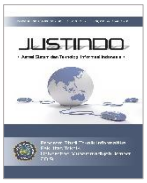
JUSTINDO
Jurnal Sistem dan Teknologi Informasi Indonesia
p-ISSN 2502-5724
e-ISSN 2541-5735

Jurnal Sistem dan Teknologi Informasi Indonesia
JUSTINDO Vol. 10 No. 2, Agustus 2025, hal. 146-154
http://ejurnal.unmuhjember.ac.id/index.php/JUSTINDO/index
https://doi.org/10.32528/justindo.v10i2.3703



Implementasi Web Service Restful API Pada Modul Wisata Aplikasi Malldesa

Implementation of Restful API Web Service in the Tourism Module of the Malldesa Application

Lutfi Ali Muharom*[1], Arindha Dyah Permata[2], Hardian Oktavianto[3]

[1,2,3] Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
Email: [1]lutfi.muharom@unmuhjember.ac.id, [2]arindhadyah16@gmail.com, [3]hardian@unmuhjember.ac.id
[*]Penulis Koresponden

Received: 05 Juli 2025 | Accepted: 03 Juli 2025 | Published: 26 Agustus 2025

This work is licensed under a Creative Commons Attribution 4.0 International License.
Copyright (c) 2025 JUSTINDO

ABSTRAK

Di era industri teknologi yang semakin berkembang secara pesat, penggunaan internet menjadi sangat popular. Perkembangan internet tersebut disertai dengan banyaknya aplikasi yang dibangun seperti aplikasi berbasis mobile dan website.Teknologi yang memungkinkan pengembang untuk membangun aplikasi yang lebih kompleks dengan efisien dan fleksibel yaitu teknologi *web service*. *Web service* adalah teknologi pada suatu sistem yang dapat berkomunikasi melalui internet dengan menggunakan protokol seperti HTTP. Web service berkembang menjadi beberapa macam salah satunya yaitu RESTful API yang akan digunakan pada penelitian ini. RESTful API merupakan gaya arsitektur web service yang memanfaatkan protokol HTTP dan format data seperti JSON atau XML. JSON merupakan standar pertukaran data pada web yang mendukung struktur data seperti object dan arrays sehingga sangat mudah menulis dan membaca data dari JSON.Keuntungan dari penggunaan RESTful API adalah lebih mudah dalam mengembangkan dan memelihara aplikasi, karena RESTful API mengikuti standar protokol HTTP yang sudah sering digunakan dan mudah dimengerti. Pengujian API dapat dilakukan secara manual atau dapat diotomatisasi dengan alat pengujian API. Alat pengujian API yang sering digunakan adalah Postman. Metode HTTP yang digunakan pada penelitian ini adalah GET, POST, PUT dan DELETE.

Kata kunci: JSON, postman, restful API, web service

ABSTRACT

In the era of rapidly developing technology industry, the use of the internet has become very popular. The development of the internet is accompanied by many applications that are built such as mobile and web based applications. Technology that allows developers to build more complex applications efficiently and flexibly is web service technology. Web service is a technology in a system that can communicate via the internet using protocols such as HTTP. Web services have developed into several types, one of which is the RESTful API which will be used in this study. RESTful API is a web service architectural style that utilizes the HTTP protocol and data formats such as JSON or XML. JSON is a data exchange standard on the web that supports data structures such as objects and arrays so that it is very easy to write and read data from JSON. The advantage of using RESTful API is that it is easier to develop and maintain applications, because RESTful API follows the HTTP protocol standard that is often used and easy to understand. API testing can be done manually or can be automated with API testing tools. The API testing tool that is often used is Postman. The HTTP methods used in this study are GET, POST, PUT and DELETE.

Keywords: *JSON, postman, restful API, web service*

1. Pendahuluan

Web service adalah teknologi pada suatu sistem yang dapat berkomunikasi melalui internet dengan menggunakan protokol seperti HTTP (Chamim, Hardyanto, & Putra, 2021). *Web service* memungkinkan aplikasi atau sistem yang berbeda untuk dapat memanfaatkan sumberdaya dan bertukar data melalui jaringan dengan mudah tanpa harus mengetahui atau memahami secara detail cara kerja sistem atau aplikasi tersebut (Hou, Zhao, & Başar, 2021). Penggunaan *web service* semakin luas dengan berkembangnya teknologi *cloud computing* dan *Internet of Things* (IoT), dimana berbagai aplikasi terhubung melalui jaringan internet dan memerlukan pertukaran data yang efisien (Falah, et al., 2021). *Web service* berkembang menjadi beberapa macam salah satunya yaitu RESTful API yang akan digunakan pada penelitian ini. RESTful API merupakan gaya arsitektur web service yang memanfaatkan protokol HTTP dan format data seperti JSON atau XML (Zhang & Arcuri, 2023). Teknologi ini dapat diterapkan untuk membangun sebuah sistem informasi berbasis web maupun *mobile* seperti aplikasi MallDesa.

