

Perancangan dan Implementasi RESTful API untuk Aplikasi *Mobile* Pembelajaran Flora dan Fauna pada Google Cloud Platform

Calista Chandra¹, Frans Wijaya², Jason Andrew Gunawan³, James Rafferty Lee⁴, Ade Maulana^{5*}
^{1,5*}Sistem Informasi (Kampus Kota Medan), Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Pelita Harapan, Medan, Indonesia
^{2,3,4}Teknik Informatika, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
Email: ¹03081200012@student.uph.edu, ²franswijaya403@gmail.com, ³jac.gunawan@gmail.com, ⁴jamesrafe10@gmail.com, ^{5*}ade.maulana@lecturer.uph.edu

ABSTRACT

In recent years, there has been an increase in smartphone ownership among school-aged children. This increase was influenced by the Covid-19 pandemic, which caused restrictions on outdoor activities and activity adjustments such as online learning. Even though the Covid-19 pandemic is now over, children spend more time at home with their smartphones. Meanwhile, playing and learning outdoors is very beneficial for their social, emotional, cognitive, and physical development. Therefore, EksFlorasi exists as a solution in the form of a mobile-based educational application to encourage children to explore nature around them and learn about flora dan fauna using their smartphones. To support EksFlorasi, a web API was designed and developed using the waterfall method. The API functions as the mobile app's back-end component and was built using Node.js with the Express JS framework and implemented REST (REpresentational State Transfer). The RESTful API was then deployed using Google Cloud Platform's cloud computing services to make EksFlorasi accessible to users across Indonesia with the lowest latency possible. To ensure that the RESTful API functions as intended, the deployment on Google Cloud Platform utilized a combination of Cloud Run, Cloud SQL and Cloud Storage services. Based on the research results, the RESTful API which has been successfully developed and deployed on Google Cloud Platform is functioning well and able to deliver optimal performance for up to 50 concurrent users.

Keywords: RESTful API, Google Cloud Platform, EksFlorasi, Mobile App.

ABSTRAK

Dalam beberapa tahun terakhir, terjadi peningkatan angka kepemilikan *smartphone* pada anak-anak usia sekolah. Peningkatan tersebut dipengaruhi oleh pandemi Covid-19 lalu yang menyebabkan pembatasan kegiatan dan penyesuaian kegiatan seperti kegiatan belajar mengajar daring. Walaupun sekarang pandemi Covid-19 tersebut telah berakhir, anak-anak lebih banyak menghabiskan waktu di rumah saja dengan *smartphone* mereka. Padahal, bermain dan belajar di luar ruangan sangat bermanfaat bagi perkembangan sosial, emosional, kognitif, hingga fisik anak. Oleh karena itu, EksFlorasi hadir sebagai solusi dalam bentuk aplikasi edukasi berbasis *mobile* untuk mendorong anak-anak dalam mengeksplorasi lingkungan alam disekitar mereka dan mengenal flora dan fauna menggunakan *smartphone* mereka. Untuk mendukung EksFlorasi, dilakukan perancangan dan pengembangan *web API* menggunakan metode *waterfall*. *API* tersebut berfungsi sebagai komponen *back-end* aplikasi dan dikembangkan menggunakan Node.js dengan *framework* Express JS dan menerapkan REST (REpresentational State Transfer). RESTful API tersebut kemudian di-*deploy* menggunakan layanan *cloud computing* dari Google Cloud Platform agar EksFlorasi dapat diakses oleh pengguna di seluruh Indonesia dengan *latency* terendah. Agar RESTful API ini dapat berfungsi sebagaimana mestinya, *deployment* pada Google Cloud Platform dilakukan menggunakan kombinasi layanan Cloud Run, Cloud SQL, dan Cloud Storage. Berdasarkan hasil penelitian, RESTful API yang telah berhasil dikembangkan dan di-*deploy* pada Google Cloud Platform tersebut telah berfungsi dengan baik dan dapat memberikan performa yang optimal bagi sekitar 50 *user* secara bersamaan.

Kata Kunci: RESTful API, Google Cloud Platform, EksFlorasi, Aplikasi *Mobile*.

1. Pendahuluan

Di era digital ini, *smartphone* telah menjadi bagian penting dalam kehidupan sehari-hari anak-anak baik dari sisi hiburan maupun edukasi. Hal ini didukung oleh data yang dipublikasikan oleh BPS [1], yang mana pada tahun 2022, terdapat 67,88% penduduk Indonesia berusia lima tahun ke atas sudah memiliki ponsel dan angka tersebut terus meningkat. Salah satu penyebab peningkatan tersebut merupakan pandemi Covid-19 lalu yang menyebabkan pembatasan aktivitas masyarakat di luar rumah dan penyesuaian kegiatan seperti kegiatan pembelajaran daring. Selain pandemi Covid-19, faktor-faktor lain seperti larangan tua dan kurangnya fasilitas yang memadai juga menyebabkan anak-anak usia sekolah sekarang lebih banyak menghabiskan waktu di dalam ruangan. Padahal, bermain dan belajar di luar ruangan sangat bermanfaat bagi pertumbuhan dan perkembangan anak-anak. Hal ini didukung oleh hasil penelitian yang dilakukan oleh [2] yang menyatakan bahwa bermain dan belajar di luar ruangan sangat bermanfaat dalam mendukung perkembangan sosial, emosional, kognitif, hingga fisik anak. Ditambah lagi, anak-anak yang rutin bermain dan belajar di luar ruangan juga memiliki kepercayaan diri dan antusiasme belajar yang lebih tinggi, serta rasa ingin tahu lebih terkait lingkungan alam di sekeliling mereka.

Oleh karena itu, tingkat penggunaan *smartphone* yang tinggi di kalangan anak-anak usia sekolah menghadirkan potensi penggunaan perangkat *mobile* dalam mendorong anak-anak dalam menghabiskan waktu di luar ruangan dan belajar mengenai lingkungan alam sekitar mereka. Ditambah lagi, dikarenakan kondisi geografis Indonesia yang mana setiap daerah memiliki ciri khas tertentu, terdapat persebaran flora dan fauna yang beragam dan berbeda-beda di setiap wilayah Indonesia [3]. Menurut penelitian yang dilakukan oleh [4], anak-anak yang belajar mengenai alam di luar ruangan memiliki rasa ingin tahu yang lebih tinggi dan merasa terhubung dengan alam. Selain itu, hasil dari penelitian yang dilakukan oleh [5] menunjukkan bahwa penggunaan perangkat *mobile* dalam proses pembelajaran dapat memberikan manfaat dimana para pelajar dapat mengakses informasi atau bahan pembelajaran tanpa terikat waktu dan tempat. Penggunaan aplikasi pembelajaran berbasis *mobile* juga dapat mendukung kolaborasi dan interaksi antara anak-anak [6].

