

PENGEMBANGAN WEB SERVICE APLIKASI MANAJEMEN ASET UPT TIK UNIVERSITAS NEGERI JAKARTA

Tri Baswara Ari Aji¹, Hamidillah Ajje², Muriennugraheni³

¹ Mahasiswa Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

² Dosen Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

³ Dosen Prodi Sistem dan Teknologi Informasi, Teknik Elektro, FT – UNJ

¹ tbariaji@gmail.com, ² hamidillah@unj.ac.id, ³ muriennugraheni@unj.ac.id

Abstrak

UPT TIK bertanggung jawab dalam mengelola aset-aset IT yang ada di UNJ. Dalam pelaksanaannya UPT TIK belum memiliki sistem manajemen aset yang baik sehingga masih banyak aset IT yang terlantar keberadaannya. Tujuan dari penelitian ini adalah menghasilkan web service yang dapat digunakan untuk membuat aplikasi manajemen aset yang bersifat adaptif terhadap penambahan jenis aset baru kedepannya.

Dalam pengembangan web service aplikasi manajemen aset UPT TIK UNJ ini menggunakan metode spiral dengan melakukan wawancara dan studi literatur. Wawancara menghasilkan daftar kebutuhan fungsional dan selanjutnya dibuat rancangan Entity Relationship Diagram (ERD), database, endpoint. Kemudian implementasi dilakukan dengan menggunakan perangkat lunak Node.JS dengan framework Express.JS dan menghasilkan 41 endpoint. Pengujian dilakukan dengan menggunakan metode unit testing dengan aplikasi Postman yang dilakukan terhadap 41 endpoint dan didapat hasil pengujian semua endpoint berjalan sukses sehingga memenuhi kriteria dan layak digunakan.

Kata kunci : *Web service, Aplikasi Manajemen Aset, UPT TIK UNJ, Metode Spiral, Unit Testing, Endpoint, Node.js, Express.js*

1. Pendahuluan

Universitas Negeri Jakarta (UNJ) adalah salah satu perguruan tinggi negeri yang terletak di Ibu kota negara Indonesia. Dengan status Jakarta sebagai Ibu kota negara Indonesia, maka UNJ menjadi salah satu cerminan pendidikan masyarakat Ibu kota negara Indonesia. Hal ini merupakan salah satu penyebab UNJ perlu dilengkapi dengan berbagai aset untuk menunjang kebutuhan yang ada. Saat ini UNJ memiliki berbagai macam aset, mulai dari kendaraan, gedung, aset dalam ruang, aset laboratorium, aset IT dan lain sebagainya. Beberapa aset IT yang dimiliki UNJ saat ini dikelola oleh UPT TIK UNJ.

UPT TIK yang dulunya bernama PUSTIKOM merupakan wadah teknologi yang menjembatani kampus dengan kehidupan baik dalam maupun luar kampus agar dapat berkomunikasi dengan jaringan internet. Sejak 2016 PUSTIKOM berganti nama menjadi UPT TIK (Unit Pelayanan Teknis Teknologi Informasi dan Komunikasi) UNJ. Di mana UPT TIK UNJ menjadi unit utama pengelola pemanfaatan teknologi informasi dan komunikasi untuk mendukung kegiatan pendidikan, penelitian, pengabdian masyarakat, penyimpanan dan pengolahan data serta dukungan bagi manajemen universitas.

UPT TIK UNJ memiliki tugas untuk mengelola seluruh aset IT yang ada di bawah naungan UPT TIK UNJ. Berdasarkan hasil wawancara penulis dengan pimpinan dan karyawan UPT TIK UNJ, dalam UPT TIK UNJ saat ini belum memiliki sistem manajemen aset sehingga masih banyak aset IT UNJ yang kondisinya tidak diketahui keberadaannya atau kepemilikannya saat ini. Belum adanya sistem manajemen aset menyebabkan pembuatan laporan aset menjadi lebih sulit. Sistem manajemen aset yang baik sangat diperlukan untuk membantu mendata seluruh aset yang ada sehingga tidak ada aset yang terlantar nantinya dan sistem ini menjadi salah satu kebutuhan yang sangat penting untuk dimiliki UPT TIK UNJ. Salah satu sistem manajemen aset yang dapat digunakan adalah sistem manajemen aset berbasis *web*.

