



Perancangan *Web Service* REST API Menggunakan PHP dan Framework Laravel di Tenta Tour Salatiga

Kent Gowell¹, Suprihadi^{2*}

^{1,2*} Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana, Kota Salatiga, Jawa Tengah, Indonesia.

article info

Article history:

Received 11 June 2023

Received in revised form

24 September 2023

Accepted 20 November 2023

Available *online* January 2024

DOI:

<https://doi.org/10.35870/jtik.v8i1.1269>

Keywords:

Tourism; Web Service; REST; Interoperability; Reservation.

Kata Kunci:

Pariwisata; Web Service; REST; Interoperabilitas; Reservasi.

abstract

Growth in Indonesia's tourism is increasing rapidly with over hundred thousands of yearly visitors. Travel agencies like Tenta tour provides travel and tourism planning services for customers, offering easy process and packages suited for any needs. Conventional methods used for planning tour plans are deemed inefficient and ineffective. To solve problems commonly found, a web service is designed to provide data transfer and management. The system is developed with a prototyping model until requirements are met, followed by system testing to ensure stable and fast performance. The research produced an integrated system by utilizing REST API web service and offering interoperability across systems. The system provides information on available travel packages and offers reservation and customization features. Results from performance testing show the capabilities of implemented web service design in managing requests and responses. The system provides responsive and easy data transfer and management.

abstract

Pertumbuhan pariwisata Indonesia meningkat pesat dengan lebih dari ratusan ribu pengunjung tahunan. Agen perjalanan seperti Tenta tour menyediakan layanan perencanaan perjalanan dan wisata untuk pelanggan, menawarkan proses yang mudah dan paket yang sesuai dengan kebutuhan. Metode konvensional yang digunakan untuk merencanakan perjalanan wisata dinilai kurang efisien dan efektif. Untuk mengatasi masalah yang ditemukan, sebuah web service dirancang untuk menyediakan transfer dan pengelolaan data. Sistem dikembangkan dengan model prototyping hingga kebutuhan terpenuhi, diikuti dengan pengujian sistem untuk memastikan kinerja yang stabil dan cepat. Penelitian ini menghasilkan sebuah sistem yang terintegrasi dengan memanfaatkan layanan web REST API dan menawarkan interoperabilitas antar sistem. Sistem menyediakan informasi tentang paket perjalanan yang tersedia dan menawarkan fitur reservasi dan kustomisasi. Hasil dari pengujian kinerja menunjukkan kemampuan web service yang diimplementasikan dalam mengelola request dan response. Sistem menyediakan transfer dan manajemen data yang responsif dan mudah.

Corresponding Author. Email: suprihadi@uksw.edu^{2}.

© E-ISSN: 2580-1643.

Copyright © 2024 by the authors of this article. Published by Lembaga Otonom Lembaga Informasi dan Riset Indonesia (KITA INFO dan Riset). This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.



Association for Computing Machinery
ACM Computing Classification System (CCS)

EBSCOhost

Communication and Mass Media Complete (CMMC)

1. Latar Belakang

Industri pariwisata Indonesia terus mengalami perkembangan, dengan jumlah pengunjung yang terus meningkat terutama setelah pandemi Covid-19 usai dan masyarakat mulai bebas bepergian. Tidak mengherankan jika bisnis pada industri pariwisata terus berkembang, dengan ratusan ribu wisatawan domestik maupun internasional melakukan perjalanan di berbagai destinasi wisata Indonesia. Tenta Tour Salatiga adalah bisnis di bidang *travel* atau pariwisata yang berperan sebagai penyedia layanan pariwisata umumnya menawarkan berbagai paket tur dan perjalanan wisata, sehingga pelanggan dapat melaksanakan kegiatan wisata dengan mudah, aman, dan lancar [1]. Paket perjalanan wisata yang disediakan dapat dirancang/disesuaikan ketika melakukan reservasi, sehingga sesuai dengan kebutuhan pelanggan saat merencanakan perjalanan wisata.

Permasalahan yang kerap muncul adalah ketika reservasi dan negosiasi memakan waktu yang cukup lama dan pertemuan dilakukan berulang-ulang, selain itu minimnya informasi mengenai detail layanan, harga, dan penawaran yang menjadi inti permasalahan. Pilihan paket perjalanan wisata yang monoton dan terpaku juga menyebabkan pelanggan kurang fleksibel dalam memilih. Sistem reservasi yang dilakukan secara manual dinilai kurang efisien, rawan kesalahan dalam pencatatan, pencarian data sulit, serta pembuatan laporan yang memakan banyak waktu sehingga menjadi tidak efektif. Maka dari itu, perlu dikembangkan sebuah sistem reservasi terpadu yang berperan menyediakan layanan akses dan informasi yang fleksibel, efisien dan responsif. Sistem informasi yang terpadu mempermudah proses reservasi sehingga dapat dilakukan dengan lebih mudah dan pengolahan data lebih efektif [2].