Malldesa dijalankan pada aplikasi *mobile* yang menggunakan arsitektur monolitik, yaitu *framework react native* di bagian *front-end* dan *framework laravel* di bagian *back-end*. Permasalahan yang timbul jika menggunakan arsitektur ini adalah, perubahan di sisi *back-end* cenderung dapat memiliki dampak kerusakan pada sisi *front-end*, sehingga tim *front-end* dan *back-end* tidak bisa bekerja paralel dengan baik, selain itu sistem ini hanya cocok untuk *react native* sehingga menimbulkan kesulitan jika ada integrasi modul atau dengan aplikasi lain.

Dalam penelitian ini dilakukan pengembangan modul wisata pada aplikasi Malldesa digunakan RESTful API untuk mempermudah interaksi antara aplikasi dengan sistem lain. Keuntungan dari penggunaan RESTful API adalah lebih mudah dalam mengembangkan dan memelihara aplikasi, karena RESTful API mengikuti standar protokol HTTP yang sudah sering digunakan dan mudah dimengerti (Ehsan, Abuhaliqa, Catal, & Mishra, 2022). Selain itu, RESTful API juga lebih fleksibel karena dapat dikembangkan pada berbagai platform dan dapat diintegrasikan dengan sistem lain dengan mudah (Amras, Yulianto, Dewi, & Setiawan, 2023).

Menurut survey yang dilakukan pada penelitian sebelumnya, rendahnya kualitas dari aplikasi *mobile* disebabkan karena *developer* tidak menerapkan *development life cycle* dalam mengembangkan aplikasi *mobile* (Almaiah, *et al*., 2022). Berdasarkan pada penelitian tersebut peneliti akan membuat aplikasi *mobile* pada *platform* android dengan menerapkan fase – fase SDLC dengan metode *waterfall* (Yas, ALazzawi, & Rahmatullah, 2023).

Dengan adanya implementasi *web service* RESTful API pada aplikasi Malldesa, pengguna aplikasi akan lebih mudah dalam melakukan akses layanan dan informasi dengan efektif dan efisien (Ma, Hsu, Chen, & Lin, 2023). Selain itu, dengan memperhatikan fase – fase dalam mengembangkan sebuah sistem dengan metode *waterfall* maka API yang dibuat mempunyai kualitas yang baik.

2. Metode Penelitian

Tahapan penelitian implementasi web service RESTful API pada modul aplikasi Malldesa berbasis mobile menggunakan model pengembangan waterfall, dimulai dengan analisis masalah, studi literatur, analisis solusi, pembuatan perangkat lunak, analisis hasil pengujian, dan penarikan kesimpulan.

Gambar 1. Alur Penelitian

2.1. Analisis Permasalahan

Analisis permasalahan didapatkan berdasarkan kebutuhan yang muncul di lapangan mengenai sistem informasi terpadu yang dapat menfasilitasi kebutuhan akses informasi secara tepat dan efisien.

2.2. Studi Literatur

Studi literatur dilakukan untuk mendapatkan pengetahuan yang bersumber dari kajian pustaka dan dasar-dasar teori yang berkaitan dengan hasil dari analisa permasalahan pada tahapan yang dilakukan sebelumnya pada proses penelitian. Kajian pustaka serta dasar teori dapat bersumber dari jurnal penelitian yang sudah dilakukan sebelumnya, internet, dan buku. Selain kajian pustaka juga digunakan dasar teori. Dasar teori yang digunakan adalah penjelasan singkat mengenai web service dan RESTful API sebagai teknologi yang digunakan. Penjelasan tentang JSON sebagai format file text untuk menampilkan struktur dara berbasis Javascript yang biasa digunakan pada proses pertukaran data antara client dan server (He, Ying, Qian, Feng, & Zhang, 2020).

