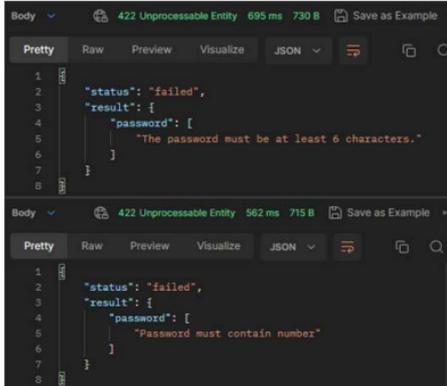
ID NFR	Kasus Uji	Hasil NFR
NFR-GPR001	Melakukan hit API login menggunakan akun dengan username rozzaqy, setelah itu akan terlihat waktu yang dibutuhkan api untuk menemukan data dengan username rozzaqy dan password yang diinputkan	 <pre> 1 { 2 "status": "success", 3 "result": { 4 "id": "74dfac0c-962c-9253-3e04-d85fa4be59b0", 5 "firebase_fcm_token": 6 "f1cu0g3Uj3eRa59gn35mml:APA91bH6hztwoGAsBFqYu_gPmgjQ5VclDpBbbU8-_wGJAOVSoyEmH9A2g40UtyTEkzdbM5Q5wAL4tzIN7dkPP3BX7z1MwxNdk--qMwc56yz3TNTxpIaRgen35_q37wv6gAv1m-S8", 7 "username": "rozzaqy", 8 "email": "rozzaqy868@gmail.com", 9 "password": 10 "\$2y\$10\$b8aIXhY1Rv7IGxjM6HgeAXef3u4gncTomfbwYsAyYc6NAHA6EKUje", 11 "first_name": "hafiz aditya", 12 "last_name": "rozzaqy", 13 "role": [14 {} 15] 16 } 17 } </pre> <p>200 OK 289 ms 2.52 KB</p>
NFR-GSR002	Melakukan hit API login menggunakan akun dengan username rozzaqy, setelah itu akan mengouputkan data dari akun rozzaqy, password akun rozzaqy dalam keadaan terenkripsi menggunakan hash	 <pre> 1 { 2 "status": "success", 3 "result": { 4 "id": "74dfac0c-962c-9253-3e04-d85fa4be59b0", 5 "firebase_fcm_token": 6 "f1cu0g3Uj3eRa59gn35mml:APA91bH6hztwoGAsBFqYu_gPmgjQ5VclDpBbbU8-_wGJAOVSoyEmH9A2g40UtyTEkzdbM5Q5wAL4tzIN7dkPP3BX7z1MwxNdk--qMwc56yz3TNTxpIaRgen35_q37wv6gAv1m-S8", 7 "username": "rozzaqy", 8 "email": "rozzaqy868@gmail.com", 9 "password": 10 "\$2y\$10\$b8aIXhY1Rv7IGxjM6HgeAXef3u4gncTomfbwYsAyYc6NAHA6EKUje", 11 "first_name": "hafiz aditya", 12 "last_name": "rozzaqy", 13 "role": [14 {} 15] 16 } 17 } </pre>
NFR-GSR003	Melakukan hit API register, password yang diinputkan kurang dari 6 dan password yang diinputkan lebih dari sama dengan 6 namun tidak memasukkan angka	 <pre> 1 { 2 "status": "failed", 3 "result": { 4 "password": [5 "The password must be at least 6 characters." 6] 7 } 8 } </pre> <pre> 1 { 2 "status": "failed", 3 "result": { 4 "password": [5 "Password must contain number" 6] 7 } 8 } </pre>