Sistem manajemen aset yang dimaksud adalah sistem yang mampu mengelola aset yang ada mulai dari usulan pengadaan, perawatan, pendataan pemegang aset, pendataan keberadaan aset, hingga pemusnahan aset. Sistem manajemen aset berbasis *web* saat ini menjadi salah satu kebutuhan yang harus terpenuhi oleh bagian perlengkapan dalam membantu mengerjakan tugas dari bagian perlengkapan. Dengan adanya sistem manajemen aset berbasis *web* maka bagian perlengkapan dapat mengetahui kondisi aset dan keberadaan aset yang dimilikinya dengan mudah karena dapat dikelola secara daring dengan menggunakan akses internet. Berdasarkan

dengan latar belakang yang telah dijelaskan tersebut maka dilakukan penelitian dengan judul “Pengembangan *Web service* Aplikasi Manajemen Aset di UPT TIK Universitas Negeri Jakarta”.

2. Dasar Teori

Sesuai dengan topik yang diangkat pada penelitian ini yaitu Pengembangan *Web service* Aplikasi Manajemen Aset di Universitas Negeri Jakarta maka penulis akan membahas kerangka teoritik penulisan sebagai berikut:

2.1. Manajemen Aset

Menurut Limbong (2016: 5) manajemen aset adalah suatu ilmu untuk pengelolaan kekayaan negara/daerah berdasarkan asas fungsional, kepastian hukum, transparansi dan keterbukaan, efisiensi, akuntabilitas, dan kepastian nilai sehingga tercapai tujuan pengelolaan aset. Manajemen aset juga dapat disebut sebagai manajemen inventori, dimana menurut Paryati & Wahyu (2019: 276) manajemen inventori adalah kegiatan pengelolaan dan pengendalian aset baik yang dimiliki oleh pemerintah, organisasi maupun perusahaan. Dari pengertian diatas, maka dapat disimpulkan manajemen aset adalah kegiatan untuk pengelolaan aset serta pengendalian aset secara terstruktur dan dapat memantau keadaan dari setiap aset yang dimiliki serta bertujuan memiliki proses dari pengadaan hingga penghapusan yang baik dan jelas.

2.2. Aplikasi

Aplikasi memiliki beberapa definisi, salah satunya menurut Siregar, dkk. (2018: 113) adalah alat terapan yang digunakan secara khusus dan terpadu sesuai dengan kemampuan yang dimilikinya. Aplikasi adalah perangkat komputer yang siap pakai bagi pengguna. Definisi lain dari aplikasi menurut Syam & Arie (2018: 2) adalah suatu subkelas dalam perangkat lunak komputer yang memanfaatkan kemampuan komputer secara langsung untuk melakukan tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung mengaplikasikan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Dari beberapa definisi di atas dapat disimpulkan bahwa aplikasi adalah suatu program komputer siap pakai yang dapat digunakan untuk mengerjakan tugas tertentu. Aplikasi juga merupakan alat terapan yang secara khusus digunakan dan terpadu sesuai dengan kemampuannya.

