

ANALISA DAN IMPLEMENTASI *WEB SERVICE* MENGGUNAKAN METODE *RESTFUL API* PADA APLIKASI PEMINJAMAN ASET

Hendrik Gunawan^{1*}, Hari Soetanto²

^{1,2}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, DKI Jakarta, Indonesia

Email: ^{1*}hendrik44711@email.com, ²hari.soetanto@budiluhur.ac.id
(* : corresponding author)

Abstrak- PT. Prestasi Retail Innovation yang berlokasi di Jl. Daan Mogot No. 151, Jakarta Barat merupakan perusahaan yang bergerak di bidang retail Apparel olahraga dan memiliki sekitar 500-700 karyawan. Dengan jumlah aset atau peralatan inventaris yang banyak, PT. Prestasi Retail Innovation memiliki tantangan dalam mengelola dan memantau aset-aset mereka secara efektif. Aset tersebut bisa dipakai bersama, sebagai simpanan cadangan ataupun dipakai oleh karyawan tertentu sebagai kebutuhan operasional. Selain itu, Pelaporan masalah secara manual seringkali menghadapi berbagai tantangan, seperti waktu yang dibutuhkan untuk melaporkan masalah, kemungkinan kesalahan manusia, sulitnya memantau status penyelesaian masalah, dan lain sebagainya. *Web service* memungkinkan sebuah sistem dapat diakses dengan mudah dan cepat karena penggunaanya dapat mengakses melalui ponsel yang terhubung ke internet serta pencatatan data yang lebih optimal karena tersimpan di database. Maka PT. Prestasi Retail Innovation perlu sebuah *web service* untuk manajemen aset dan pelaporan masalah dapat membantu perusahaan untuk tetap *up-to-date* dengan teknologi terbaru dan memaksimalkan potensi aset-aset mereka. *Web service* juga di perlukan untuk kebutuhan retail yang dimana pergerakan *mobile* sangat memungkinkan sebuah sistem dapat diakses dengan mudah, cepat dan bisa di akses dimana saja perusahaan dapat dengan mudah melacak aset-aset tersebut, memperbaharui informasi aset secara *real-time*, serta meningkatkan efisiensi dan produktivitas dalam penanganan masalah. Hasil dari penelitian ini menunjukkan bahwa teknologi *web service* yang diimplementasikan dapat digunakan oleh karyawan maupun admin, seperti aplikasi *mobile* dan aplikasi *web*. Hal ini dapat membantu meningkatkan produktivitas bagi para karyawan untuk membuat laporan masalah maupun pengajuan peminjaman aset. Dengan adanya enkripsi yang diterapkan, maka data perusahaan menjadi lebih aman dan terhadap penyalahgunaan oleh pihak yang tidak bertanggung jawab.

Kata Kunci: *Web Service*, Android, REST, Algoritme, Peminjaman Aset

WEB SERVICE ANALYSIS AND IMPLEMENTATION USING THE RESTFUL ALGORITME METHOD IN ASSET LENDING APPLICATIONS

Abstract- PT. Prestasi Retail Innovation which is located on Jl. Daan Mogot No. 151, West Jakarta is a company engaged in the sports apparel retail sector and has around 500-700 employees. With a large number of assets or inventory equipment, PT. Prestasi Retail Innovation has challenges in managing and monitoring their assets effectively. These assets can be used together, as reserve deposits or used by certain employees as operational needs. In addition, manual problem reporting often faces various challenges, such as the time it takes to report problems, the possibility of human error, the difficulty of monitoring problem resolution status, and so on. *Web services* allow a system to be accessed easily and quickly because users can access it via a cell phone connected to the internet and more optimal data recording because it is stored in a database. Then PT. Prestasi Retail Innovation needs a *web service* for asset management and problem reporting to help companies stay *up-to-date* with the latest technology and maximize the potential of their assets. *Web services* are also needed for retail needs where *mobile* movement allows a system to be accessed easily, quickly and can be accessed anywhere companies can easily track these assets, update asset information in *real-time*, and increase efficiency and productivity in handling problems. The results of this study indicate that the implemented *web service* technology can be used by employees and admins, such as *mobile* applications and *web* applications. This can help increase productivity for employees to make problem reports and apply for loan assets. With the encryption applied, company data becomes more secure and against misuse by irresponsible parties.