Perancangan sistem Biro perjalanan pariwisata berbasis web bertujuan memudahkan pelanggan dalam melakukan proses pemesanan, mempermudah dan memperluas media promosi, sehingga memberikan kemudahan bagi pihak penyedia jasa wisata dalam mengembangkan usaha. Implementasi sistem web dengan memanfaatkan *web service* memudahkan pelanggan dalam melakukan pencarian, perencanaan hingga pemesanan paket perjalanan wisata, selain itu pihak administrator juga

lebih mudah dalam pengolahan data [3]. *Web service* berperan sebagai unit penyedia fungsi layanan dalam sebuah sistem atau aplikasi menjadi komponen perangkat lunak yang terstandarisasi, dapat digunakan dan diperbaharui secara dinamis dan independen [4]. Layanan - layanan yang disediakan oleh suatu *web service* berbentuk informasi melalui transfer data bertipe JSON atau XML antar sistem dalam suatu jaringan melalui protokol HTTP, sehingga memungkinkan interaksi antara antar sistem dengan dukungan interoperabilitas yang ditawarkan [5].

Aplikasi berbasis web dengan layanan *web service* dikembangkan menggunakan bahasa pemrograman PHP dengan framework Laravel dan arsitektur REST. Implementasi sistem *web service* memudahkan dalam proses developer dan proses pengembangan. Karena dengan menggunakan metode *web service* tidak akan dibatasi oleh bahasa pemrograman yang digunakan [6]. Sistem dengan arsitektur REST dipilih untuk mengatasi keterbatasan platform, mulai dari aplikasi yang sulit ditemukan oleh masyarakat, keterbatasan *platform*, dan penyediaan layanan yang masih terbilang kurang mumpuni [7]. *Framework* Laravel dipilih karena mudah dipahami dan *powerful* juga menyediakan berbagai komponen seperti autentikasi, *routing*, *session manager*, *caching*, dan banyak lagi lainnya. Fitur seperti sistem *packaging* modular dengan *dependency manager* tersendiri, *tools* untuk migrasi database, dukungan unit testing yang sudah terintegrasi, dan banyak lagi lainnya memungkinkan pengembangan aplikasi web sederhana sampai tingkat kompleks. Penerapan *web service* dengan arsitektur REST memungkinkan pengembangan sistem yang sederhana, *scalable*, efektif, aman, dan dapat diandalkan sehingga perancangan API sederhana dan kokoh berbasis Ajax dan *web service* Restful menjadi lebih mudah. Arsitektur REST memungkinkan sistem yang melakukan *request* untuk mengakses dan manipulasi representasi dari *web resource* dengan menggunakan metode - metode HTTP seperti GET, POST, PUT, dan DELETE [8]. Penelitian ini mengimplementasikan sistem *web service* sesuai dengan kebutuhan dan permasalahan yang dimiliki oleh Tenta Tour Salatiga dalam mengembangkan proses bisnis yang ada sehingga menjadi lebih efektif, fleksibel, dan efisien. Alur pengembangan sistem yang diterapkan menggunakan *prototyping model* yang merupakan salah satu jenis dari *Software Development Life Cycle* (SDLC). Model *prototyping* memungkinkan pengembangan prototipe

awal sebagai gambaran dari sistem, sehingga proses pengembangan disesuaikan dengan tahapan sistem dan disesuaikan dengan hasil pengujian pengguna melalui beberapa iterasi berdasarkan penyesuaian prototipe sebelumnya hingga didapat hasil akhir yang dibutuhkan [9].

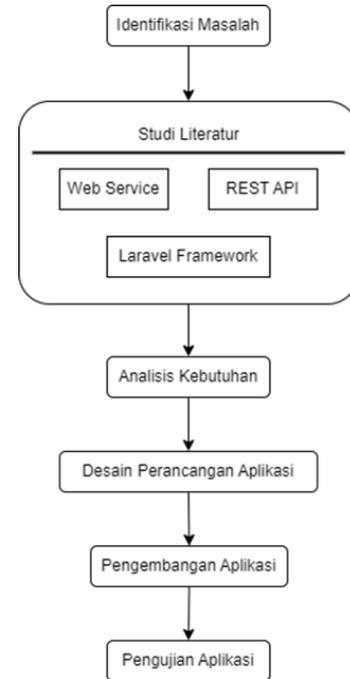
Penelitian terdahulu yang dilakukan oleh Tedyyana dan rekan (2021), yang melakukan revamp pada keamanan Web Service menggunakan REST API pada framework Lumen. Mereka mempertimbangkan aspek kecepatan load data dan kemudahan deploy di shared hosting, hal ini memiliki implikasi penting dalam merancang Web Service REST API untuk Tenta Tour Salatiga guna memastikan keamanan dan kinerja yang optimal [11]. Penelitian oleh Herdiyatomoko (2022) membahas desain sistem backend berbasis REST API menggunakan Framework Laravel 7. Mereka menyoroti pentingnya REST API dalam mendukung layanan pencarian dan media sosial berbasis fitur tunggal, yang dapat memberikan ide-ide berharga dalam implementasi REST API dengan efektif dalam konteks Tenta Tour Salatiga [12]. Penelitian oleh Suwarno dan Afandi (2022), yang menganalisis perbandingan antara Codeigniter dan Yii framework dalam perancangan website, meskipun tidak secara langsung berkaitan dengan Laravel, namun perbandingan tersebut dapat membantu dalam pemilihan pendekatan terbaik dalam merancang Web Service REST API [13]. Dengan mengintegrasikan temuan dari penelitian-penelitian terdahulu yang relevan, perancangan Web Service REST API menggunakan PHP dan Framework Laravel di Tenta Tour Salatiga dapat dioptimalkan untuk keamanan, kinerja, dan efisiensi yang maksimal.