2.3. Analisis Solusi

Analisis solusi merupakan kegiatan yang dilakukan untuk mendapatkan sebuah solusi. Analisis solusi ini nantinya akan memberikan output tawaran solusi untuk mengatasi permasalahan yang ada. Dimana setelah melakukan analisis permasalahan peneliti mampu menemukan masalah yang kemudian dapat digunakan sebagai acuan dalam pembuatan alternatif solusi. Dengan mempertimbangkan permasalahan pada analisis permasalahan yang dilakukan sebelumnya, peneliti mencari solusi yang dapat merubah pola yang kurang efisien ke dalam pola yang lebih mudah dan sesuai kebutuhan.

2.4. Pembuatan Perangkat Lunak

Dalam menerapkan web service Restful API kedalam aplikasi yang sudah ada dibutuhkan model pengembangan agar proses implementasi dapat dilakukan secara terstruktur. Pada penelitian ini model pengembangan yang dipilih yaitu model waterfall, dipilih karena dengan menggunakan model waterfall akan lebih mudah mengembangkan aplikasi sesuai dengan kebutuhan yang dikumpulkan pada awal proses (Jardim, Santos, Neto, da Silva, & de Barros, 2020).

2.4.1 Analisis Kebutuhan

Tahap analisa kebutuhan perangkat lunak pada model waterfall menjelaskan mengenai analisa spesifikasi perangkat lunak.Kegiatan yang akan dilakukan oleh peneliti adalah mengumpulkan kebutuhan secara keseluruhan dari awal sampai selesai. Kemudian setelah kebutuhan terkumpul maka setelah itu dianalisis kebutuhan apasaja yang harus dipenuhi untuk implementasi teknologi web service RESTful API. Hasil akhir dari fase ini adalah identifikasi actor yang terlibat dalam sistem dan spesifikasi kebutuhan dari perangkat lunak yang kemudian dapat ditampilkan dengan diagram usecase serta dijelaskan dengan usecase skenario.

2.4.2 Perancangan Perangkat Lunak

Tahap perancangan pada model waterfall dilakukan apabila semua kebutuhan untuk implementasi web service RESTful API pada aplikasi MallDesa terpenuhi pada tahapan analisis kebutuhan. Pada tahap ini dijelaskan bagaimana implementasi arsitektur REST untuk saling berkomunikasi antar sistem pada jaringan internet. Hasil dari tahap perancangan kemudian akan digunakan dan diimplementasi pada tahap selanjutnya yaitu tahap implementasi.

2.4.3 Implementasi

Hasil rancangan pada tahap perancangan perangkat lunak kemudian akan diimplementasikan menjadi sebuah sistem pada fase implementasi. Untuk implementasi web service RESTful API diperlukan beberapa komponen aplikasi.Implementasi perangkat lunak ini dapat dikelompokkan menjadi dua bagian yaitu server side dan client side. Pada server side implementasi dilakukan dengan menggunakan framework Laravel. Pada pertukaran data menggunakan JSON untuk menerjemahkan data agar dapat di tampilkan pada aplikasi android. Hasil akhir dari fase ini adalah sistem yang sesuai dengan fase perancangan.

2.4.4 Pengujian Sistem

Tahap pengujian sistem dilakukan dengan menguji hasil dari implementasi perangkat lunak. Hasil dari pengujian adalah sesuai atau tidaknya implementasi sistem terhadap kebutuhan yang sudah didefinisikan pada awal fase pengembangan. Pengujian yang dilakukan yaitu pengujian White-box dengan menggunakan menggunakan Postman.

2.5. Analisis Hasil Penelitian

Analisa hasil penelitian dilakukan ketika semua tahapan pada pengembangan perangkat lunak sudah selesai dilakukan. Analisa dilakukan ketika tahapan pengujian dari implementasi *web service* RESTful API dengan menggunakan model pengembangan *waterfall* telah selesai dilakukan.

2.6. Penarikan Kesimpulan

Tahap terakhir dalam penelitian ini adalah pengambilan kesimpulan dari aplikasi yang telah didokumentasikan dan disempurnakan. Pengambilan kesimpulan didasarkan pada kesesuaian antara perancangan dengan hasil pengujian aplikasi.

3. Hasil dan Pembahasan

Bagian ini membahas mengenai hasil dari implementasi RESTful API pada aplikasi Malldesa di desa Sidomulyo yang berfokus pada sisi *back-end* menggunakan *framework* Laravel. Hal yang akan dibahas pada bagian ini meliputi lingkungan implementasi, implementasi kode program, serta pembahasan dari hasil pengujian yang dilakukan.