Keywords: *Web Service*, Android, REST, ALGORITME, Problem Reporting, Asset Loan.

1. PENDAHULUAN

Perkembangan teknologi informasi dan internet yang cukup pesat membuat banyak perubahan kebiasaan manusia pada kehidupan sehari-hari. Segala macam jenis teknologi dapat diakses dengan mudah hanya melalui genggaman tangan tanpa batasan ruang, waktu dan tempat. Kehadiran perangkat pintar dan internet kecepatan tinggi membuat segala hal mungkin terjadi. Jika dahulu manusia harus bergantung pada komputer yang harus terhubung dengan listrik dan tidak mudah dibawa kemana-mana untuk bekerja, maka saat ini laptop dan *Smartphone* dapat dimanfaatkan untuk menggantikan penggunaan komputer dan lebih mempermudah pekerjaan.

Hal tersebut dikarenakan akses yang mudah, cepat dan dapat dilakukan dimanapun sehingga membuat para pekerja dapat bekerja dari manapun tanpa batasan waktu dan tempat.

Dalam era *digital* seperti saat ini, penggunaan *web service* semakin meningkat. *Web service* adalah teknologi yang memungkinkan aplikasi dan sistem berbeda untuk saling berinteraksi dan berkomunikasi melalui jaringan internet. Penggunaan *web service* menjadi semakin penting sebagai alat untuk memfasilitasi pertukaran data dan informasi antar aplikasi dan sistem yang berbeda, sehingga memungkinkan bisnis untuk beroperasi secara lebih efisien dan efektif [1]. *Representational State Transfer (REST)* adalah sebuah gaya arsitektur untuk implementasi *web service* yang mengadopsi konsep perpindahan antara state. *REST* menjadi populer karena kesederhanaannya dalam menciptakan standar, kerangka kerja, dan teknologi baru. Salah satu keuntungan utama *REST* adalah penggunaan interaksi berbasis *Hypertext Transfer Protocol (HTTP)* yang sudah umum digunakan di internet [2]. Contohnya, dalam interaksi berbasis *REST*, semua komunikasi status menggunakan kode status *HTTP* standar. Misalnya, kode 404 menunjukkan bahwa sumber daya yang diminta tidak ditemukan, kode 401 menandakan permintaan tidak diotorisasi, kode 200 menunjukkan bahwa semuanya berjalan dengan baik, dan kode 500 menandakan adanya kesalahan aplikasi yang tidak dapat dipulihkan di *server*.

PT. Prestasi Retail Innovation yang berlokasi di Jl. Daan Mogot No. 151, Jakarta Barat merupakan perusahaan yang bergerak di bidang retail Apparel olahraga dan memiliki sekitar 500-700 karyawan. Dengan jumlah aset atau peralatan inventaris yang banyak, PT. Prestasi Retail Innovation memiliki tantangan dalam mengelola dan memantau aset-aset mereka secara efektif. Aset tersebut bisa dipakai bersama, sebagai simpanan cadangan ataupun dipakai oleh karyawan tertentu sebagai kebutuhan operasional. Selain itu, pelaporan masalah secara manual seringkali menghadapi berbagai tantangan, seperti waktu yang dibutuhkan untuk melaporkan masalah, kemungkinan kesalahan manusia, sulitnya memantau status penyelesaian masalah, dan lain sebagainya.