Penelitian oleh Grasiyas dan Supriyadi (2022), yang membahas perancangan REST Web Service untuk pengembangan sistem pengajuan simpan pinjam pengembangan layanan serupa [14], dimana REST web service untuk meningkatkan keamanan dan sangar handal diimplementasikan agar menghasilkan tingkat efisiensi dan keamanan yang handal untuk [15]. Buwono (2019), menyebutkan *web services* dengan format JSON yang dapat membantu sistem informasi untuk mendapatkan informasi yang dibutuhkan dan mendapatkan integritas data yang dihasilkan antara sistem informasi tersebut [16]. Dengan mempertimbangkan temuan dari penelitian-

penelitian tersebut, perancangan Web Service REST API menggunakan PHP dan Framework Laravel di Tenta Tour Salatiga dapat diarahkan menuju tingkat keamanan, kinerja, dan efisiensi yang optimal.

2. Metode Penelitian

Tahapan penelitian dilakukan secara bertahap, dengan alur penelitian dapat digambarkan seperti bagan berikut:



Gambar 1. Alur Tahapan Penelitian

Identifikasi masalah dilakukan pada tahap penelitian untuk menentukan pokok-pokok permasalahan yang akan diselesaikan melalui penelitian, selain itu batasan-batasan dari penelitian juga ditentukan pada tahap ini agar tujuan dan manfaat penelitian sesuai dengan kebutuhan penelitian. Setelah melakukan perumusan masalah, dilakukan studi literatur mengenai topik yang diangkat pada penelitian ini, sehingga dapat mendukung penyelesaian masalah pada penelitian. Analisis kebutuhan dilakukan setelah studi literatur untuk mencari rincian kebutuhan, terutama mengenai aplikasi yang akan dikembangkan beserta fitur – fitur yang diperlukan. Dari hasil analisa data – data dan proses bisnis sebelumnya, dapat menjadi gambaran awal rancangan sistem yang akan dikembangkan. Tahap analisis kebutuhan dengan melakukan desain sistem berbentuk UML dan menentukan *requirement* untuk pengembangan aplikasi

secara spesifik. Visualisasi sistem mulai dari rancangan data berbentuk (ERD), cara guna (*Use Case*) hingga *mockup* atau bentuk dari *interface* sistem. Setelah analisis kebutuhan dilakukan dan *requirement* awal telah ditentukan, sistem dikembangkan sesuai dengan rancangan. *Prototyping Model* digunakan sebagai model *Software Development Life Cycle* (SDLC) atau siklus hidup pengembangan perangkat lunak sehingga proses pengembangan fleksibel dengan *requirement* yang masih bisa berubah dan disesuaikan dengan kebutuhan. Tahapan awal siklus diawali dengan *prototyping/ development* dilanjutkan dengan pengujian sistem, dan hasil pengujian dijadikan sebagai acuan untuk *requirement* baru untuk perbaikan sistem pada iterasi *prototype* selanjutnya. Proses siklus pengembangan dilakukan hingga iterasi *prototype* yang dihasilkan dinilai telah sesuai dengan kebutuhan sistem. Pengujian aplikasi dilakukan setelah sistem selesai dikembangkan dengan menggunakan Jmeter untuk melakukan pengujian HTTP *request*. Pengujian HTTP *request* digunakan untuk menguji performa dari *web service* dalam menerima *request* dan mengirim *response*. sehingga aplikasi yang dikembangkan sesuai dengan rancangan yang dibuat.