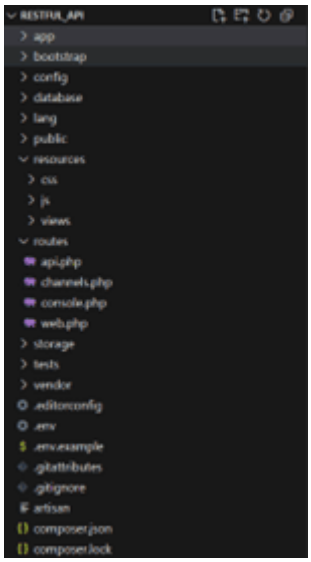
3.1. Arsitektur Sistem

Malldesa merupakan aplikasi *client-server* yang memiliki berbagai informasi terkait tentang desa Sidomulyo kab. Jember. Pada penelitian ini hanya akan berfokus pada pengembangan *back-end* yaitu penerapan Restful API dengan menggunakan *framework* laravel. Pada sisi *client* hanya dapat melakukan proses *create*, *read*, *delete*, *update* data yang akan digunakan pada penelitian ini. Gambaran proses pertukaran data antara *client server* menggunakan Restful API sebagai berikut :

1). Client mengirim permintaan HTTP ke server menggunakan metode GET, POST, DELETE, PUT.
2). Server menerima permintaan dari client dan kemudian memproses permintaan sesuai dengan salah satu mentode yang digunakan.
3). Server kemudian merespon permintaan client dengan mengembalikan permintaan dalam format JSON
4). Client menerima respon dari server berupa data yang diinginkan.

3.2. Implementasi Kode

Pada pengembangan Restful API menggunakan framework laravel, hal pertama yang dilakukan yaitu menginstall laravel dan kemudian akan didapatkan struktur komponen seperti Gambar 2.



Gambar 2. Struktur Komponen Laravel

Laravel menggunakan pola desain *Model-View-Controller* atau dikenal dengan MVC. Ketika meng-*install* laravel maka komponen tersebut akan secara otomatis ter-*install*. Selain ketiga komponen tersebut dalam membuat API dibutuhkan sebuah *resource*. API *resource* adalah sebuah layer yang

mengubah data yang diperoleh dari Model ke dalam format JSON dengan cepat. Penggunaan *resource* memudahkan dalam mengubah *response*. Dalam penelitian ini hanya membuat satu *resource* yaitu WisataResource.php yang berada pada *folder* app\Http\Resource.$resource adalah data yang akan ditransformasikan berupa data model yang dikirim dari *controller*. Kemudian dibuat 3 komponen utama dalam pembuatan Restful API dengan menggunakan laravel. Model yang dibuat dengan nama Wisata.php dan *controller* dengan nama WisataController.php. Selanjutnya didalam WisataController.php terdapat method store() yang berfungsi untuk menambahkan data ke dalam *database*. Detail dari validasi data disajikan pada Tabel 1.

Apabila data yang dikirim tidak sesuai dengan validasi yang ditentukan maka akan dilakukan *return* ke dalam format JSON dengan status *code* 422 dan memberi pesan validasi *error*. Tetapi apabila data yang dikirim sudah sesuai maka akan dilanjutkan eksekusi baris program selanjutnya yaitu menggunggah gambar ke dalam *server*. Setelah semua proses dieksekusi, selanjutnya data akan di tambahkan ke dalam *database* menggunakan *Eloquent*.

Tabel 1. Detail validasi pada method store()

Key	Validasi	Keterangan
nama	required	field harus diisi
info	required	field harus diisi
contact	required	field harus diisi
image	required	field harus diisi
	image	field harus berupa gambar
	mimes:jpeg,png,jpg,gif,svg	field harus memiliki extensi jpeg, png, jpg, gif, svg
lokasi	required	field harus diisi

Method selanjutnya yang ada pada controller WisataController.php adalah method show(). Method show() berfungsi untuk mengambil data dan menampilkan data sesuai dengan id request. Method ini memiliki parameter $id untuk mencari mengambil data berdasarkan id. Apabila proses berjalan dengan baik maka akan mengembalikan data sesuai id. Apabila id yang dicari tidak ada di dalam database maka akan ditampilkan status false dan pesan ‘Data wisata tidak ditemukan’ tetapi tidak memunculkan pesan error karena terdapat kondisi dimana apabila data tidak ditemukan maka yang ditampilkan adalah ‘data wisata tidak ditemukan’.