Dengan kemajuan teknologi, semakin banyak perusahaan yang beralih ke sistem manajemen aset digital untuk meningkatkan efisiensi dan produktivitas. Maka PT. Prestasi Retail Innovation perlu sebuah *web service* untuk manajemen aset dan pelaporan masalah dapat membantu perusahaan untuk tetap *up-to-date* dengan teknologi terbaru dan memaksimalkan potensi aset-aset mereka [3]. Perusahaan dapat dengan mudah melacak aset-aset tersebut, memperbaharui informasi aset secara real-time, serta meningkatkan efisiensi dan produktivitas dalam penanganan masalah. *Web service* harus bersifat *stateless*, sehingga dapat dibaca dan diakses dari berbagai platform. Arsitektur *web service* dapat dibangun menggunakan *Simple Object Access Protocol (SOAP)* atau *Representational State Transfer (REST)*, dan layanan tersebut direpresentasikan dalam format teks seperti *JSON* atau *XML* [4]. Penelitian ini memiliki kualitas yang lebih tinggi dan dapat meminimalkan risiko kesalahan. Hal ini disebabkan oleh penerapan metode *Waterfall* yang menerapkan pendekatan satu per satu dan bertahap dalam pengembangannya, sehingga lebih memperhatikan detail pekerjaan dibandingkan metode lain [5]. Berdasarkan studi literatur yang digunakan, terdapat perbedaan penelitian sebelumnya yang telah menggunakan metode Agile, atau bahkan beberapa penelitian yang tidak menggunakan metode tertentu dalam pengembangannya. Penelitian ini memiliki kualitas yang lebih tinggi dan dapat meminimalkan risiko kesalahan. Hal ini disebabkan oleh penerapan metode *Waterfall* yang menerapkan pendekatan satu per satu dan bertahap dalam pengembangannya, sehingga lebih memperhatikan detail pekerjaan dibandingkan metode lain. Model *waterfall*, yang diperkenalkan oleh Winston Royce sekitar tahun 1970, masih tetap menjadi salah satu model yang banyak digunakan dalam Rekayasa Perangkat Lunak (*Software Engineering*), meskipun dianggap kuno. Pendekatan pengembangan ini dilakukan secara sistematis dan berurutan. Model ini dikenal sebagai *waterfall* karena setiap tahap harus menunggu tahap sebelumnya selesai dan berlangsung secara berurutan. Meskipun banyak model pengembangan perangkat lunak yang lebih fleksibel dan adaptif telah muncul sejak itu, model *waterfall* masih sering digunakan dalam praktik pengembangan perangkat lunak saat ini.

Penggunaan konstruksi *HTTP* oleh *REST* juga menghadirkan beberapa batasan. Beberapa batasan dari *HTTP* juga berubah menjadi kelemahan dalam gaya arsitektur *REST*. Contohnya, *HTTP* tidak menyimpan informasi berbasis negara (*state*) antara siklus permintaan-respon. Hal ini berarti aplikasi yang mengadopsi *REST* harus bersifat *stateless*, dan manajemen state apa pun harus dilakukan oleh klien [6]. Arsitektur untuk implementasi *web service* yang mengadopsi konsep perpindahan antara state. *REST* menjadi populer karena kesederhanaannya dalam menciptakan standar, kerangka kerja, dan teknologi baru [7].

2. METODE PENELITIAN

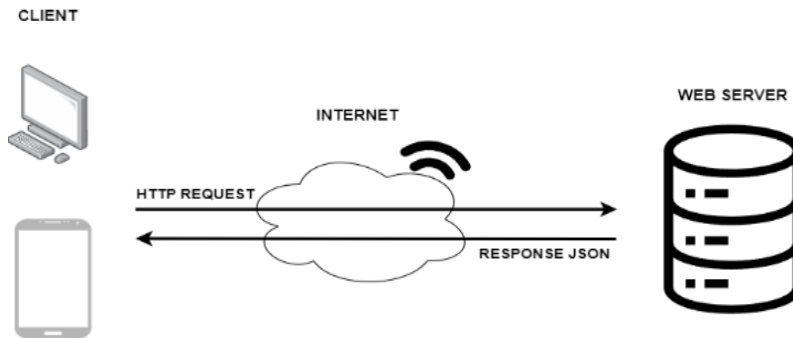
2.1 Data Penelitian

Sebagai perusahaan yang memiliki sekitar kurang lebih 500 karyawan, PT. Prestasi Retail Innovation menyadari betapa pentingnya layanan pelaporan masalah dan pencatatan aset. Setiap masalah yang dilaporkan harus segera diperbaiki agar tidak mengganggu produktivitas dan tidak menambah beban kerja jika masalah tersebut dibiarkan membesar. Untuk memastikan bahwa semua masalah dapat dilaporkan dan diterima dengan cepat oleh tim terkait, diperlukan sebuah aplikasi atau layanan yang memudahkan karyawan dalam membuat