3. Hasil dan Pembahasan

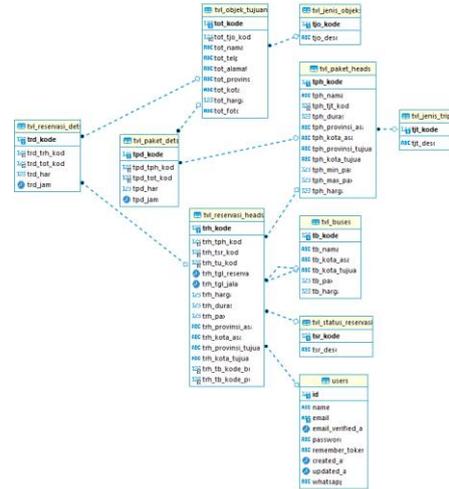
Berdasarkan permasalahan yang telah dirumuskan, peneliti melakukan analisa kebutuhan sistem sehingga dapat digambarkan dengan UML (*Universal Markup Language*).



Gambar 2. Use Case Diagram

Use Case Diagram pada gambar 2 menggambarkan fitur dan interaksi antara pengguna dan sistem. Dalam sistem, jenis pengguna dibedakan menjadi dua aktor, yaitu *customer* dan *admin*. Aktor *customer*

memiliki hak untuk melihat dan mencari paket - paket wisata yang disediakan, melakukan registrasi dan login akun pengguna, melakukan reservasi paket pilihan, dan melihat riwayat-riwayat reservasi perjalanan wisata yang telah dilakukan. Aktor *admin* dapat melakukan aktivitas pengolahan data CRUD (*Create, Read, Update, Delete*) untuk entitas paket wisata, objek wisata, transportasi, hingga reservasi.



Gambar 3. Entity Relation Diagram

Entity Relation Diagram (ERD) pada gambar 3 menggambarkan hubungan antar entitas pada database, dan berfungsi sebagai alat bantu dalam pembuatan database dan memberikan gambaran cara kerja data hubungan database yang akan dibuat sesuai dengan proses bisnis yang sudah ada secara detail [10]. Dari desain ERD dapat terlihat bahwa sistem basis data yang dirancang sudah secara jelas dan detail menggambarkan relasi data di mana setiap entitas pada database berhubungan satu sama lain, setiap entitas memiliki atribut utama *Primary Key* dan atribut lainnya sebagai definisi hubungan antar entitas yang deskriptif. Hasil dari penelitian ini berupa sistem pengolahan data berbasis web dengan API sebagai *controller* pada sistem untuk mengolah *request* dan *response* dari *client* dalam transaksi CRUD data. Pada penelitian ini, sistem diimplementasikan dengan memanfaatkan *business layer* dalam bentuk *controller* sebagai *web service* dengan arsitektur REST untuk menyediakan data kepada *client* dan *presentation layer* berbentuk halaman web untuk melakukan pengolahan data.

Kode Program 1. Fungsi *query select* paket

```

function getAll() {
    $resp = DB::table("tbl_paket_heads")
    ->select("tph_kode", "tjt_desc",
    "tph_nama", "tph_durasi", "tph_harga",
  
```

```
"tph_provinsi_asal",
"tph_provinsi_tujuan",
"tph_kota_tujuan",      "tph_kota_asal",
"tph_min_pax", "tph_max_pax")
->join(
"tvl_jenis_trips","tph_tjt_kode","=", "t
jt_kode")
->orderBy("tph_kode")
->get();
        return $resp;
    }
}
```

Kode program 1 adalah kode yang digunakan sebagai *endpoint* API untuk mengambil list paket wisata yang tersedia dengan melakukan *query* ke database dengan menggunakan Eloquent ORM dari *Framework* Laravel sebagai *query builder*. Dengan menggunakan Eloquent, pengambilan dan pengolahan data dari database menjadi lebih sederhana dan bisa dibangun sesuai dengan kebutuhan fungsi.

Kode Program 2. Fungsi *query insert* data paket detail

```
//save to DB

$paketHead =
tvl_paket_head::create($data);
if ($paketHead) {
    return response()->json([
        'success' => true,
        'message' => 'paket tujuan
Created'],
        201);
}
//Failed to create Data
Return response()->json([
    'success' => false,
    'message' => 'paket tujuan Failed
to save'], 409);
```

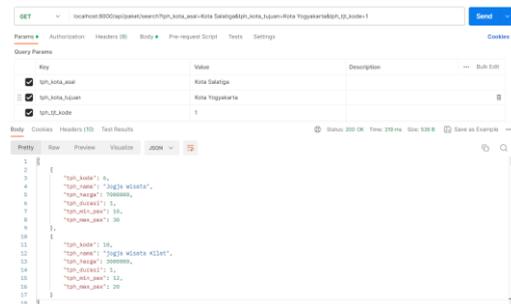
Selain mengambil data, dengan Eloquent ORM juga memungkinkan transaksi seperti mengubah, menghapus, dan memasukkan data ke database pada *controller* API. Seperti pada kode program 2, fungsi *endpoint* dapat ditambahkan logika pemrograman seperti perulangan dan percabangan sehingga ketika digabungkan dengan *query builder* dapat mengembalikan *response* menurut kondisi hasil insert data ke database dimana apabila transaksi yang dilakukan berhasil maka *response* yang diberikan berupa status 'success' TRUE, 'message' berhasil, dan kode status 201 namun apabila transaksi gagal dilakukan maka *response* berupa 'success' FALSE, 'message' gagal dan kode status 409.