2.3. *Web service*

Web service adalah bagian dari *website*, di mana *website* sebuah aplikasi yang mengandung halaman tentang dokumen-dokumen ataupun informasi dalam bentuk text, gambar dan lain-lain dalam sebuah *website* (Nurhayati, dkk., 2017: 17). Definisi lain dari *website* datang dari Elu (2013: 113) di mana *website* merupakan suatu layanan penyajian informasi yang menggunakan konsep tautan (*hyperlink*) guna mempermudah pengguna internet untuk melakukan pencarian informasi. Dari beberapa definisi di atas dapat disimpulkan bahwa *website* adalah kumpulan dari beberapa halaman situs yang menyajikan beragam informasi dalam bentuk *text*, video, audio, gambar, ataupun *file-file* yang lain dengan menggunakan konsep tautan (*hyperlink*) yang dapat diakses melalui jalur internet sehingga dapat diakses dimanapun selama terhubung dengan internet. Menurut Ahmad dkk., (2021), dalam beberapa tahun terakhir, *web service* yang berkomunikasi melalui protokol *web HyperText Transfer Protocol* (HTTP) semakin populer.

Web service dapat diartikan juga sebuah metode pertukaran data, tanpa memperhatikan dimana sebuah *database* ditanamkan, bahasa yang digunakan aplikasi dimana sebuah aplikasi mengkonsumsi data, dan di suatu *platform* dimana data itu dikonsumsi (Utomo, dkk., 2017: 4). Dari definisi di atas, *web service* adalah perangkat lunak yang mampu membantu interaksi antar sistem pada suatu jaringan dan memudahkan pertukaran data.

2.4. *Node.JS*

Menurut Fajrin (2017: 41) *Node.JS* adalah sistem perangkat lunak yang didesain untuk mengembangkan aplikasi *web*. *Node.JS* dapat juga disebut sebagai *runtime environment*. Aplikasi ini ditulis dalam campuran JavaScript dan Bahasa C++, mempunyai model basis *event* dan *asynchronous* I/O. Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada *web browser*, *Node.JS* dijalankan sebagai aplikasi server. *Node.JS* mampu berjalan di server karena mempunyai dukungan dari *V8 Engine* yang dibuat oleh Google dan beberapa modul bawaan yang terintegrasi seperti modul *http*, modul *filesystem*, modul *security* dan beberapa modul penting lainnya.

Sedangkan menurut Nugroho & Katardi (2016: 8) *Node.JS* adalah sebuah *platform* untuk membangun *real-time application*. *Node.JS* dapat menangani *event input-output* server, dengan kata lain *Node.JS* dapat memungkinkan para *developer* Javascripts untuk membuat *event-driven servers* dalam JavaScript. Dari beberapa pengertian diatas dapat disimpulkan bahwa *Node.JS* adalah sebuah sistem perangkat lunak yang berupa *platform* untuk membangun *real-time* aplikasi yang dibuat dari campuran bahasa C++ dan JavaScript dan membuat *event-*

driven server dalam JavaScript. *Node.JS* dapat berjalan dengan dukungan dari *V8 Engine* buatan Google dan beberapa modul lainnya serta dapat menangani *event input-output* server.

2.5. Framework

Framework menurut Mediana & Nurhidayat (2018: 76) adalah kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka kerja tersebut, waktu yang digunakan akan lebih singkat dan memudahkan dalam melakukan perbaikan. Pengertian *framework* juga datang dari Rais & Sirad (2019: 27) di mana *framework* adalah sebuah struktur konseptual dasar yang digunakan untuk memecahkan sebuah permasalahan atau isu-isu kompleks. Dari beberapa definisi diatas dapat disimpulkan bahwa *framework* adalah sebuah kerangka kerja dari sebuah *website* yang akan dibangun dan memudahkan untuk membuat *website* lebih singkat serta memudahkan dalam sisi perbaikannya.