Untuk menyelesaikan permasalahan diatas, EksFlorasi hadir sebagai solusi dalam bentuk aplikasi edukasi berbasis *mobile* yang berfungsi untuk mendorong anak-anak untuk mengeksplorasi lingkungan alam serta mengenal flora dan fauna sekitar mereka melalui media interaktif. Aplikasi *mobile* EksFlorasi ini merupakan hasil proyek *capstone* Bangkit Academy 2023 Batch 1. Aktivitas utama dalam EksFlorasi adalah “penangkapan” hewan atau tumbuhan menggunakan kamera *smartphone* yang kemudian memberikan informasi terkait hewan atau tumbuhan yang ditangkap beserta fitur gamifikasi berupa hadiah poin dari aktivitas

“penangkapan” tersebut yang dapat dikumpulkan untuk mendapatkan peringkat pada *leaderboard* berdasarkan provinsi pengguna yang di-*refresh* setiap minggu. Maka dari itu, aplikasi EksFlorasi ini bertujuan untuk mendukung pembelajaran di luar ruangan tanpa memisahkan anak-anak dari *smartphone* mereka dengan memanfaatkan potensi *smartphone* dalam pembelajaran anak-anak.

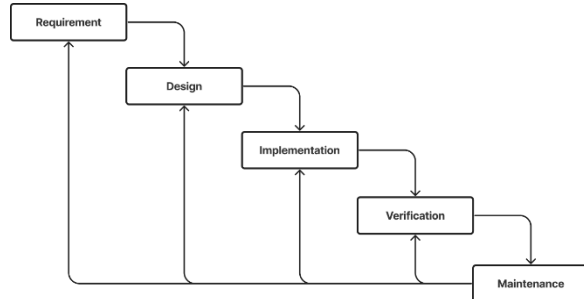
Untuk mendukung fungsionalitas aplikasi EksFlorasi, diperlukan perancangan dan pengembangan RESTful API sebagai *back-end* yang berfungsi sebagai perantara antara aplikasi *mobile* dengan *server* dengan mematuhi protokol jaringan HTTP [7], [8], [9], [10]. Lalu, diperlukan juga teknologi *cloud computing* berupa Google Cloud Platform (GCP) yang menyediakan sumber daya komputasi secara *on-demand* melalui Internet sehingga RESTful API tersebut dapat diakses oleh aplikasi *mobile* EksFlorasi tanpa hambatan di seluruh daerah Indonesia [11].

Terdapat beberapa penelitian terdahulu yang membahas tentang pengembangan RESTful API pada aplikasi *mobile* dan *deployment* API tersebut pada Google Cloud Platform. Pertama, penelitian yang dilakukan oleh [12] berfokus pada pengembangan REST API menggunakan Node.js dan Python pada Google Cloud Platform. Penelitian ini dilakukan menggunakan metode *extreme programming*. Berbeda dengan penelitian pertama, penelitian yang dilakukan oleh [13] juga bertujuan untuk mengembangkan RESTful API untuk aplikasi *mobile* AyamHub yang dikembangkan menggunakan metode *waterfall* dan menggunakan *framework* Node.js Express yang kemudian di-*deploy* pada Google Cloud Platform. Kedua penelitian tersebut menggunakan teknik pengujian yang sama untuk RESTful API yang telah dikembangkan keduanya, yaitu menggunakan metode *black box testing*. Berbeda dengan kedua penelitian sebelumnya, penelitian yang dilakukan oleh [14] berfokus pada *deployment* aplikasi Getfix pada Google Cloud Platform. Walaupun ketiga penelitian tersebut berbeda dari segi objek penelitian dan metode pengembangan, ketiganya berhasil mengembangkan RESTful API untuk sebuah aplikasi *mobile* dan kemudian men-*deploy* API tersebut pada Google Cloud Platform.

Oleh karena itu, tujuan utama dari penelitian ini adalah untuk mengembangkan RESTful API yang berfungsi sebagai komponen *back-end* aplikasi *mobile* EksFlorasi dan *deployment* RESTful API tersebut pada Google Cloud Platform. RESTful API ini dibangun menggunakan metode *waterfall* menggunakan Node.js dengan *framework* Express JS. RESTful API tersebut kemudian di-*deploy* ke lingkungan *cloud* menggunakan layanan *cloud computing* Google Cloud Platform. Dengan begitu, RESTful API yang telah dikembangkan akan dapat berfungsi dengan baik sebagai *back-end* aplikasi EksFlorasi dan dapat diakses oleh pengguna di seluruh Indonesia.

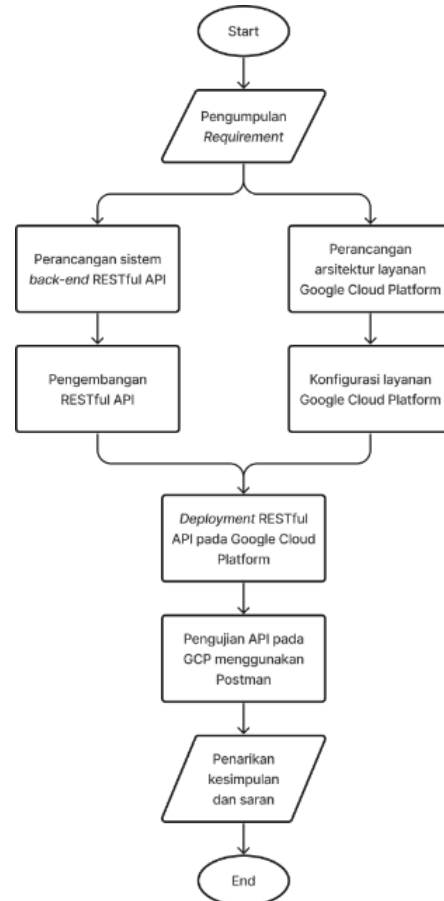
2. Metode Penelitian

Pada penelitian ini, metode penelitian dilakukan dengan menggunakan metode *waterfall* yang bersifat sistematis dan merupakan salah satu model *Software Development Lifecycle* (SDLC) yang paling sering digunakan dalam pengembangan sistem [15].



Gambar 1. Metode *Waterfall*

Berdasarkan metode *waterfall*, penelitian ini terdiri dari beberapa tahapan yang dimulai dengan pengumpulan *requirement*. Lalu, dilakukan perancangan sistem RESTful API menggunakan diagram UML yang berguna untuk memvisualisasikan keseluruhan sistem yang akan dikembangkan [16]. Bersamaan dengan perancangan RESTful API tersebut, terdapat perancangan *database* menggunakan *Entity-Relationship Diagram* (ERD) yang berfungsi untuk memetakan objek, informasi objek, serta hubungan antar objek dalam *database* [17]. Selain itu, dilakukan juga perancangan arsitektur layanan Google Cloud Platform yang digunakan. Tahap perancangan sistem kemudian dilanjutkan dengan pengembangan RESTful API yang telah dirancang beserta konfigurasi layanan Google Cloud Platform untuk persiapan tahap *deployment*. Setelah tahap sebelumnya selesai, dilakukan tahap *deployment* RESTful API pada Google Cloud Platform yang kemudian dilanjutkan dengan tahap pengujian. Tahap pengujian tersebut dilakukan dengan menggunakan metode *black box testing* yang berfokus pada kemampuan sistem dalam memberikan *output* yang sesuai dengan *input* yang diberikan sehingga membuktikan bahwa sistem telah dapat bekerja sebagaimana mestinya dan telah memenuhi kualitas fungsional yang diharapkan [18], [19]. Pada tahap ini, dilakukan dua jenis pengujian, yaitu pengujian fitur RESTful API dan pengujian arsitektur Google Cloud Platform. Pada pengujian fitur RESTful API, setiap *endpoint* fitur diuji untuk memastikan *output* yang dihasilkan telah sesuai dengan *input* yang diberikan. Lalu, pengujian arsitektur Google Cloud Platform dilakukan dengan *load testing* untuk menguji performa arsitektur Google Cloud Platform yang digunakan dengan melakukan simulasi *traffic* dengan jumlah *virtual user* yang berbeda-beda. Kemudian, tahap terakhir dari penelitian ini adalah penarikan kesimpulan dan saran yang didasari oleh hasil pengujian yang dilakukan pada tahap sebelumnya. Tahapan penelitian dapat dilihat pada Gambar 2 berikut.