Kode Program 3. Kode *endpoint route* API

```
//list endpoint paket wisata

Route::group(['prefix' =>
'paket'],function(){
    Route::get('/all',[PaketHead::class,'g
etAll']);
    Route::get('/find',[PaketHead::class,'
findPaket'];
    Route::get('/search',[PaketHead::class
,'searchPaket']);
    Route::post('/add',[PaketHead::class,'
store']);
    Route::post('/update',[PaketHead::clas
s,'update']);
    Route::get('/delete',[PaketHead::class
,'delete']);
    Route::get('/trip',[PaketHead::class,'
jenisTrip']);
    Route::post('/harga',[PaketHead::class
,'updateHarga']);
```

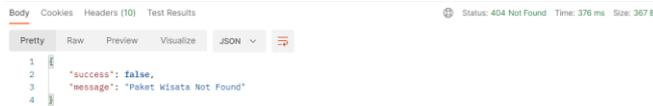
Pada kode program 3, fungsi – fungsi pada *controller* dijadikan *route* sebagai *endpoint* yang dikonfigurasi pada file *routes/api.php* sehingga dapat diakses melalui URL dengan metode HTTP yang sesuai dengan konfigurasi masing-masing *route*. Untuk menerima *response* dari fungsi “getAll” pada *controller* “PaketHead”, pengguna dapat mengirim *request* ke *route* /api/paket/all dengan metode HTTP GET.



Gambar 4. Pengiriman *request* ke *web service*

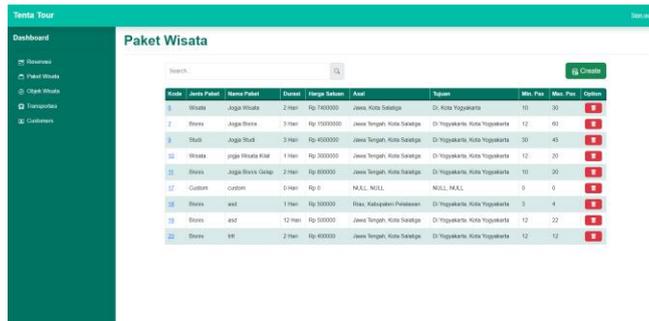
Gambar 4 adalah hasil *request* dan *response* yang dikirim ke *endpoint* pencarian paket dengan *route* /api/paket/search. Untuk melakukan pencarian paket wisata, diperlukan tiga parameter untuk disertakan pada *request* yaitu kota asal, kota tujuan, dan jenis paket wisata. Apabila hasil pencarian pada *controller* berdasarkan parameter memperoleh hasil, maka akan dikembalikan *response* berupa *array* JSON dengan kode status 200. Sedangkan apabila pencarian berdasarkan parameter tidak ditemukan maka *response* yang diterima berupa kode status 404 (*not found*) dan status *success* FALSE seperti yang ditunjukkan pada gambar

5.



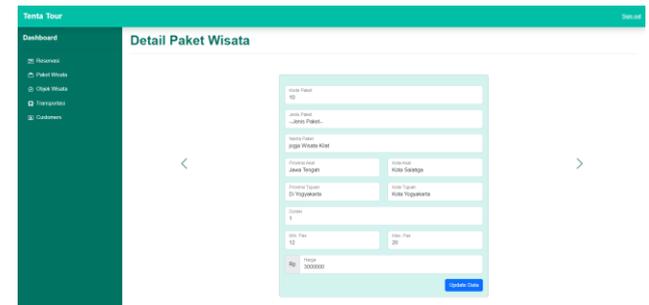
Gambar 5. Response kode 404

Penerapan aplikasi dan pemanfaatan *web service* memungkinkan pengguna untuk mengakses dan mengolah data atau *resource* yang disediakan sistem melalui berbagai perangkat tidak terpaku pada satu jenis perangkat maupun *platform* tertentu. Selain pengembangan *web service* dengan memanfaatkan *controller* sebagai *business layer*, implementasi pada *presentation layer* berupa halaman web juga dikembangkan untuk keperluan olah data.



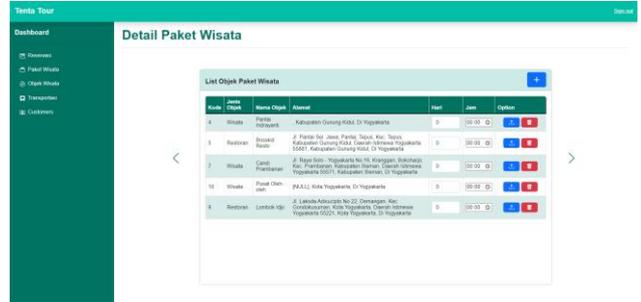
Gambar 6. Halaman *dashboard* paket wisata

Gambar 6 merupakan halaman paket wisata untuk melakukan transaksi CRUD data paket wisata yang tersedia. Data disajikan dalam bentuk tabel dengan informasi mengenai paket seperti nama, lokasi asal dan tujuan, biaya dan lain – lain. Di bagian atas tabel terdapat fungsi pencarian untuk mencari paket wisata berdasarkan kode atau nama yang sesuai dan tombol ‘Create’ untuk menambahkan data baru. Data paket dapat dihapus dengan menekan tombol hapus pada opsi bagian kanan, atau diubah dan diperbarui dengan menekan kode paket untuk diarahkan menuju halaman detail paket.