Berikutnya yaitu method update() digunakan untuk mengupdate data yang ada pada database. Method ini memiliki parameter berupa id yang berfungsi untuk mengambil data sesuai dengan ID yang diminta.

Hampir sama dengan source code yang ada pada method store() hanya yang membedakan pada method update() perlu adanya pengambilan data terlebih dahulu sesuai dengan id yang digunakan pada parameter untuk kemudian diubah dan disimpan kembali. Didalamnya juga terdapat proses validasi untuk memastikan data yang dikirim oleh request telah sesuai apabila tidak sesuai maka akan dikembalikan menggunakan JSON dengan status kode 422 yang berarti error validasi.

Untuk method yang terakhir adalah method destroy() yang berfungsi untuk menghapus data memiliki parameter id sehingga data yang dihapus adalah data yang sesuai dengan id yang diminta.Method ini didefinisikan di dalam Controller dan menggunakan eloquent ORM untuk menghapus data di dalam database.

3.3. Pembuatan Route API

Membuat *route* RESTful API berarti mendesain dan mengimplementasikan *endpoint* pada aplikasi untuk menangani permintaan HTTP berdasarkan prinsip REST (*Representational State Transfer*). Restful API menggunakan metode HTTP standart untuk operasi sumber daya. Pada penelitian ini, *route* yang dibuat adalah sebagai berikut:

1) GET /api/wisata : untuk mengambil item atau data dari database
2) POST /api/wisata : untuk menambahkan data baru kedalam database
3) GET /api/wisata/{id} : untuk mengambil data sesuai dengan id request
4) PUT /api/wisata/{id} : untuk mengupdate data sesuai dengan id request
5) DELETE api/wisata/{id} : untuk menghapus data di database sesuai request id

3.4. Struktur URL

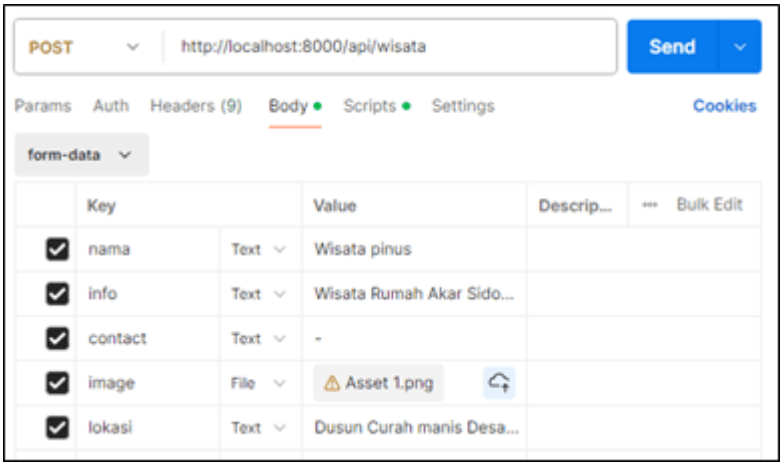
Pada penelitian ini URL yang digunakan untuk menguji API pada alat uji adalah sebagai berikut :

1). http://localhost:8000/api/wisata menggunakan method GET
2). http://localhost:8000/api/wisata menggunakan method POST
3). http://localhost:8000/api/wisata/{id} menggunakan method GET dengan parameter id
4). http://localhost:8000/api/wisata/{id} menggunakan method POST yang kemudian di override dengan method PUT
5). http://localhost:8000/api/wisata/{id} menggunakan method DELETE

3.5. Pengujian Postman

Pengujian *postman* dilakukan untuk menguji API yang sudah dibuat pada proses sebelumnya dengan menggunakan *framework* laravel. Pada *postman* dapat mengirim permintaan HTTP ke API server dan menguji respons yang diterima. *Postman* memastikan API yang dibuat telah memenuhi persyaratan fungsional, mengidentifikasi masalah sebelum diterapkan dan memvalidasi integritas data. Terdapat beberapa *method* HTTP untuk melakukan *request* yang digunakan pada penelitian ini yaitu GET, POST, PUT dan DELETE. Setelah itu permintaan dikirim, *respond* API dianalisis. *Respond* bisa berupa data dalam format JSON yang kemudian diperiksa untuk memastikan bahwa data yang diterima sesuai dengan yang diharapkan.

Permintaan POST berbeda dengan permintaan GET karena diperlukan adanya manipulasi data pada bagian *body* untuk menguji kesesuaian input data dan juga untuk menguji *server* dapat menerima permintaan HTTP POST dengan benar. Akses URL yang digunakan adalah http:/localhost:8000/api/wisata.