Gambar 2. Tahapan Penelitian

2.1. Pengumpulan *Requirement*

Pengumpulan *requirement* dilakukan dengan mengidentifikasi fitur-fitur yang diperlukan dengan menganalisis kebutuhan aplikasi *mobile* EksFlorasi. Maka dari itu, fitur-fitur RESTful API aplikasi EksFlorasi yang perlu dikembangkan adalah sebagai berikut.

Tabel 1. Fitur-Fitur RESTful API EksFlorasi

Nama Fitur	Deskripsi	Route URL
<i>Log In</i>	Membuat <i>session</i> baru untuk <i>user</i> .	/auth/signin
<i>Sign Up</i>	Menyimpan data <i>user</i> baru di <i>database</i> dan membuat <i>session</i> baru untuk <i>user</i> .	/auth/register
Lihat Koleksi	Menampilkan data koleksi <i>user</i> dari <i>database</i> .	/collections
Lihat Detail Koleksi	Menampilkan informasi tentang flora atau fauna dalam koleksi <i>user</i> .	/collections
Tambah Koleksi	Meng- <i>upload</i> foto flora atau fauna yang difoto oleh <i>user</i> dan menyimpannya ke dalam koleksi milik <i>user</i> dan menambah poin <i>user</i> .	/collections

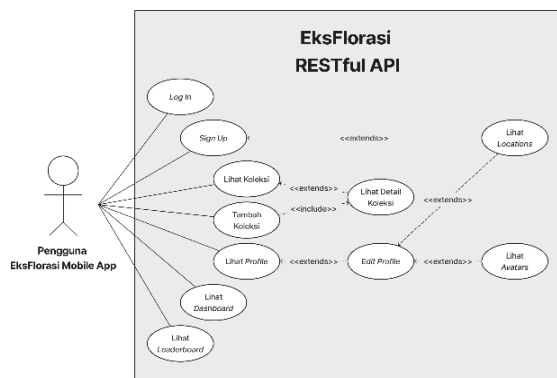
Lihat Profile	Menampilkan informasi tentang <i>user</i> , seperti <i>email</i> , lokasi, avatar, dan <i>username</i> .	/profile
Edit Profile	Memperbarui data <i>user</i> .	/profile
Lihat Dashboard	Menampilkan informasi berupa jumlah poin harian, poin mingguan, <i>streak</i> , <i>ranking</i> , dan <i>tips</i> untuk <i>user</i> .	/dashboard_information
Lihat Leaderboard	Menampilkan peringkat <i>user</i> beserta <i>user-user</i> lain di lokasi yang sama berdasarkan jumlah poin yang didapat di minggu tersebut.	/rank
Lihat Locations	Mengambil ID dan nama lokasi yang dapat dipilih oleh <i>user</i> .	/references/locations
Lihat Avatars	Mengambil ID dan link dari foto avatar yang dapat dipilih oleh <i>user</i> .	/references/avatars

2.2. Perancangan Sistem

Perancangan sistem RESTful API dilakukan menggunakan diagram UML berupa *use case diagram*, *activity diagram*, dan *sequence diagram*. Selain itu, perancangan *database* juga dilakukan dengan menggunakan *Entity-Relationship Diagram* (ERD). Lalu, perancangan layanan Google Cloud Platform yang digunakan juga dilakukan dalam bentuk rancangan arsitektur layanan Google Cloud Platform.

1. Use Case Diagram

Use case diagram berikut menggambarkan interaksi antara pengguna aplikasi EksFlorasi dengan RESTful API EksFlorasi.

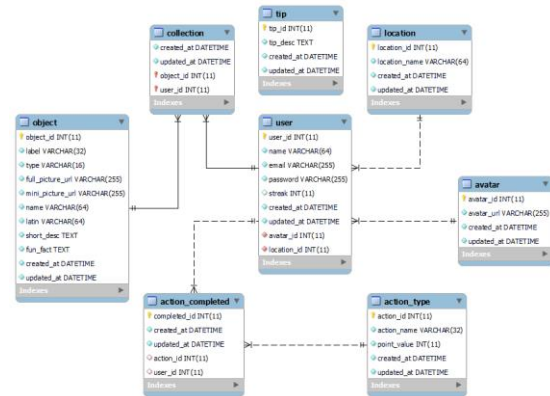


Gambar 3. Use Case Diagram RESTful API EksFlorasi

2. Entity-Relationship Diagram (ERD)

Entity-Relationship Diagram (ERD) berfungsi untuk menggambarkan struktur *database* yang digunakan

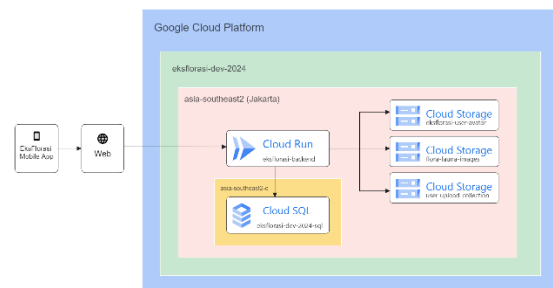
untuk menyimpan data pengguna aplikasi EksFlorasi.



Gambar 4. ERD Database RESTful API EksFlorasi

3. Arsitektur Layanan Google Cloud Platform

Agar *deployment* RESTful API EksFlorasi dapat dilakukan dengan performa yang optimal serta hemat biaya, rancangan arsitektur layanan *cloud computing* pada Google Cloud Platform tersusun dari layanan Cloud Run, Cloud SQL, dan Cloud Storage yang saling bekerja sama dan masing-masing di-*deploy* pada *region* asia-southeast2 yang merupakan lokasi terdekat dengan target pengguna aplikasi, yaitu anak-anak di Indonesia, untuk mencapai *latency* terendah. Berikut adalah rancangan arsitektur layanan Google Cloud Platform yang digunakan dalam penelitian ini.



Gambar 5. Rancangan Arsitektur Cloud RESTful API EksFlorasi

3. Hasil dan Pembahasan

3.1. Hasil Pengembangan RESTful API

Sesuai dengan fitur-fitur yang telah diidentifikasi pada tahapan pengumpulan *requirement* sebelumnya, berikut adalah fitur-fitur RESTful API EksFlorasi yang telah berhasil dikembangkan beserta tampilan hasil fitur tersebut pada Postman dan tampilan aplikasi *mobile* EksFlorasi.