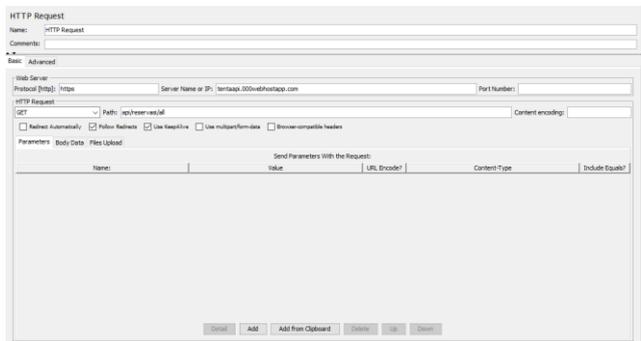
Gambar 7. Halaman detail paket wisata

Pada halaman detail paket wisata, pengguna dapat mengubah data-data dari paket wisata seperti asal, tujuan, jenis wisata, durasi wisata, dsb. Halaman ini juga memuat tabel yang merupakan daftar objek wisata yang ditawarkan oleh paket seperti pada gambar 8.



Gambar 8. Daftar detail objek paket wisata

Objek wisata dapat dihapus dan ditambahkan ke daftar yang akan disediakan pada paket tersebut berdasarkan daftar objek wisata yang tersimpan dalam database berdasarkan lokasi tujuan wisata. Hari dan waktu kunjungan masing – masing objek wisata dapat diubah secara individu sesuai kebutuhan. Berdasarkan daftar objek tujuan paket wisata, sistem akan secara otomatis menghitung harga dari jumlah total biaya objek wisata yang dipilih. Dengan halaman web terpadu, pengolahan data pada sistem dan *database* menjadi lebih terstruktur dan mudah. Pengujian sistem dilakukan dengan mengirimkan *HTTP request* ke sistem *web service* yang telah diimplementasikan pada layanan *web hosting*. Konfigurasi pengujian pada Jmeter diatur dengan jumlah *thread* sebanyak 50 dengan perulangan sebanyak dua kali tanpa *ramp-up period*, sehingga dihasilkan sebanyak 100 sampel.



Gambar 9. Konfigurasi *HTTP request*

Gambar 9 merupakan konfigurasi dari *sampler HTTP request* pada Jmeter. Protocol *https* digunakan pada *web service* dengan *server name* berupa alamat IP atau URL

dari *web service*, *route* yang diuji adalah /api/reservasi/all dengan method GET tanpa parameter ataupun body data. *Response* yang diterima berupa *array* yang berisi daftar reservasi sebanyak 71 data yang tersimpan pada database dengan format JSON.



Gambar 10. Grafik hasil pengujian Jmeter

Data yang dihasilkan pada grafik hasil adalah waktu *request* rata-rata (*average*) sebesar 1216 ms, nilai tengah (*median*) 1040 ms, dan standar deviasi (*deviation*) 402 ms. Hasil grafik pada gambar 10 juga mendapatkan nilai *throughput* sebesar 1,734 *request* yang dapat diolah setiap menitnya, atau sebanyak 29 *request* per detik.