Table 14. Pengujian Non Functional Requirements 2

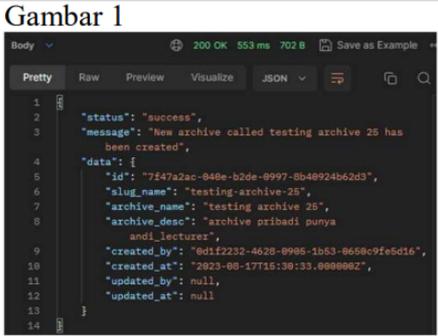
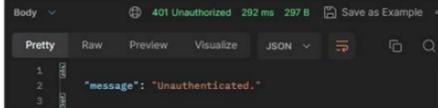
ID NFR	Kasus Uji	Hasil NFR								
MSR004	<p>NFR- Sebelum melakukan hit API add archive, user diharuskan melakukan login terlebih dahulu. Gambar 1 menunjukkan user berhasil membuat archive baru bernama archive 25. Gambar 2 menunjukkan ketika melakukan hit API add archive namun tidak dalam kondisi login, maka tidak dapat membuat archive dan akan mengeluarkan pesan "Unauthenticated"</p>	<p>Gambar 1</p>  <pre> 200 OK 553 ms 702 B {"status": "success", "message": "New archive called testing archive 25 has been created", "data": { "id": "7f47a2ac-048e-b2de-0997-8b40924b62d3", "slug_name": "testing-archive-25", "archive_name": "testing archive 25", "archive_desc": "archive pribadi punya andi_lecturer", "created_by": "6d12232-4628-0985-1b53-0650c9fe5d16", "created_at": "2023-08-17T15:30:33.000000Z", "updated_by": null, "updated_at": null }} </pre> <p>Gambar 2</p>  <pre> 401 Unauthorized 292 ms 297 B {"message": "Unauthenticated."} </pre>								
GSR005	<p>NFR- Setelah user melakukan login maka sistem akan mencatat waktu kapan terakhir user melakukan login pada database</p>	<table border="1"> <thead> <tr> <th>last_used_at</th> <th>expires_at</th> <th>created_at</th> <th>updated_at</th> </tr> </thead> <tbody> <tr> <td>2023-08-17 15:30:33</td> <td>NULL</td> <td>2023-08-17 15:29:39</td> <td>2023-08-17 15:30:33</td> </tr> </tbody> </table>	last_used_at	expires_at	created_at	updated_at	2023-08-17 15:30:33	NULL	2023-08-17 15:29:39	2023-08-17 15:30:33
last_used_at	expires_at	created_at	updated_at							
2023-08-17 15:30:33	NULL	2023-08-17 15:29:39	2023-08-17 15:30:33							
MSF006	<p>NFR- Pada aplikasi MIFIK Mobile user hanya dapat melihat foto profil dan nama pengguna</p>									

Table 15. Pengujian Non Functional Requirements 3

ID NFR	Kasus Uji	Hasil NFR
--------	-----------	-----------

NFR- GRR013 Gambar disamping merupakan monitoring dari REST API yang telah di deploy pada server hostinger, gambar tersebut menunjukkan bahwa REST API dalam keadaan online 24 Jam



Gambar 1

```

1  {
2    "status": "success",
3    "message": "New event has been created",
4    "data": {
5      "header": {
6        "id": "695a24da-987d-69f9-13bd-739c68f4d2c2",
7        "slug_name": "presentasi-tugas-akhir-kelompok-mi-fik-0017041728",
8        "content_title": "Presentasi tugas akhir kelompok MI-FIK",
9        "content_desc": "Presentasi sidang tugas akhir kelompok MI-FIK diadakan jam 8:30 sampai jam 11:30 di gedung TULT ruang 6602",
10       "content_date_start": "2023-08-17 08:30",
11       "content_date_end": "2023-08-17 12:00",
12       "content_reminder": "reminder_1_hour_before",
13       "content_image": null,

```

Gambar 2

```

1  {
2    "message": "Unauthenticated."
3  }

```

NFR- WUR004 Sebelum melakukan hit API add content, user atau admin diharuskan melakukan login terlebih dahulu. Gambar 1 menunjukkan user atau admin berhasil membuat content baru. Gambar 2 menunjukkan ketika melakukan hit API add content namun tidak dalam kondisi login, maka tidak dapat membuat archive dan akan mengeluarkan pesan "Unauthenticated"



NFR- WUR007 Jika user (Dosen atau Staff) dan admin ingin mengetahui cara pembuatan event, maka dapat navigasi ke bagian About, akan ada Help Center yang berisi cara melakukan pembuatan event

NFR- WTR014 Semua fungsionalitas fitur yang dikembangkan akan dicatat tanggal pembuatan, pengujian, hingga perilsan

Terdapat di dokumen SRS