1. Log In

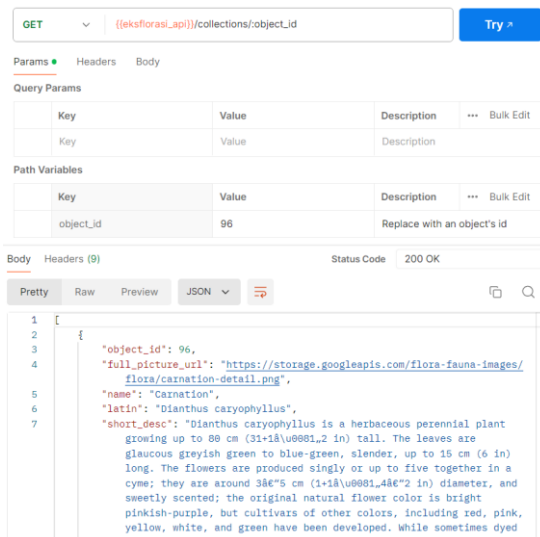
Pada fitur *Log In*, pengguna aplikasi EksFlorasi dapat mengakses akun mereka dengan memasukkan *email* dan *password* mereka yang telah didaftarkan sebelumnya. Setelah data yang di-*input* pengguna telah diperiksa oleh RESTful API dan dicocokkan



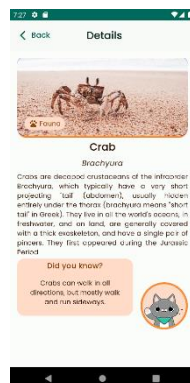
Gambar 11. Tampilan Fitur Lihat Koleksi Aplikasi EksFlorasi

4. Lihat Detail Koleksi

Pada fitur Lihat Detail Koleksi, RESTful API menghasilkan detail lengkap dari suatu flora atau fauna dalam koleksi pengguna yang dipilih. API akan mengambil data flora atau fauna dari *database* menggunakan *object_id* dalam *path variable* dari *request* dan *user_id* pengguna dari token JWT yang disertakan di bagian Header sebagai Bearer Token. Berikut adalah hasil fitur Lihat Detail Koleksi pada Postman dan *screenshot* aplikasi *mobile* EksFlorasi.



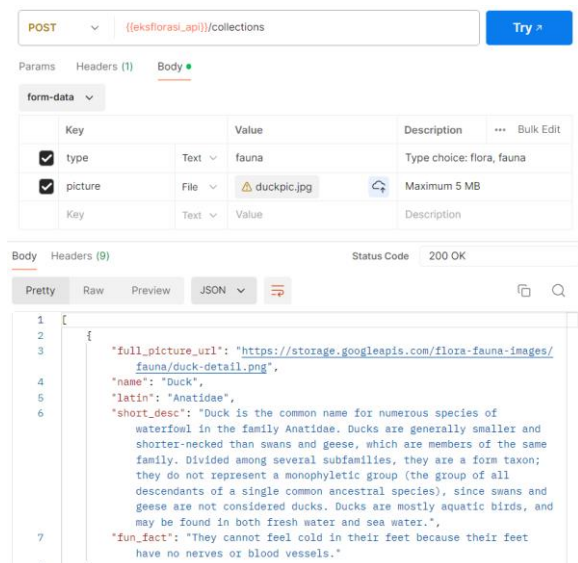
Gambar 12. Hasil Fitur Lihat Detail Koleksi



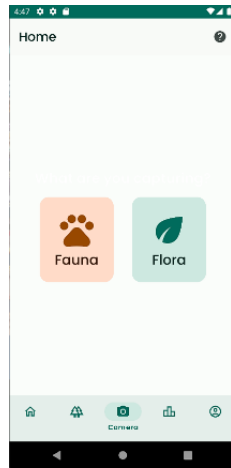
Gambar 13. Tampilan Fitur Lihat Detail Koleksi Aplikasi EksFlorasi

5. Tambah Koleksi

Pada fitur utama RESTful API aplikasi EksFlorasi, yaitu fitur Tambah Koleksi, pengguna aplikasi EksFlorasi dapat menangkap flora atau fauna dengan memfoto flora atau fauna yang ditemui dengan memasukkan tipe objek yang difoto. Setelah foto tersebut dikirimkan ke RESTful API dalam bentuk *request*, API kemudian mengunggah foto tersebut ke Google Storage dan mengirimkan Public URL foto tersebut ke model *machine learning* berdasarkan tipe objeknya. Model *machine learning* yang menggunakan metode Tensorflow.js tersebut berfungsi untuk mengklasifikasikan flora atau fauna yang difoto oleh pengguna sesuai dengan *dataset* yang dimilikinya. Setelah objek berhasil diidentifikasi, model *machine learning* akan mengirimkan label nama flora atau fauna yang difoto oleh pengguna ke RESTful API. API kemudian mengambil data flora atau fauna tersebut dari *database* dan memberikan poin kepada pengguna serta menyimpan objek ke dalam koleksi jika sebelumnya belum pernah ada dalam koleksi pengguna. API kemudian akan mengirimkan *response* berupa informasi detail dari objek yang diunggah. Karena fitur ini hanya dapat digunakan oleh pengguna aplikasi EksFlorasi yang telah terotentikasi, token JWT disertakan dalam bagian *Header* sebagai Bearer Token. Fitur Tambah Koleksi pada tampilan aplikasi EksFlorasi terbagi menjadi beberapa bagian, yaitu tampilan halaman pemilihan tipe objek, tampilan halaman kamera untuk menangkap (memfoto) objek, dan kemudian tampilan halaman detail koleksi yang baru ditangkap. Berikut adalah hasil fitur Tambah Koleksi pada Postman dan *screenshot* aplikasi *mobile* EksFlorasi.



Gambar 14. Hasil Fitur Tambah Koleksi



Gambar 15. Tampilan Pemilihan Tipe Objek Aplikasi EksFlorasi



Gambar 16. Tampilan Kamera Aplikasi EksFlorasi

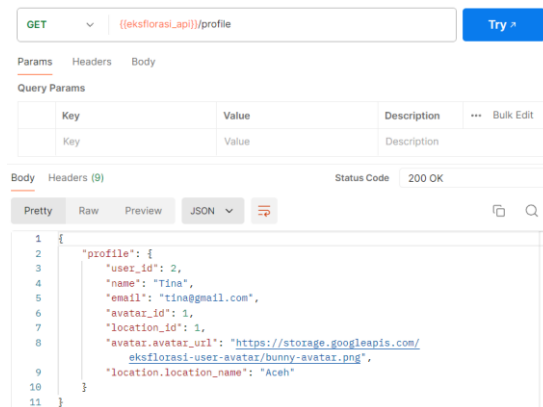


Gambar 17. Tampilan Hasil Tambah Koleksi Aplikasi EksFlorasi

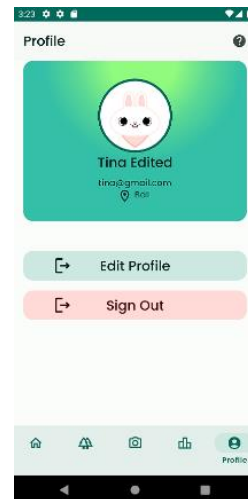
6. Lihat Profile

Pada fitur Lihat Profile, RESTful API menghasilkan informasi *profile* pengguna berdasarkan *user_id* pengguna dalam token JWT yang disertakan dalam bagian *Header* sebagai *Bearer Token*. API akan mengambil data pengguna dari *database* berupa *user_id*, nama, *email*, *avatar_id*, *location_id*, URL

foto avatar pengguna, dan nama lokasi provinsi pengguna. Berikut adalah hasil fitur Lihat Profile pada Postman dan *screenshot* aplikasi *mobile* EksFlorasi.