laporan dan memastikan tim terkait dapat menerima laporan tersebut dengan aman. Oleh karena itu, dalam penelitian ini, sumber data yang digunakan adalah data hasil dari laporan *user* dan aset yang digunakan, *category* aset dan laporan masalah, serta data pelaporan semua masalah yang saat ini dimiliki oleh PT. Prestasi *Retail Innovation*. Manfaat dari penelitian ini adalah website pelaporan masalah dan peminjaman aset yang diimplementasikan dengan *RESTFUL web service* agar semua fitur di dalam *web* dapat dengan mudah diakses dan digunakan oleh *user* maupun *admin*.

2.1.1 Arsitektur Kerja Sistem

Berikut merupakan rancangan arsitektur kerja system pada *web service* gambar 1:



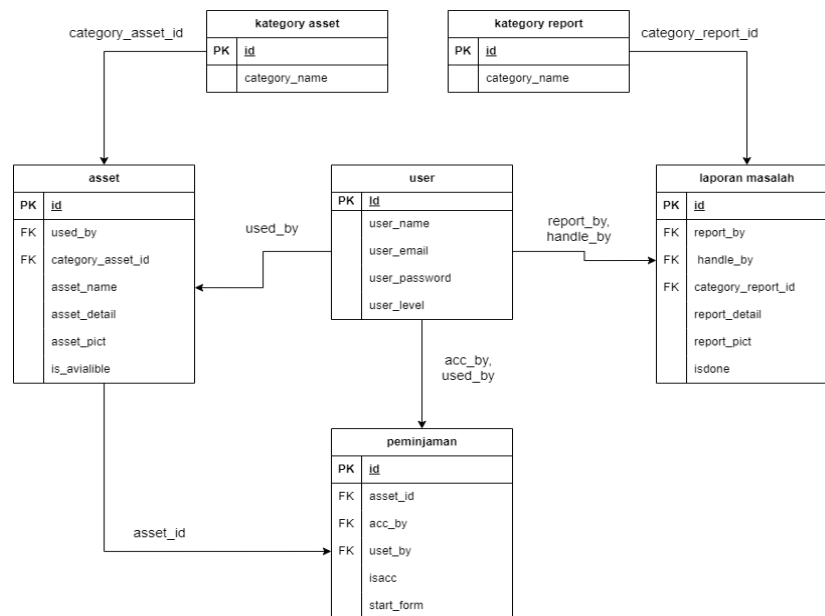
Gambar 1. Arsitektur kerja sistem

2.2 Rancangan Sistem Basis Data

Rancangan basis data adalah proses pembuatan struktur Database yang sesuai dengan kebutuhan data. Rancangan ini mencakup pembuatan ERD (*Entity Relationship Diagram*), LRS (*Logical Record Structure*), dan spesifikasi basis data [8].

2.2.1 Logical Record Sturcture (LRS)

Berdasarkan hasil transformasi ERD ke LRS (*Logical Record Structure*), adapun rancangan LRS ditunjukkan pada gambar 2:



Gambar 2 Logical Record Sturcture (LRS)

2.3 Rancangan Layanan Web Service

Berikut ini adalah merupakan rancangan *web service* yang akan menjelaskan tentang URL API (*Uniform Resource Locator Application Programming Interface*) yang juga dikenal sebagai *Endpoint* dari Server [9].