Gambar 3. Uji endpoint POST

Key dan *type* yang perlu di masukkan kedalam form data pada body adalah sebagai berikut seperti pada Tabel 2.

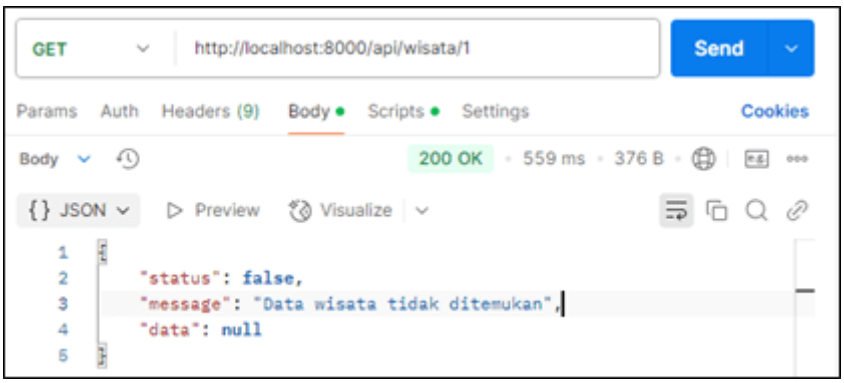
Tabel 2. Key dan type untuk endpoint POST

Key	*Type*	*Value*
nama	text	masukkan nama tempat
info	text	masukkan info wisata
contact	text	masukkan contact wisata
image	file	pilih gambar dari komputer
lokasi	text	masukkan lokasi tempat

Setelah mengisi form data selanjutnya pilih Send. Kemudian data akan di kirim menggunakan endpoint yang sudah ditetapkan. Dari hasil permintaan diatas dikembalikan nilai seperti pada gambar. Kode yang diberikan adalah 201 yang artinya bahwa permintaan berhasil dilakukan dan sumber daya baru sudah ditambahkan.

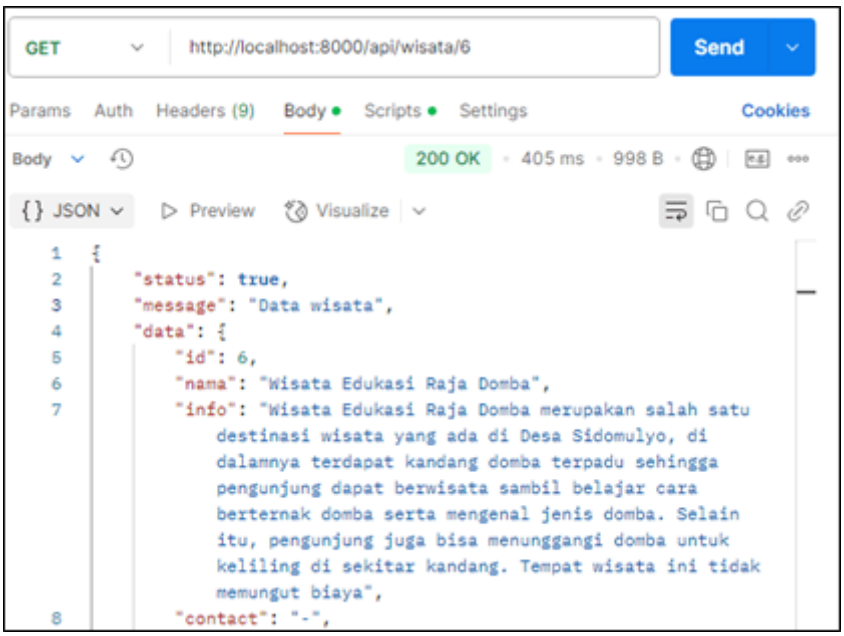
Pengujian endpoint GET by id bertujuan untuk mengetahui bahwa endpoint dapat mengambil data berdasarakan dengan id yang diberikan. Cara yang dilakukan adalah dengan membuat permintaan GET lalu menambahkan id yang akan diuji ke URL. URL yang digunakan adalah

http:/localhost:8000/api/wisata. Id yang digunakan untuk menguji adalah 1 dan 6. Berikut hasil dari pengujian endpoint GET dengan id = 1.