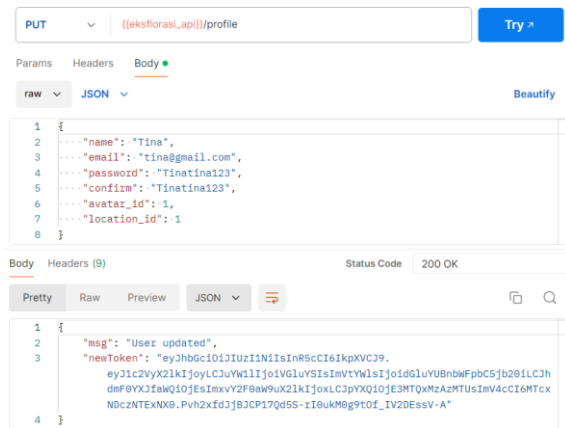
Gambar 18. Hasil Fitur Lihat Profile



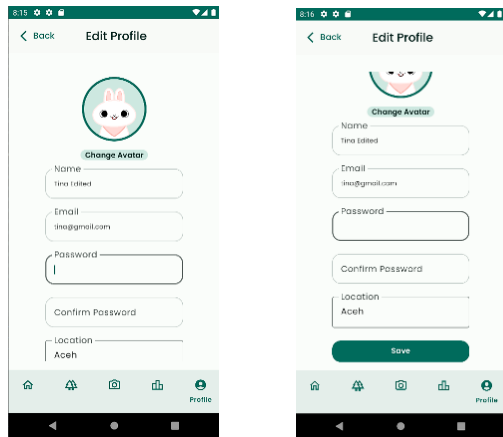
Gambar 19. Tampilan Fitur Lihat Profile Aplikasi EksFlorasi

7. Edit Profile

Pada fitur *Edit Profile*, pengguna dapat mengubah informasi *profile* mereka dengan memasukkan nama, *email*, *password*, *confirm password*, *avatar_id*, dan *location_id* baru. Jika pengguna memasukkan *email* baru, API akan memeriksa jika *email* baru pengguna sudah digunakan oleh pengguna lain. Selain itu, jika pengguna memasukkan *password* baru, API juga akan memeriksa validitas *password* baru tersebut. Setelah pemeriksaan selesai, data pengguna dalam *database* akan disimpan dan API akan menghasilkan token JWT baru untuk menandakan bahwa data pengguna telah berhasil di-*update*. Karena fitur ini hanya dapat digunakan oleh pengguna yang telah terotentikasi, token JWT juga perlu disertakan dalam bagian *Header* sebagai *Bearer Token*. Berikut adalah hasil fitur *Edit Profile* pada Postman dan *screenshot* aplikasi *mobile* EksFlorasi.



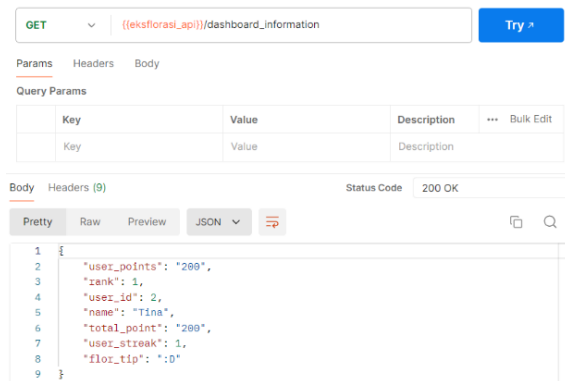
Gambar 20. Hasil Fitur *Edit Profile*



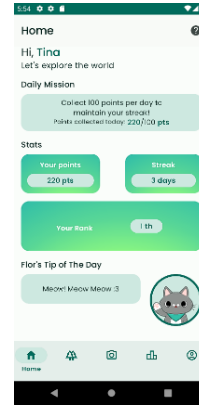
Gambar 21. Tampilan Fitur *Edit Profile* Aplikasi EksFlorasi

8. Lihat *Dashboard*

Pada fitur Lihat *Dashboard*, RESTful API menghasilkan informasi *dashboard* pengguna berdasarkan *user_id* pengguna dalam token JWT yang disertakan dalam bagian *Header* sebagai *Bearer Token*. API akan mengakses data pengguna dalam *database* dan memproses serta menampilkan data berupa jumlah poin yang sudah dikumpulkan hari ini, peringkat user dalam *leaderboard*, *user_id*, nama, total poin yang telah dikumpulkan dalam satu minggu, *streak*, dan *tips* dari Flor (maskot aplikasi EksFlorasi). Berikut adalah hasil fitur Lihat *Dashboard* pada Postman dan *screenshot* aplikasi *mobile* EksFlorasi.



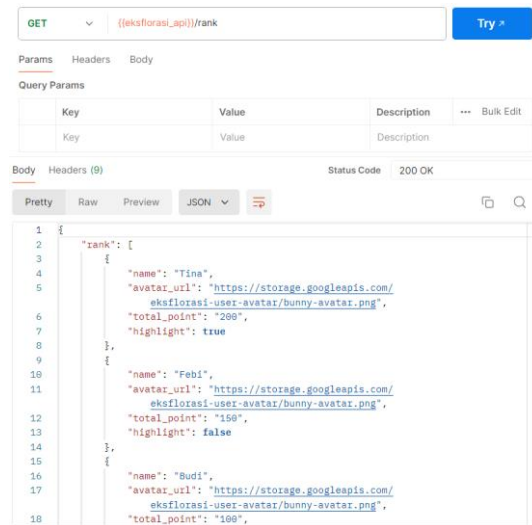
Gambar 22. Hasil Fitur Lihat *Dashboard*



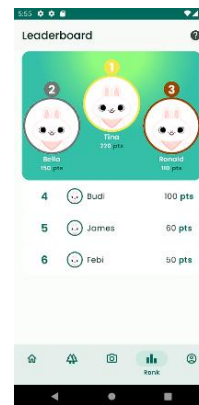
Gambar 23. Tampilan Fitur Lihat *Dashboard* Aplikasi EksFlorasi

9. Lihat *Leaderboard*

Pada fitur Lihat *Leaderboard*, RESTful API menghasilkan peringkat *leaderboard* dari provinsi yang sama dengan *highlight* pada nama pengguna. *Highlight* pada pengguna tersebut berdasarkan *user_id* pengguna dalam token JWT yang disertakan dalam bagian *Header* sebagai *Bearer Token*. Berikut adalah hasil fitur Lihat *Leaderboard* pada Postman dan *screenshot* aplikasi *mobile* EksFlorasi.