2.3.1 Rancangan Web Service Endpoint

Berikut adalah rancangan *web service endpoint user*. Table 1:

Tabel 1. Rancangan *web service endpoint* dari sisi *user*

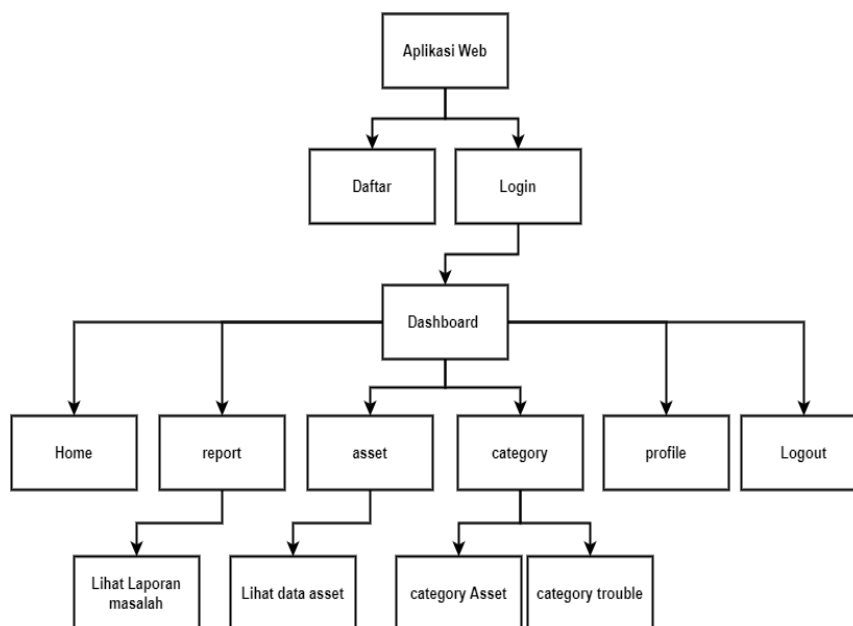
Nama Layanan	Metode	Endpoint	Parameter	Output
Daftar	POST	User (Register)		Sukses, message
Login	POST	User (login)		Sukses, token, message
Data Laporan	POST	User (report trouble)		Success, message
Data aset	POST	User (aset)		Success, message
List status Laporan	GET	User (report trouble)	Id report trouble	Success, message
Profile	GET	User	Id User	Success, User, message
Profile	PUT	User (edit)	Id User	Success, User, message

2.4 Rancangan Menu

Rancangan menu aplikasi adalah sekumpulan pilihan menu yang ada dalam sebuah aplikasi. Menu aplikasi ini menyediakan fasilitas dan fungsi yang akan ditampilkan kepada pengguna. Dalam rancangan menu aplikasi ini, terdapat dua platform yang berbeda, yaitu rancangan menu aplikasi *web* dan aplikasi Android.

2.4.1 Rancangan Menu Aplikasi Web

Adapun gambar rancangan menu aplikasi web tersebut dapat dilihat pada gambar 3:



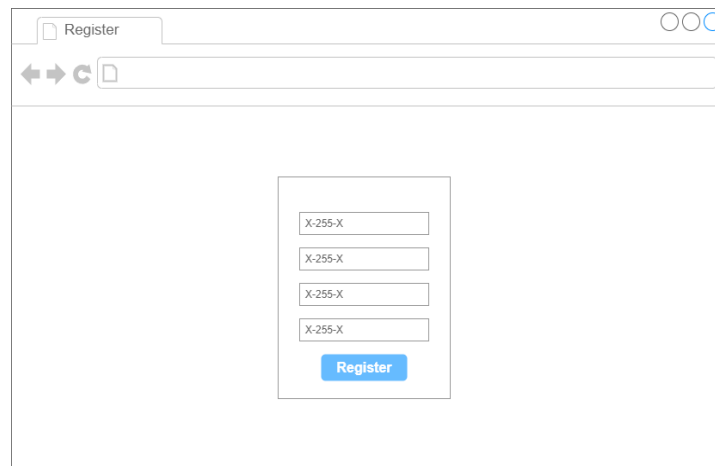
Gambar 3. Rancangan menu aplikasi *web*

2.5 Rancangan Layar

Dalam pengembangan suatu sistem aplikasi, rancangan tampilan (*user interface*) memiliki peranan yang sangat penting. Rancangan tampilan haruslah mudah dipahami, sehingga pengguna atau pengguna merasa nyaman saat menggunakan aplikasi [10].