2.6. Express.JS

Berdasarkan pendapat dari Fajrin (2017: 40-41) *Express.JS* adalah sebuah *Node.JS framework* yang berfungsi mempermudah pembuatan aplikasi berbasis *Node.JS* dengan memanfaatkan *design pattern* yang dapat disesuaikan dan sangat fleksibel. Selain itu, *Express.JS* juga merupakan *framework* yang sangat ringan dan cocok untuk membuat web aplikasi dan *API*. Sedangkan menurut Nirgudkar & Singh (2017: 3238) *Express.JS* adalah *framework server side* yang dibangun dalam lingkup *node js*. *Framework* ini digunakan untuk mengelola *request* dari *client* ke *server* termasuk untuk pengelolaan *routing* dan operasi HTTP (*PUT, GET, POST, dll*). *Express* dapat diartikan sebagai suatu *middleware* yang bertanggung jawab pada pengelolaan siklus *request response* dan menjamin tidak ada suatu *request* yang tertinggal (dibiarkan menggantung). Dari beberapa pengertian diatas dapat diambil kesimpulan bahwa *Express.JS* adalah *framework* yang berasal dari *Node.JS* yang berfungsi untuk mengelola *request* dari *client* ke *server* dan menjamin tidak ada *request* yang tertinggal serta sangat fleksibel untuk aplikasi berbasis *Node.JS*.

2.7. Restful API

Menurut Putra & Putera (2019: 2) *Restful web service* atau yang dikenal sebagai *Restful Web API* merupakan sebuah *web service* yang diimplementasikan menggunakan *http* dengan menggunakan prinsip-prinsip *REST*. *Service*nya menggunakan *method* milik *http* antara lain *GET, PUT, POST or DELETE*. Definisi lain dikemukakan oleh Indrawan & Adil (2017: 12) *REST* adalah suatu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat diartikan seperti jika *browser* meminta suatu halaman *web*, maka server akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Dari beberapa pengertian diatas dapat diambil kesimpulan bahwa *Restful API* adalah salah satu jenis *web service* dengan penerapan konsep perpindahan antar *state* yang menggunakan *hypertext transfer protocol (http)* untuk komunikasi datanya dan menggunakan *method* dari *http* seperti *GET, PUT, POST, DELETE*.

2.8. JavaScript Object Notation (JSON)

Berdasarkan pengertian dari Styawan (2018: 4) *JSON* adalah suatu format singkat pertukaran data komputer. Format *JSON* berbasis teks dan terbaca oleh manusia dan berfungsi untuk merepresentasikan struktur data sederhana dan larik asosiatif. Format *JSON* sering digunakan untuk transmisi data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi. Pendapat lain datang dari *website json.org*, dimana *JSON (JavaScript Object Notation)* adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, dan mudah diterjemahkan dan dibuat oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. *JSON* merupakan format teks yang independen dan tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena itu, menjadikan *JSON* ideal sebagai bahasa pertukaran-data. Dengan kata lain, *JSON (JavaScript Object Notation)* adalah format untuk pertukaran data yang ringan dan dapat dengan mudah dibaca dan ditulis oleh manusia serta mudah diterjemahkan dan dibuat oleh komputer. *JSON* berupa teks/*string* dan tidak bergantung pada bahasa pemrograman apapun.