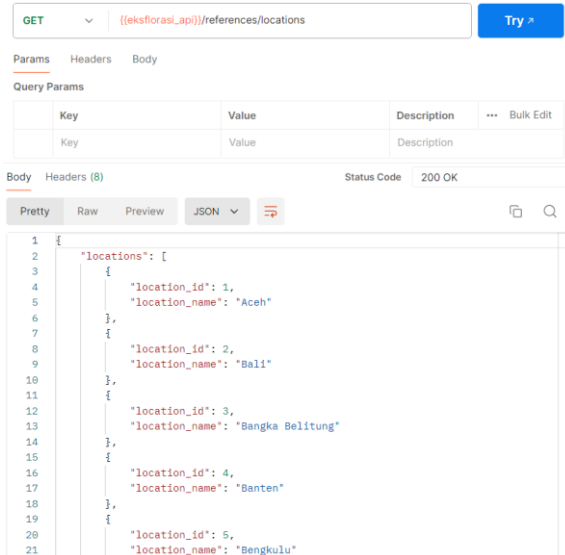
Gambar 24. Hasil Fitur Lihat *Leaderboard*



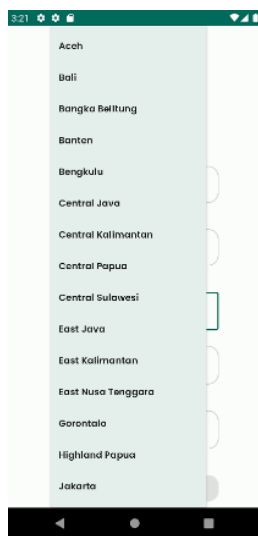
Gambar 25. Tampilan Fitur Lihat *Leaderboard* Aplikasi EksFlorasi

10. Lihat *Locations*

Pada fitur Lihat *Locations*, RESTful API menghasilkan *list* provinsi Indonesia dari *database* yang dapat dipilih pengguna pada saat proses fitur *Sign Up* dan *Edit Profile*. Provinsi pengguna tersebut berfungsi untuk mengelompokkan pengguna berdasarkan provinsi untuk *leaderboard*. Berikut adalah hasil fitur Lihat *Locations* pada Postman dan *screenshot* aplikasi *mobile* EksFlorasi.



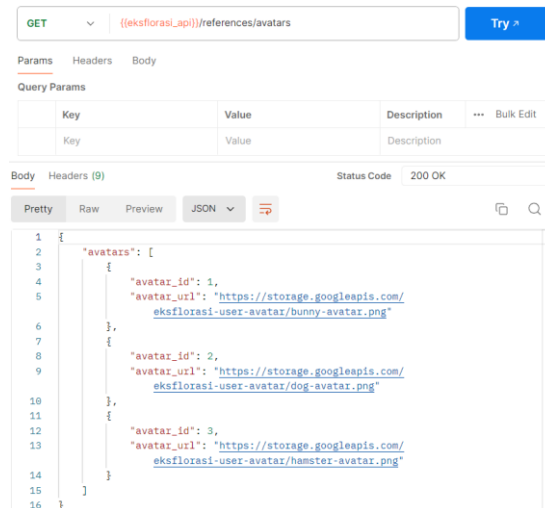
Gambar 26. Hasil Fitur Lihat *Locations*



Gambar 27. Tampilan Fitur Lihat *Locations* Aplikasi EksFlorasi

11. Lihat *Avatars*

Pada fitur Lihat *Avatars*, RESTful API menghasilkan *list* *avatar_id* dan URL foto avatar yang disimpan dalam Cloud Storage dari *database*. Pilihan foto avatar tersebut dapat diakses dan dipilih oleh pengguna saat proses *Edit Profile* sebagai gambar *profile* pengguna. Berikut adalah hasil fitur Lihat *Avatars* pada Postman dan *screenshot* aplikasi *mobile* EksFlorasi.



Gambar 28. Hasil Fitur Lihat *Avatars*



Gambar 29. Tampilan Fitur Lihat *Avatars* Aplikasi EksFlorasi

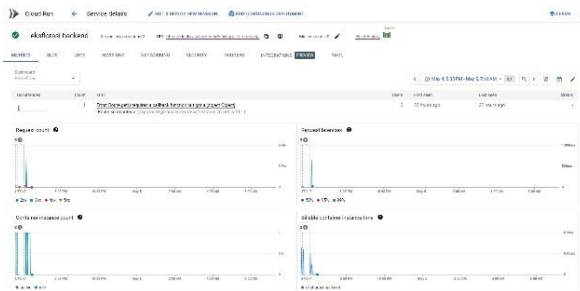
3.2. Hasil *Deployment* RESTful API pada GCP

Sesuai dengan rancangan arsitektur layanan Google Cloud Platform sebelumnya, berikut adalah konfigurasi masing-masing layanan Google Cloud yang digunakan dalam *deployment* RESTful API EksFlorasi.

1. Cloud Run

Service Cloud Run yang digunakan untuk menyajikan RESTful API bernama eksflorasi-backend ini berlokasi di region asia-southeast2 (Jakarta). Konfigurasi *service* ini memiliki *concurrency* sebesar 50 dan terdiri dari satu vCPU dan RAM sebesar 4GiB. Dengan fitur *autoscaling* yang merupakan fitur unggulan Cloud Run, *service* eksflorasi-backend ini dapat dengan mudah menambah hingga lima *instance* tambahan jika diperlukan. Ditambah lagi, *service* eksflorasi-backend ini juga memiliki fitur pengalokasian CPU hanya saat request *processing* sehingga dapat menghemat biaya. Agar RESTful API dapat berinteraksi dengan *database*, *Service* eksflorasi-

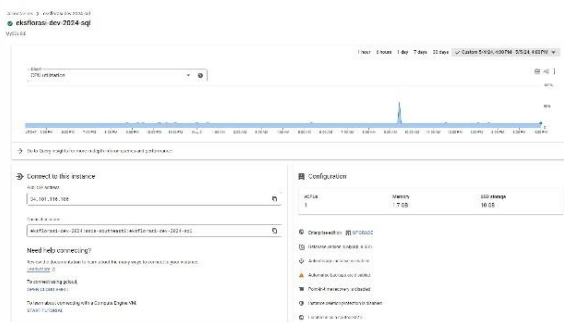
backend ini terhubung dengan *instance* Cloud SQL eksplorasi-dev-2024-sql yang berfungsi sebagai *server database* MySQL yang diperlukan dalam proses kerja RESTful API.



Gambar 30. Dashboard Service Cloud Run

2. Cloud SQL

Sebagai *database server* yang berfungsi menyimpan seluruh data yang diperlukan untuk memastikan RESTful API aplikasi EksFlorasi dapat bekerja dengan baik, konfigurasi *instance* Cloud SQL yang bernama eksplorasi-dev-2024-sql ini terdiri dari satu vCPU dengan kapasitas memori 1.7GB dan SSD sebesar 10 GB. *Instance* ini juga berlokasi di *zone* asia-southeast2-c yang berada di dalam *region* asia-southeast2.



Gambar 31. Instance Cloud SQL

3. Cloud Storage

Berfungsi untuk mendukung RESTful API dengan menyimpan seluruh foto atau gambar yang diperlukan dalam aplikasi *mobile* EksFlorasi, tiga *bucket* telah berhasil dibuat di lokasi *region* asia-southeast2 (Jakarta) dan memiliki *default object class* Standard.