2.5.1 Rancangan Layar Web Register

Rancangan layar register ini merupakan tampilan form pendaftaran yang dimana saat user jika ingin mendaftarkan diri, adapun rancangan layar form pendaftaran tersebut dapat dilihat pada gambar 4:



Gambar 4. Rancangan layer web register

3. HASIL DAN PEMBAHASAN

Pada bagian ini, akan dibahas mengenai implementasi, instalasi, konfigurasi, penerapan, dan evaluasi dari suatu sistem *web service*. Pembahasan akan mencakup hal-hal sebagai berikut.

3.1 Lingkungan Percobaan

Dalam pembuatan sistem *web service* ini memerlukan beberapa perangkat lunak (*software*) ataupun perangkat keras untuk mendukung kinerja dari alat ini. Berikut merupakan spesifikasi yang digunakan:

3.1.1 Spesifikasi Perangkat Lunak (*software*)

Perangkat lunak atau (*software*) yang digunakan untuk membuat program tugas akhir ini selain banyak digunakan oleh pengembang dan juga masih mendapatkan update sehingga cukup memudahkan untuk penggunaannya, berikut software yang dapat digunakan:

- Sistem Operasi Windows 10
- XAMPP
- phpMyAdmin
- PHP 8.0 / Laravel 9
- Visual Studio Code
- Postman
- Goggle Chrome
- Draw.io
- Android Studio

3.1.2 Spesifikasi Pada Perangkat Keras (*Hardware*)

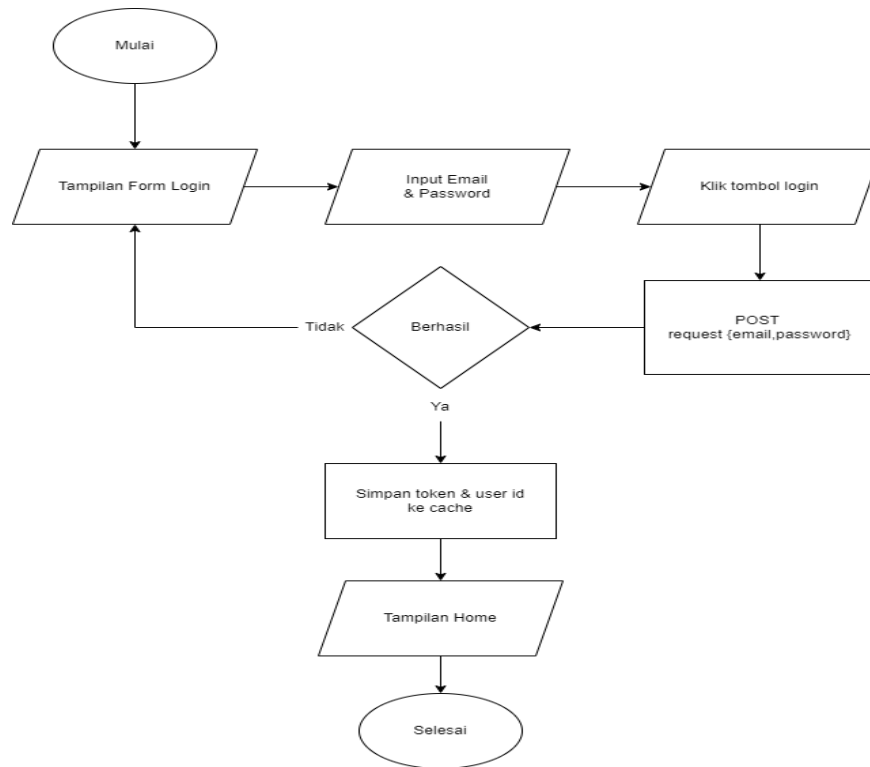
Perangkat keras atau (*hardware*) yang di rekomendasikan untuk digunakan dan mendukung kinerja program dari tugas akhir ini adalah sebagai berikut:

- Laptop HP Pavilion x360, Intel Core i5, RAM 8GB, HDD 500GB

3.2 Flowchart sistem

3.2.1 Flowchart login

Flowchart ini menggambarkan proses *login* pengguna. Pengguna akan diminta untuk memasukkan *email* dan *password* sebagai permintaan *login* ke *algoritme*. Jika pada *email* dan *password* yang dimasukkan *valid*, pengguna akan menerima data diri dan token yang digunakan untuk mengakses fitur-fitur dalam aplikasi. Namun, jika *email* dan *password* tidak *valid*, pengguna akan diminta untuk memasukkan *email* dan *password* yang *valid*. Gambar 5:



Gambar 5. Flowchart Login

3.3 Algoritme

Algoritme adalah urutan Langkah-langkah logis dan sistematis yang digunakan untuk memecahkan masalah secara khusus.