2.9. POSTMAN

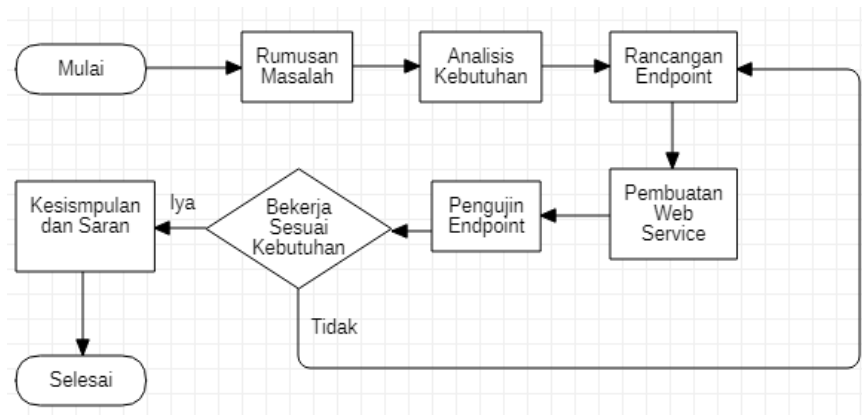
Postman menurut Duskarnaen, dkk. (2019: 41) adalah aplikasi (berupa plugin untuk browser chrome, yang berfungsi sebagai *REST Client* atau dengan kata lain adalah aplikasi yang berfungsi untuk melakukan uji coba *REST API* yang telah dibuat (Pranawa). Definisi lain datang dari Widyoutomo (2019: 38) dimana *Postman* adalah alat yang disediakan oleh Google Chrome dengan fungsi utama sebagai *GUI API caller* namun sekarang juga menyediakan fitur lain yaitu *Sharing Collection API for Documentation, Testing API, Realtime Collaboration Team*, dan lain-lain. Dari beberapa pengertian diatas dapat disimpulkan bahwa *Postman* adalah aplikasi/tools yang disediakan oleh Chrome dan hanya bisa berjalan di browser Chrome dimana berfungsi untuk melakukan uji coba *REST API* yang telah dibuat.

2.10. Metode Pengembangan Spiral

Ada beberapa metodologi yang dapat digunakan dalam mengembangkan suatu perangkat lunak, salah satunya adalah metode Spiral yang akan digunakan dalam pengembangan *web service* pada penelitian ini. Menurut Lolong, dkk. (2016: 266) metode spiral adalah kombinasi antara proses repetisi yang dimiliki oleh *prototyping* dan aspek sistematis dan terkontrol yang dimiliki oleh spiral model yang memungkinkan *software* dikembangkan dengan lebih cepat dan lengkap. Sedangkan menurut Utomo & Alfaritzi (2018: 44) Model spiral (*spiral model*) adalah model proses *software* yang evolusioner yang merangkai sifat iteratif dari prototipe dengan cara kontrol dan aspek sistematis dari model sekuensial linier. Model ini berpotensi untuk pengembangan versi pertambahan *software* secara cepat. Dapat disimpulkan bahwa metode pengembangan spiral merupakan pengembangan dari metode yang sudah ada sebelumnya yang dapat memungkinkan *software* lebih cepat dikembangkan dan lebih lengkap.

3. Metodologi

Alat yang digunakan dalam melaksanakan penelitian ini adalah sebuah laptop yang dilengkapi dengan beberapa *tools* yang digunakan seperti Visual Studio Code, Node.JS, Postman. Bahan yang digunakan untuk penelitian ini adalah hasil wawancara dengan pimpinan dan karyawan UPT TIK UNJ. Untuk memperlihatkan prosedur penelitian yang digunakan untuk mencapai tujuan penelitian. Diagram alir penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

Proses pertama adalah merumuskan masalah yang dipaparkan pada bahasan sebelumnya. Kemudian, melakukan analisis kebutuhan berdasarkan hasil wawancara dengan pimpinan dan karyawan UPT TIK UNJ. Setelah analisis kebutuhan dilakukan, langkah berikutnya adalah melakukan perancangan *web service* sesuai dengan kebutuhan dari sistem yang dibuat menjadi berbagai *endpoint*, kemudian menentukan analisis resiko dengan melakukan enkripsi pada *password* di *database* dengan menggunakan algoritma *bcrypt* agar tidak dapat diketahui isi *password*-nya. Langkah selanjutnya adalah membangun *web service* sesuai dengan hasil rancangan dan melakukan pengujian menggunakan *Postman* untuk berbagai *endpoint* yang telah dibuat.

Postman adalah sebuah aplikasi (berupa *plugin*) untuk *browser Chrome* dengan fungsi utama sebagai *REST Client* untuk uji coba *REST API*. Jika nantinya masih terdapat kesalahan, maka dilihat kembali rancangan dan hasil *script* koding yang sudah dibuat untuk ditelusuri kesalahannya, karena bisa saja kesalahan terdapat pada rancangan yang tidak sesuai atau logika yang kurang tepat. Jika sudah layak, maka langkah selanjutnya adalah menarik kesimpulan dan saran.