Gambar 4. Uji endpoint GET by id

Hasil yang diberikan setelah permintaan dikirim dengan id = 1 adalah permintaan dapat dipenuhi tetapi status yang dimunculkan adalah false karena tidak ada data dengan id = 1 di dalam tabel. Selanjutnya akan diuji dengan menggunakan id = 6 dengan hasil seperti gambar 5.

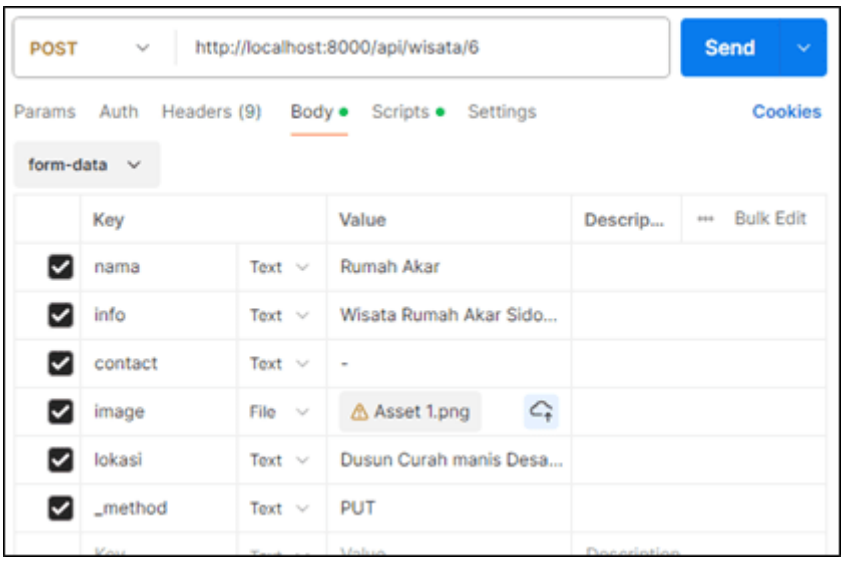


Gambar 5. Hasil uji endpoint GET by id

Setelah dilakukan pengujian dengan *request* id = 6 hasil yang dikembalikan adalah status true yang artinya data dapat ditemukan dan dapat ditampilkan. Pada kedua percobaan menggunakan id yang berbeda hasil yang ditampilkan memiliki status yang berbeda yaitu true untuk id = 6 dan false untuk id = 1. Tetapi keduanya memiliki status kode respons HTTP yang sama yaitu 200 berarti OK. Status kode ini diartikan bahwa permintaan yang dikirim berhasil diterima serta dapat diproses oleh server.

Pengujian *endpoint* PUT merupakan proses yang dilakukan untuk memastikan bahwa operasi PUT pada API berfungsi dengan baik. Sebelum melakukan request perlu disiapkan terlebih dahulu data yang akan diuji. Data diambil dari data yang sudah tersimpan di dalam database berdasarkan id. ID yang akan diuji adalah 10.

Akses URL yang digunakan adalah http:/localhost:8000/api/wisata/10. Method yang dipilih pada aplikasi postman adalah POST. Untuk alasan mengapa menggunakan method POST bukan method PUT padahal endpoin yang diuji adalah PUT akan dijelaskan setelah Gambar 6.



Gambar 6 Uji endpoint PUT

Key dan type yang perlu di masukkan kedalam form data pada body adalah sebagai berikut seperti pada Tabel 3.

Tabel 3. Key dan type untuk endpoint PUT

Key	*Type*	*Value*
nama	text	masukkan nama tempat
info	text	masukkan info wisata
contact	text	masukkan contact wisata
image	file	pilih gambar dari komputer
lokasi	text	masukkan lokasi tempat
_method	text	PUT

Proses yang dilakukan untuk menguji *endpoint* PUT hampir sama dengan *endpoint* POST *method* yang dipilih juga sama yaitu POST. Hal tersebut dikarenakan HTML hanya mendukung method POST dan GET sehingga untuk menggunakan method PUT dan DELETE ditambahkan parameter _method untuk menggubah *method* HTTP.

Setelah mengisi *form* data selanjutnya pilih *Send*. Kemudian data akan di kirim menggunakan *endpoint* yang awalnya adalah POST kemudian berganti dengan PUT dengan menggunakan parameter khusus yang disediakan laravel yaitu _method.