3.3.1 Algoritme web service

Berikut merupakan algoritme dari web service dapat di lihat pada gambar 6:

1. Start
2. Client mengirim http request
3. Server menerima request dari client
- 4.
5. if endpoint tersedia
6. if endpoint = login/register
7. if username & password valid
8. Server mengembalikan status kode 200 "OK"
9. Server mengembalikan response json = data jwt token
10. else
11. Server mengembalikan status kode 200 "OK"
12. Server mengembalikan response json = "Username atau password salah"
13. end if
- 14.
15. else
16. if token valid
17. Server mengembalikan status kode 200 "OK"
18. Server mengembalikan response json = sesuai data yang diminta
19. else
20. Server mengembalikan status kode 200 "OK"
21. Server mengembalikan response json = "tidak memiliki otentikasi"
22. end if
23. else
24. Server mengembalikan status kode 400 "Not Found"
25. Server mengembalikan response json = "Not Found"

```

26. end if
27.
28. End
  
```

Gambar 6. Web service

3.4 Pengujian

Pada bagian ini, penulis melakukan pengujian menggunakan metode *black box testing*. Metode yang di digunakan ini untuk mengidentifikasi kesalahan dan menguji fungsi-fungsi dari aplikasi secara keseluruhan. Dalam pengujian ini, penulis memeriksa apakah hasil input diterima dengan benar dan output yang dihasilkan sesuai dengan harapan. Badalah hasil pengujian *black box testing* pada aplikasi tersebut. Tabel 6:

Tabel 2. Black Box pada Service User

No	Nama Layanan	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Login	Menginput <i>email</i> dan <i>password</i> pada <i>request body</i> dengan metode POST	Merespons dengan status <i>success</i> dan mendapatkan token	Sesuai Harapan	Valid
2.	Register	Menginput nama, <i>email</i> , dan <i>password</i> pada <i>request body</i> dengan metode POST	Merespons dengan status <i>success</i>	Sesuai Harapan	Valid
3.	Logout	Mengirim <i>request body</i> kosong dengan metode POST dan otorisasi JWT	Merespons dengan status <i>success</i> dan menghapus token di <i>database</i>	Sesuai Harapan	Valid
4.	Get Category Aset	Menampilkan semua data kategori aset dengan metode GET dan otorisasi JWT	Merespons dengan status <i>success</i> dan mendapatkan semua data kategori aset	Sesuai Harapan	Valid
5.	Get Data Aset	Menampilkan semua data aset yang bisa dipinjam dengan metode GET dan otorisasi JWT	Merespons dengan status <i>success</i> dan mendapatkan semua data aset yang bisa dipinjam	Sesuai Harapan	Valid
6.	Get Data Aset by User Id	Menampilkan semua data aset yang dipinjam oleh <i>user</i> terkait dengan metode GET dan otorisasi JWT	Merespons dengan status <i>success</i> dan mendapatkan semua data aset yang dipinjam oleh <i>user</i> terkait	Sesuai Harapan	Valid
7.	Pengajuan Peminjaman	Menginput aset <i>id</i> , <i>User id</i> , dan tanggal pengajuan pada <i>request body</i> dengan metode POST dan otorisasi JWT	Merespons dengan status <i>success</i>	Sesuai Harapan	Valid
8.	Get Category Laporan	Menampilkan semua data kategori laporan dengan metode GET dan otorisasi JWT	Merespons dengan status <i>success</i> dan mendapatkan semua data kategori laporan	Sesuai Harapan	Valid
9.	Get Data Laporan by User Id	Menampilkan semua data laporan yang di ajukan <i>User</i> terkait dengan metode GET dan otorisasi JWT	Merespons dengan status <i>success</i> dan mendapatkan semua laporan yang diajukan <i>user</i> terkait	Sesuai Harapan	Valid