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
45	22:14:16.633	Thread Group 1-17	HTTP Request	904	Success	11709	147	904	0
46	22:14:16.633	Thread Group 1-14	HTTP Request	961	Success	11709	147	900	0
47	22:14:16.633	Thread Group 1-17	HTTP Request	937	Success	11709	147	927	0
48	22:14:16.633	Thread Group 1-4	HTTP Request	931	Success	11709	147	921	0
51	22:14:16.630	Thread Group 1-3	HTTP Request	936	Success	11709	147	926	0
52	22:14:16.644	Thread Group 1-3	HTTP Request	971	Success	11709	147	970	0
53	22:14:16.644	Thread Group 1-33	HTTP Request	935	Success	11709	147	924	0
54	22:14:16.644	Thread Group 1-31	HTTP Request	935	Success	11709	147	923	0
55	22:14:16.644	Thread Group 1-38	HTTP Request	937	Success	11709	147	927	0
56	22:14:16.702	Thread Group 1-4	HTTP Request	940	Success	11709	147	937	0
58	22:14:16.703	Thread Group 1-4	HTTP Request	941	Success	11709	147	949	0
59	22:14:16.800	Thread Group 1-30	HTTP Request	940	Success	11709	147	946	0
6	22:14:16.800	Thread Group 1-14	HTTP Request	938	Success	11709	147	924	462
7	22:14:16.800	Thread Group 1-17	HTTP Request	941	Success	11709	147	946	0
8	22:14:16.800	Thread Group 1-1	HTTP Request	938	Success	11709	147	926	463
9	22:14:16.800	Thread Group 1-1	HTTP Request	938	Success	11709	147	926	463
10	22:14:16.800	Thread Group 1-3	HTTP Request	935	Success	11709	147	933	460
11	22:14:16.800	Thread Group 1-17	HTTP Request	941	Success	11709	147	943	464
12	22:14:16.800	Thread Group 1-4	HTTP Request	939	Success	11709	147	941	463
13	22:14:16.800	Thread Group 1-30	HTTP Request	937	Success	11709	147	937	467
14	22:14:16.800	Thread Group 1-3	HTTP Request	939	Success	11709	147	949	460
15	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
16	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
17	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
18	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
19	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
20	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
21	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
22	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
23	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
24	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
25	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
26	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
27	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
28	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
29	22:14:16.813	Thread Group 1-38	HTTP Request	935	Success	11709	147	935	480
30	22:14:16.822	Thread Group 1-46	HTTP Request	996	Success	11709	147	988	494

Gambar 11. Tabel hasil pengujian Jmeter

Gambar 11 menyajikan data dari Jmeter dalam bentuk tabel dengan rincian *request* dan *response* dari setiap sampel yang dilakukan. Setiap sampel mengirim data *request* sebesar 147 Byte dan menerima *response* berukuran 11709 Byte. *Sample time* atau waktu yang dibutuhkan sampel untuk menerima *response* sejak *request* dikirimkan tertinggi didapat pada nilai 2602 ms dengan waktu terendah pada 904 ms. Data lebih terperinci dari setiap *request* yang dikirimkan oleh sampel dapat dilihat pada *result tree listener* dari Jmeter, di mana hasil sampel, *request*, *header*, dan *response* dijabarkan seperti pada gambar 12.



Gambar 12. Rincian *thread* HTTP *request*

Hasil dari *performance testing* terhadap 100 sampel menggunakan *HTTP request* Jmeter menunjukkan bahwa sistem *web service* dengan arsitektur REST API mumpuni dalam menyediakan layanan pengolahan data. Rata-rata waktu yang diperlukan untuk mengolah data *request* sebesar 1216 ms atau 1.2 detik dengan waktu tersingkat sebesar 904 ms dan terpanjang sebesar 2602 ms. Nilai *throughput* dari API *web service* menunjukkan jumlah *request* yang dapat diolah sebanyak 1734 per menit atau setara dengan 29 *request* bersamaan setiap detiknya dengan mengirimkan data *response* sebesar 11709 Byte.

4. Kesimpulan

Berdasarkan penelitian yang dilakukan dan hasil penerapan sistem terpadu berbasis *web service* dengan menerapkan arsitektur REST menyediakan layanan pengolahan data dan informasi yang tersimpan pada database secara responsif tanpa keterbatasan *platform* berkat fungsi interoperabilitas yang ditawarkan. Selain API *web service*, hasil penelitian ini juga diimplementasikan pada halaman web untuk transaksi pengolahan data *travel agent* oleh administrator. Dengan memanfaatkan halaman web yang terhubung pada *web service*, pengguna dapat mengakses sistem pada berbagai jenis perangkat. Sistem memungkinkan pengembangan ke depan yang fleksibel dan efisien, implementasi dari sistem dapat dikembangkan sesuai dengan kebutuhan. Pengujian *performance testing* yang dilakukan pada sistem *web service* memperoleh hasil *average request time* sebesar 1216ms atau 1,2 detik dengan nilai tengah 1040 ms, dan standar deviasi sebesar 402 ms. Nilai rata-rata yang lebih besar daripada nilai tengah mengindikasikan distribusi data miring positif, dimana sebagian besar data berada pada nilai rendah dengan beberapa hasil nilai ekstrim. Dari 100 sampel pengujian, didapatkan *request time* terendah pada angka 904ms dan tertinggi 2602 ms sehingga dapat disimpulkan bahwa web service dapat

mengolah data dengan responsif dan efisien. Hasil *throughput* menunjukkan bahwa *web service* mumpuni dalam mengolah *request* dalam jumlah banyak sekaligus, dengan jumlah sebesar 1734 *request* per menit dan data yang ditransfer berukuran 11,7 KB pada setiap *response*.