Berdasarkan hasil wawancara dengan staf dan pimpinan UPT TIK UNJ, maka didapatkan beberapa kebutuhan fungsional untuk membuat *web service* aplikasi manajemen aset UPT TIK UNJ yang terlampir pada Tabel 3.1.

Tabel 3.1 Kebutuhan Fungsional

No	Kebutuhan	Keterangan
1	Melakukan login	Admin dan petugas melakukan login untuk masuk ke dalam website
2	Merubah password	Admin dan petugas dapat merubah password
3	Menampilkan jenis aset	Admin dan petugas dapat menampilkan jenis aset
4	Menambah jenis aset	Admin dapat menambahkan jenis aset
5	Mengubah jenis aset	Admin dapat mengubah data jenis aset
6	Menampilkan tipe aset	Admin dan petugas dapat menampilkan tipe aset
7	Menambah tipe aset	Admin dapat menambahkan tipe aset

8	Mengubah tipe aset	Admin dapat mengubah data tipe aset
9	Menampilkan detail aset	Admin dan petugas dapat menampilkan detail aset
10	Menambah detail aset	Admin dapat menambah detail aset
11	Mengubah detail aset	Admin dapat mengubah data detail aset
13	Menampilkan pemeriksaan berkala	Admin dan petugas dapat menampilkan pemeriksaan berkala
14	Menambah pemeriksaan berkala	Admin dan petugas dapat menambah pemeriksaan berkala
15	Mengubah pemeriksaan berkala	Admin dan petugas dapat mengubah data pemeriksaan berkala
16	Menampilkan daftar surat tugas	Admin dapat menampilkan surat tugas
17	Menambah daftar surat tugas	Admin dapat menambah surat tugas
18	Mengubah daftar surat tugas	Admin dapat mengubah data surat tugas
19	Menampilkan pengaduan	Admin dan petugas dapat menampilkan pengaduan
20	Menambah pengaduan	Admin dan petugas dapat menambah pengaduan
21	Mengubah pengaduan	Admin dan petugas dapat mengubah data pengaduan
22	Menampilkan tindakan	Admin dan petugas dapat menampilkan tindakan
23	Menambah tindakan	Admin dan petugas dapat menambah tindakan
24	Mengubah tindakan	Admin dan petugas dapat mengubah data tindakan
25	Menampilkan peminjaman	Admin dapat menampilkan peminjaman
26	Menambah peminjaman	Admin dapat menambah peminjaman
27	Mengubah peminjaman	Admin dapat mengubah data peminjaman
28	Menampilkan detail peminjaman	Admin dapat menampilkan detail peminjaman
29	Menambah detail peminjaman	Admin dapat menambah detail peminjaman
30	Mengubah detail peminjaman	Admin dapat mengubah data detail peminjaman
31	Menampilkan data gudang	Admin dan petugas dapat menampilkan data gudang
32	Menambah data gudang	Admin dapat menambah data gudang
33	Mengubah data gudang	Admin dapat mengubah data gudang
34	Menampilkan data rak	Admin dan petugas dapat menampilkan data rak
35	Menambah data rak	Admin dapat menambah data rak
36	Mengubah data rak	Admin dapat mengubah data rak
37	Menampilkan petugas	Admin dapat menampilkan data petugas
38	Menambah petugas	Admin dapat menambahkan data petugas
39	Menghapus petugas	Admin dapat menghapus data petugas
40	Menampilkan gedung	Admin dan petugas dapat menampilkan gedung
41	Menambah gedung	Admin dapat menambah gedung