4. Kesimpulan

Perangkat lunak yang dikembangkan dengan model pengembangan menjadikan proses pengembangan menjadi lebih terstruktur. Tahapan – tahapan yang ada pada model pengembangan membantu dalam proses analisa sampai dengan proses pengujian tanpa melewatkan tahapan lain. Laravel adalah framework sederhana dan mudah digunakan sehingga mempercepat proses pengembangan serta cocok untuk membangun sebuah API. Pada penelitian ini API dibangun menggunakan konsep REST dengan nama RESTFul API. RESTful API berkomunikasi dengan menggunakan metode URL dan HTTP. URL dibuat didalam file route api pada laravel untuk mengakses resource. Metode HTTP yang digunakan pada penelitian ini adalah GET, POST, PUT dan DELETE. Penulis sebagai client mengirim request dengan metode HTTP kemudian diterima oleh API dan kemudian akan di response. Data dikirim dalam format JSON. Pengujian API dilakukan dengan bantuan alat uji API yaitu postman. Dari pengujian endpoin GET dengan route /api/wisata di dapatkan status kode 200 yang artinya permintaan dieksekusi dengan benar. Pada pengujian POST dengan route /api/wisata input benar mendapat response status kode 201 yang artinya permintaan dieksekusi dengan benar dan resource baru berhasil ditambahkan. Untuk pengujian dengan input data tidak sesuai format response dikirim dengan status kode 422 yaitu data tidak dapat di proses. Pada pengujian menggunakan method GET/{id} dengan route /api/wisata/{id} response status kode 200 yaitu berhasil meskipun data tidak ditemukan hanya memunculkan pesan data tidak ditemukan. Pengujian endpoin PUT dengan route /api/wisata/{id} dapat dieksekusi dengan benar dengan response status code 200. Pengujian endpoint DELETE dengan route

/api/wisata/{id} berhasil dilakukan dengan response status kode yang diberikan 200. Data yang dihasilkan pada penelitian ini disajikan dalam format JSON JSON dapat menampilkan data yang jelas dan mudah dibaca.

Daftar Pustaka

Almaiah, M. A., Hajjej, F., Lutfi, A., Al-Khasawneh, A., Alkhdour, T., Almomani, O., & Shehab, R. (2022). A Conceptual Framework for Determining Quality Requirements for Mobile Learning Applications Using Delphi Method. *Electronics, 11(5)*.

Amras, M., Yulianto, E., Dewi, D. A., & Setiawan, A. (2023). Implementation of the REST API Model using QR Codes on Mobile Devices to Order Parking Tickets. *International Journal of Advanced Computer Science and Applications (IJACSA), 14(10)*.

Chamim, A. N., Hardyanto, D. C., & Putra, K. T. (2021). Web-Based Flood Hazard Monitoring. *J Robot Control (JRC), vol. 2, no. 5*, 373–379.

Ehsan, A., Abuhaliqa, M. A., Catal, C., & Mishra, D. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions. *Applied Sciences, 12(9)*.

Falah, M. F., Panduman, Y. Y., Sukaridhoto, S., Tirie, A. W., Kriswantoro, M. C., Satria, B. D., & Usman, S. (2021). Comparison of Cloud Computing Providers for Development of Big Data and Internet of Things Application. *Indonesian Journal of Electrical Engineering and Computer Science Vol.22 No.3*.

He, J., Ying, Q., Qian, Z., Feng, G., & Zhang, X. (2020). Semi-Structured Data Protection Scheme Based On Robust Watermarking. *J Image Video Proc. 12*.

Hou, C., Zhao, Q., & Başar, T. (2021). Optimization of Web Service-Based Data-Collection System With Smart Sensor Nodes for Balance Between Network Traffic and Sensing Accuracy. *IEEE Transactions on Automation Science and Engineering, vol. 18, no. 4*, 2022-2034.

Jardim, R. R., Santos, M., Neto, E., da Silva, E., & de Barros, F. (2020). Integration Of The Waterfall Model With ISO/IEC/IEEE 29148:2018 For The Development Of Military Defense System. *IEEE Latin America Transactions, vol. 18, no. 12*, 2096-2103.

Ma, S.-P., Hsu, M.-J., Chen, H.-J., & Lin, C.-J. (2023). RESTful API Analysis, Recommendation, and Client Code Retrieval. *Electronics, 12(5)*.

Yas, Q. M., ALazzawi, A., & Rahmatullah, B. (2023). A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions. *Iraqi Journal for Computer Science and Mathematics: Vol. 4: Iss. 4*.

Zhang, M., & Arcuri, A. (2023). Open Problems in Fuzzing RESTful APIs: A Comparison of Tools. *ACM Trans. Softw. Eng. Methodol. 32*.