Berdasarkan hasil dari penelitian yang dilakukan, terlihat masih terdapat kekurangan baik secara penerapan dan pengembangan dari sistem aplikasi, sehingga masih terdapat banyak ruang untuk pengembangan dan perbaikan ke depan. Fitur promosi dan pemesanan dapat dilengkapi lagi dengan sistem pembayaran baik secara manual maupun dengan *payment gateway*, sehingga sistem menyediakan fitur lebih lengkap bagi pengguna. Pemanfaatan *web service* juga tidak terpaku pada satu platform tertentu, sehingga dapat digunakan pada aplikasi – aplikasi berbasis web maupun *mobile* lainnya dengan memanfaatkan API dari sistem.

5. Daftar Pustaka

- [1] Wirawan, I.M.A., Wirawan, D.N., Kurniasari, N.M.D. and Merati, K.T.P., 2020. Travel agent and tour guide perceptions on travel health promotion in Bali. *Health Promotion International*, 35(1), pp.e43-e50. DOI: 10.1093/heapro/day119.
- [2] Oktapiah, T. and Hasti, N., 2020. Sistem Informasi Reservasi Paket Wisata Berbasis Web. *Jurnal Teknik Informatika*, 12(1), pp.1-7.
- [3] Mukminin, A. and Rachman, R., 2019. Reservasi Destinasi Wisata Menggunakan Algolia Search Pada Bandung Trans Service Berbasis Web. *Jurnal Responsif: Riset Sains dan Informatika*, 1(1), pp.70-76. DOI: <https://doi.org/10.51977/jti.v1i1.69>.
- [4] Song, Y., 2021. Web service reliability prediction based on machine learning. *Computer Standards & Interfaces*, 73, p.103466. DOI: <https://doi.org/10.1016/j.csi.2020.103466>.
- [5] Rizal, R. and Rahmatulloh, A., 2019. Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan. *Jurnal Ilmiah Informatika*, 7(01), pp.54-59. DOI: <https://doi.org/10.33884/jif.v7i01.1004>.
- [6] Sutrisno, S., Asyidiq, M. and Santoso, S., 2018. Perancangan sistem pemasangan iklan online pada aplikasi e-commerce (E-gemana) menggunakan metode restful api dan framework laravel. *Jurnal Ilmiah Sains dan Teknologi*, 2(2), pp.119-132.
- [7] Somya, R. and Nathanael, T.M.E., 2019. Pengembangan Sistem Informasi Pelatihan Berbasis Web Menggunakan Teknologi Web Service Dan Framework Laravel. *Jurnal Techno Nusa Mandiri*, 16(1), pp.51-58. DOI: <https://doi.org/10.33480/techno.v16i1.164>.
- [8] Neumann, A., Laranjeiro, N. and Bernardino, J., 2018. An analysis of public REST web service APIs. *IEEE Transactions on Services Computing*, 14(4), pp.957-970. DOI: <https://doi.org/10.1109/TSC.2018.2847344>.
- [9] Wulandari, D.A.N., Bahar, A.A.H., Arfananda, M.G. and Apriyani, H., 2021. Prototyping Model in Information System Development of Al-Ruhamaa'bogor Yatim Center Foundation. *Jurnal Pilar Nusa Mandiri*, 17(2), pp.127-136. DOI: <https://doi.org/10.33480/pilar.v17i2.2375>
- [10] Azzahra, Z.F. and Anggoro, A.D., 2022. Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review. *INTECH (Informatika dan Teknologi)*, 3(1), pp.8-11.
- [11] Tedyyana, A., Fauzi, M. D., & Ratnawati, F. (2021). Revamp keamanan web service milik pt xyz menggunakan rest api. *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 12(1), 1-10. DOI: <https://doi.org/10.31849/digitalzone.v12i1.6378>

- [12] Herdiyatomoko, H. F. (2022). Desain sistem backend berbasis rest api menggunakan framework laravel 7. *Skanika*, 5(2), 136-144. DOI: <https://doi.org/10.36080/skanika.v5i2.2947>
- [13] Suwarno, S. and Afandi, A. (2022). Analisis perbandingan codeigniter dan yii framework pada perancangan website rencana anggaran biaya. *Jurnal CoSciTech (Computer Science and Information Technology)*, 3(3), 249-258. DOI: <https://doi.org/10.37859/coscitech.v3i3.4338>
- [14] Grasiyas, N. J. and Supriyadi, S. (2022). Perancangan rest web service untuk pengembangan sistem pengajuan simpan pinjam pt. xyz. *Jurnal Teknologi Informatika Dan Komputer*, 8(2), 72-86. DOI: <https://doi.org/10.37012/jtik.v8i2.1182>
- [15] Wachyuni, D. and Candra, J. C. (2022). Implementasi web service untuk manajemen akun linux pada server badan litbang perhubungan menggunakan php. *KRESNA: Jurnal Riset Dan Pengabdian Masyarakat*, 2(2), 202-211. DOI: <https://doi.org/10.36080/jk.v2i2.49>
- [16] Buwono, R. C. (2019). *Web services* menggunakan format json. *Respati*, 14(2). DOI: <https://doi.org/10.35842/jtir.v14i2.282>.