10.	Tambah laporan	data	Menginput keterangan, kategori, dan gambar pendukung pada <i>request body</i> dengan metode POST dan otorisasi JWT	Merespons dengan status <i>success</i>	Sesuai Harapan	<i>Valid</i>
11.	Update laporan	data	Menginput keterangan, kategori, dan gambar pendukung pada <i>request body</i> dengan metode <i>UPDATE</i> dan otorisasi JWT pada laporan yang sudah ada	Merespons dengan status <i>success</i>	Sesuai Harapan	<i>Valid</i>

4. KESIMPULAN

Setelah melalui tahap pengujian, dapat disimpulkan bahwa implementasi *RESTful web service* pada PT. Prestasi Retail Indonesia telah berhasil tercapai. *Web service* yang dibangun telah efektif dan *user-friendly*, memudahkan akses dari berbagai lokasi, serta meningkatkan efisiensi dalam penanganan masalah di perusahaan. Selain itu, langkah-langkah keamanan yang kuat telah diterapkan dengan sukses untuk melindungi data perusahaan dari potensi pencurian, yang telah membangun kepercayaan dari internal maupun eksternal terhadap perusahaan. Meskipun masih ada beberapa fungsi dasar dari sistem pinjaman yang perlu diperbaiki, langkah awal ini menunjukkan progres positif dalam menciptakan sistem yang andal dan aman dalam mengelola masalah dan informasi di PT. Prestasi Retail Indonesia.

DAFTAR PUSTAKA

- [1] Gustiegan, G.Y. and Painem (2022) 'Implementasi *WEB SERVICE Restful* dengan Autentikasi *JSON Web Token* dan *Algoritme Kriptografi AES-256* Untuk Aplikasi Pinjaman Laboratorium Berbasis Mobile Pada Universitas Budi Luhur', *Jurnal Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, 19(1), pp. 9–16.
- [2] M. A. Kurniawan, and R. Saputra, "Implementasi *WEB SERVICE* Pada Integrasi Aplikasi Rental Mobil Online," *JURNAL MASYARAKAT INFORMATIKA*, vol. 9, no. 2, pp. 33-43, Nov. 2018.
- [3] Putra, M.G.L. and Putera, M.I.A. (2019) 'Analisis Perbandingan Metode Soap Dan Rest Yang Digunakan Pada Framework Flask Untuk Membangun *WEB SERVICE*', *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, 14(2), pp. 1–7. Available at: <https://doi.org/10.33005/scan.v14i2.1480>.
- [4] Rahmatulloh, A., Sulastri, H. and Nugroho, R. (2018) 'Keamanan *RESTful WEB SERVICE* Menggunakan *JSON Web Token (JWT) HMAC SHA-512*', *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 7(2). Available at: <https://doi.org/10.22146/jnteti.v7i2.417>.
- [5] Aceng Abdul Wahid (2020) 'Analisis Metode *Waterfall* Untuk Pengembangan Sistem Informasi', *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK*, (November), pp. 1–5.
- [6] Rizal, R. and Rahmatulloh, A. (2019) 'Restful *WEB SERVICE* Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan', *Jurnal Ilmiah Informatika*, 7(01), p. 54. Available at: <https://doi.org/10.33884/jif.v7i01.1004>.
- [7] Verma, S. (2018) 'APIs versus *WEB SERVICES*' (<https://blogs.mulesoft.com/dev/apidev/apis-versus-web-services/>) diakses 29 April 201
- [8] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING (Satuan Tulisan Ris. dan Inov. Teknol., vol. 3, no. 2, p. 206, 2018.*
- [9] A. Bogdan-Alexandru, A.-C. Cosu-Pop, G. Sorin Catalin dan B. Costin Anton, "A STUDY ON USING WATERFALL AND AGIL
- [10] D. S. Purnia, A. Rifai dan S. Rahmatullah, "Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Aplikasi Bantuan Sosial Berbasis Android," *Semnastek*, 2019.