Name	Location type	Location	Default storage class
eksplorasi-user-avatar	Region	asia-southeast2	Standard
flora-fauna-images	Region	asia-southeast2	Standard
user-upload-collection	Region	asia-southeast2	Standard

Gambar 32. Bucket-Bucket Cloud Storage

Walaupun memiliki konfigurasi yang sama, setiap *bucket* memiliki fungsi yang berbeda. *Bucket* eksplorasi-user-avatar berfungsi untuk menyimpan gambar-gambar avatar yang dapat diakses melalui Public URL yang disimpan dalam *database*. Lalu,

bucket flora-fauna-images berguna untuk menyimpan semua foto flora dan fauna yang diperlukan dalam aplikasi EksFlorasi. Sama halnya dengan *bucket* sebelumnya, semua foto flora dan fauna pada *bucket* ini dapat diakses melalui Public URL yang disimpan dalam *database*. *Bucket* yang terakhir adalah *bucket* user-upload-collection yang berfungsi untuk menyimpan foto-foto yang diunggah oleh pengguna untuk kemudian diproses oleh model *machine learning*. Berbeda dengan *bucket* sebelumnya, *bucket* ini memiliki *lifecycle rule* untuk menghapus *object* di dalamnya yang telah berusia satu hari. *Lifecycle rule* ini diberlakukan karena foto yang di-upload *user* hanya diperlukan untuk pemrosesan sehingga hanya akan disimpan sebentar untuk menghemat biaya.

3.3. Hasil Pengujian RESTful API

Setelah pengembangan RESTful API dan *deployment* API tersebut pada Google Cloud Platform, dilakukan *black box testing* untuk memastikan bahwa RESTful API yang telah dikembangkan dapat bekerja dengan baik. Pertama-tama, dilakukan pengujian pada fungsionalitas RESTful API sebagai berikut.

Tabel 2. Hasil Pengujian RESTful API

Kondisi	Hasil	Status
Log In		
Log In berhasil	API memberikan respons "Logged in successfully" dengan token JWT pengguna.	Terpenuhi
Email tidak ada dalam database	API memberikan respons error "Email is incorrect".	Terpenuhi
Password yang dimasukkan salah	API memberikan respons error "Email or Password is incorrect".	Terpenuhi
Sign Up		
Sign Up berhasil.	API memberikan respons "Account successfully created" dengan token JWT pengguna.	Terpenuhi
Satu atau lebih field tidak terisi.	API memberikan respons error "All fields should not be empty".	Terpenuhi
Email sudah digunakan oleh user lain.	API memberikan respons error "Email is already in use!".	Terpenuhi
Password dan Confirm Password tidak sama.	API memberikan respons error "Password and Confirm Password should match".	Terpenuhi
Password invalid.	API memberikan respons error "Password should contain at least 8 characters and include 1 lowercase letter, 1 uppercase letter, and 1 number".	Terpenuhi
Lihat Koleksi		
Lihat Koleksi berhasil	API memberikan respons berupa list koleksi flora atau fauna pengguna.	Terpenuhi

Koleksi pengguna kosong.	API memberikan respons "There's nothing here yet...".	Terpenuhi
Lihat Detail Koleksi		
Lihat Detail Koleksi berhasil.	API memberikan respons berupa detail koleksi berdasarkan object_id yang diminta pengguna.	Terpenuhi
Tambah Koleksi		
Tambah Koleksi berhasil	API menambah poin dan streak pengguna serta memberikan respons berupa detail koleksi yang ditambahkan.	Terpenuhi
Tidak ada foto yang di-upload.	API memberikan respons error "No image file provided".	Terpenuhi
Objek yang di-upload tidak dapat diidentifikasi.	API memberikan respons error "Something went wrong, please try again later".	Terpenuhi
Lihat Profile		
Lihat Profile berhasil.	API memberikan respons berupa informasi profile pengguna.	Terpenuhi
Edit Profile		
Edit Profile berhasil.	API memberikan respons "User updated" dengan token JWT baru pengguna.	Terpenuhi
Satu atau lebih field tidak terisi.	API memberikan respons error "All fields should not be empty".	Terpenuhi
Email sudah digunakan oleh user lain.	API memberikan respons error "This email is unavailable, please insert another email".	Terpenuhi
Password dan Confirm Password tidak sama.	API memberikan respons error "Password and Confirm Password should match".	Terpenuhi
Password invalid.	API memberikan respons error "Password should contain at least 8 characters and include 1 lowercase letter, 1 uppercase letter, and 1 number".	Terpenuhi
Lihat Dashboard		
Lihat Dashboard berhasil.	API memberikan respons berupa informasi dashboard pengguna.	Terpenuhi
Lihat Leaderboard		
Lihat Leaderboard berhasil.	API memberikan respons berupa peringkat dalam satu lokasi dengan pengguna.	Terpenuhi
Lihat Locations		
Lihat Locations berhasil.	API memberikan respons berupa list provinsi Indonesia.	Terpenuhi
Lihat Avatars		
Lihat Avatars berhasil.	API memberikan respons berupa list URL foto avatar.	Terpenuhi

Berdasarkan hasil pengujian *black box* diatas, dapat disimpulkan bahwa RESTful API EksFlorasi yang telah dikembangkan sudah sesuai dengan kebutuhan aplikasi EksFlorasi.

3.4. Hasil Pengujian Arsitektur Google Cloud Platform

Untuk mengukur performa arsitektur Google Cloud Platform yang digunakan dalam *deployment* RESTful API EksFlorasi, dilakukan pengujian dengan metode *black box testing* dalam bentuk *load testing* untuk menguji performa layanan Google Cloud yang digunakan dengan melakukan simulasi *traffic* sebesar 20, 50, 70, dan 100 *virtual user* (VU) yang melakukan *request* terus menerus selama lima menit. Berikut adalah hasil *load testing* pada RESTful API yang telah di-deploy pada Google Cloud Platform.

Tabel 3. Hasil *Load Test* RESTful API pada GCP

Virtual User	Total Request Sent	Throughput (req/s)	Avg. Response Time	Error Rate
20	5.833	18,86	837 ms	0,19%
50	7.610	24,49	1.729 ms	0,51%
70	8.139	26,27	2.366 ms	6,57%
100	10.489	33,97	2.636 ms	27,86%

Berdasarkan hasil *load testing* yang telah dilakukan, dapat disimpulkan bahwa pengujian menggunakan 50 VU menghasilkan performa yang paling optimal daripada pengujian dengan jumlah *virtual user* lain. Pada pengujian 50 VU selama lima menit, API berhasil memproses 7.610 *request* dengan rata-rata 24,49 *request* setiap detik dan menghasilkan nilai *error rate* yang cukup rendah, yaitu 0,51%. Hasil ini didukung oleh penelitian yang dilakukan oleh [22], yang menyatakan bahwa *error rate* dibawah 1% menunjukkan aplikasi yang dibangun dapat berjalan dengan baik.

Walaupun *average response time* yang diperlukan RESTful API lebih tinggi daripada 1.000 ms (1 detik), hal ini tidaklah mengherankan karena terdapat beberapa fitur yang memerlukan sumber daya dan waktu lebih dengan *load* sebesar 50 VU. Sementara itu, pada pengujian 70 dan 100 VU, angka *average response time* melebihi 2 detik dan *error rate* melebihi 1% menunjukkan bahwa jumlah VU tersebut terlalu tinggi untuk menghasilkan performa yang optimal. Selain itu, meskipun performa RESTful API dengan 20 VU memiliki nilai yang lebih baik dalam segi *average response time* dan *error rate*, performa RESTful API dengan 50 VU dianggap lebih optimal karena jumlah VU yang dapat ditampung dan jumlah *request* yang dapat diproses lebih tinggi dengan nilai *average response time* dan *error rate* yang cukup rendah untuk *load* tersebut.

4. Kesimpulan

Berdasarkan hasil penelitian berupa RESTful API dan *deployment* RESTful API tersebut pada Google Cloud Platform, dapat disimpulkan bahwa RESTful API untuk aplikasi *mobile* EksFlorasi telah berhasil dirancang dan dibangun menggunakan Node.js dan *framework* Express JS. *Deployment* RESTful API tersebut pada Google Cloud Platform juga telah dapat mendukung performa RESTful API aplikasi EksFlorasi di *cloud* dengan *latency* terendah di seluruh Indonesia. Dengan pengujian *black box*, RESTful API dan *deployment*-nya pada Google Cloud telah terbukti dapat berfungsi dengan baik dan memberikan performa optimal pada 50 *user* secara bersamaan dengan tingkat *error* rendah.

Untuk penelitian selanjutnya, disarankan agar melakukan optimisasi pada RESTful API yang dikembangkan agar API tersebut lebih ringan sehingga memerlukan sumber daya yang lebih sedikit. Dengan begitu, API akan lebih hemat dalam penggunaan sumber daya dan sumber daya tersebut dapat dialokasikan kepada bagian-bagian yang lebih penting, seperti pemrosesan foto flora dan fauna. Selain itu, layanan Google Cloud juga disarankan menggunakan alokasi sumber daya yang lebih besar agar dapat menopang beban lebih seperti jumlah *user* yang lebih banyak tanpa mengalami peningkatan tingkat *error*.

Ucapan Terimakasih

Penulis mengucapkan terima kasih kepada Mohammad Azri Harahap dan Sholeh Rodhi Putra Siswanto yang telah bekerja sama dengan penulis di kelompok C23-PR499 dalam proyek *capstone* Bangkit 2023 Batch 1.

SUMBER RUJUKAN

Referensi

- [1] Badan Pusat Statistik, "Statistik Telekomunikasi Indonesia 2022," Badan Pusat Statistik, Jakarta, Aug. 2023.
- [2] K. F. Ne'matullah, N. A. Talib, R. W. M. Mee, L. S. Pek, S. Amiruddin, and M. R. Ismail, "The impact of outdoor play on children's well-being: A scoping review," *Masyarakat, Kebudayaan dan Politik*, vol. 35, no. 3, pp. 282–296, Aug. 2022, doi: 10.20473/mkp.v35i32022.282-296.
- [3] N. Aninsi, "Persebaran Flora dan Fauna di Indonesia." Accessed: Jul. 09, 2023. [Online]. Available: <https://katadata.co.id/safrezi/berita/61cc2ed801659/persebaran-flora-dan-fauna-di-indonesia>
- [4] S. Kawas, N. S. Kuhn, M. Tari, A. Hiniker, and K. Davis, "Otter this world: Can a mobile application promote children's connectedness to nature," in *Proceedings of the Interaction Design and Children Conference, IDC 2020*, Association for Computing Machinery, Inc, Jun. 2020, pp. 444–457. doi: 10.1145/3392063.3394434.
- [5] S. A. Abdullah *et al.*, "Determining Elements in Mobile Learning Implementation among Instructors in Vocational Colleges: A Fuzzy Delphi Method," *IEEE Access*, vol. 9, pp. 150839–150845, 2021, doi: 10.1109/ACCESS.2021.3121703.

- [6] D. Patni Ninghardjanti, Mp. Chairul Huda Atma Dirgatama, Mp. Arif Wahyu Wirawan, and Mp. C. Penerbit Pena Persada, *BUKU BERBASIS RISET: PEMBELAJARAN MULTIMEDIA BERBASIS MOBILE LEARNING*. 2020.
- [7] R. Setiawan, "Apa itu REST Web Service: Tutorial Back-End Developer." Accessed: Jul. 09, 2023. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-rest-web-service/>
- [8] IBM, "What is a REST API?" Accessed: Jul. 09, 2023. [Online]. Available: <https://www.ibm.com/topics/rest-apis#:~:text=the%20next%20step-,What%20is%20a%20REST%20API%3F,representational%20state%20transfer%20architectural%20style>.
- [9] Codecademy, "What is REST?" Accessed: Jul. 09, 2023. [Online]. Available: <https://www.codecademy.com/article/what-is-rest>
- [10] N. Huda, "Pengertian Web Service, Cara Kerja, dan Fungsinya." Accessed: Jul. 09, 2023. [Online]. Available: <https://www.dewaweb.com/blog/apa-itu-web-service/>
- [11] P. Vergadia, *Visualizing Google Cloud: 101 Illustrated References for Cloud Engineers and Architects*. Wiley, 2022.
- [12] R. F. Falah, M. Komarudin, and M. Pratama, "Perancangan Microservice Berbasis REST API pada Google Cloud Platform Menggunakan Nodejs dan Python," *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, vol. 11, no. 3, pp. 1103–1112, Sep. 2023, doi: <http://dx.doi.org/10.23960/jitet.v11i3%20s1.3506>.
- [13] E. Nurhayati and Agussalim, "Rancang Bangun Back-end API pada Aplikasi Mobile AyamHub Menggunakan Framework Node JS Express," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, vol. 11, no. 3, pp. 524–531, Jul. 2023, doi: 10.26418/justin.v11i3.66823.
- [14] S. Febriani and F. Purwaningtias, "Implementasi Platform As A Service (PAAS) Pada Aplikasi Getfix Berbasis Cloud Computing," *Jurnal Sains dan Informatika*, vol. 8, no. 2, pp. 86–95, Nov. 2022, doi: 10.22216/jsi.v8i2.1653.
- [15] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, Oct. 2020.
- [16] R. Miles and K. Hamilton, *Learning UML 2.0: A Pragmatic Introduction to UML*. O'Reilly Media, 2006.
- [17] S. M. Pulungan, R. Febrianti, T. Lestari, N. Gurning, and N. Fitriana, "Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database," *Jurnal Ekonomi Manajemen Dan Bisnis (JEMB)*, vol. 2, no. 1, pp. 98–102, Feb. 2023, doi: 10.47233/jemb.v2i1.533.
- [18] A. Arifita Arwaz, K. Putra, R. Putra, T. Kusumawijaya, and A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 2, no. 4, pp. 130–134, Oct. 2019.
- [19] R. Parlita, T. A. Nisaa', S. M. Ningrum, and B. A. Haque, "LITERATURE STUDY OF THE LACK AND EXCESS OF TESTING THE BLACK BOX," *TEKNOMATIKA*, vol. 10, no. 02, pp. 131–140, Sep. 2020.
- [20] A. Rahmatulloh, H. Sulastrri, and R. Nugroho, "Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512," 2018.
- [21] R. Poddar, "What is a JWT? Understanding JSON Web Tokens." Accessed: Oct. 18, 2023. [Online]. Available: <https://supertokens.com/blog/what-is-jwt>
- [22] D. Madhani, E. Darwiyanto, and A. Gandhi, "Performance Testing Menggunakan Metode Load Testing dan Stress Testing pada Sistem Core Banking PT. XYZ," *e-Proceeding of Engineering*, vol. 10, no. 6, pp. 5431–5441, Dec. 2023.