4. Hasil dan Analisis

Produk yang dikembangkan pada penelitian ini menghasilkan endpoint sebanyak 41 buah. Setiap *endpoint* yang dihasilkan, dikembangkan dengan beberapa tahapan, yaitu: 1) penentuan data yang akan digunakan; 2) penentuan algoritma; 3) penulisan kode program; 4) pengujian terhadap *endpoint*. Pengembangan *endpoint* dilakukan dengan menggunakan metode penelitian *spiral* hingga berhasil dikembangkan dengan baik tanpa kesalahan sesuai dengan target pembuatan produk. Tidak lupa juga untuk memperhatikan pesan yang akan dikirimkan agar memudahkan pihak pengembang di sisi *front-end* dalam menerima *endpoint* yang dikirimkan.

Setelah dilakukan pengembangan terhadap 41 *endpoint* yang telah dibuat, menunjukkan bahwa setiap *endpoint* memiliki *response time* yang berbeda-beda. *Endpoint* yang memiliki *response time* terlama terdapat pada *requirement* nomor 37, yaitu *endpoint* untuk menambahkan data petugas baru. Hal ini disebabkan karena banyaknya data yang dimasukkan ke dalam *database* dan juga melakukan proses enkripsi untuk menjaga kerahasiaan *password* dari petugas. Sementara itu, *endpoint* yang memiliki paling sedikit *response time* berada pada *requirement* nomor 3, yaitu untuk menampilkan data master jenis aset. Hal ini disebabkan karena hanya ada 2 tabel yang harus ditampilkan sehingga tidak memerlukan *response time* yang lama. Total *response time* yang berbeda-beda menunjukkan bahwa setiap pengembangan *endpoint* dipengaruhi oleh banyaknya hal yang harus diperhatikan untuk memenuhi rancangan *requirement* pada perancangan.

Pengujian ini dilakukan oleh penguji ahli, yaitu seorang *programmer back-end* yang juga sebagai staf pranata komputer di UPT TIK UNJ bernama Febrianto Widyoutomo, S.Pd. Berdasarkan hasil pengujian pada Tabel 4.1 – Tabel 4.41 yang dievaluasi oleh penguji ahli, dapat dilihat bahwa hasil pengujian berjalan lancar tanpa adanya kesalahan dari sisi fungsional sistem. Penulis juga melakukan pengujian pada 41 *endpoint* dan didapatkan hasil 41 *endpoint* berjalan sesuai fungsinya 100%. Dengan penggunaan spesifikasi personal computer yang berbeda, disimpulkan bahwa setiap *personal computer* memiliki perbedaan *response time*.

Penguji ahli memberi masukan untuk lebih memperhatikan organisasi penyimpanan *file* agar memudahkan pengembangan ke depan. Masukan dari penguji ahli juga berfungsi agar produk *web service* ini dapat diimplementasikan menjadi satu kesatuan *website* yang utuh dan bisa digunakan oleh *end user*.

Hasil dari penelitian ini adalah menghasilkan 41 *endpoint* berdasarkan *requirement* yang disetujui oleh UPT TIK. Hasil dari *web service* ini dapat diimplementasikan menjadi satu kesatuan *website* sistem manajemen aset UPT TIK UNJ dengan menggabungkan pengembangan di sisi *front-end*.

5. Kesimpulan dan Saran

Web service pada penelitian kali ini dikembangkan dengan menggunakan *framework Express.js* dan daftar *requirement* yang disetujui oleh UPT TIK UNJ. Daftar *requirement* menghasilkan *endpoint*. Satu buah *requirement* dapat menghasilkan satu atau lebih *endpoint*. Satu *endpoint* dapat dibentuk dari beberapa *requirement*.

Dalam pengembangan *web service* aplikasi manajemen aset UPT TIK UNJ, dihasilkan 41 buah *endpoint* yang dihasilkan berdasarkan pada daftar *requirement*. Kemudian masing-masing *endpoint* diuji dengan metode *unit testing* untuk mengetahui apakah masing-masing *endpoint* sudah berjalan sesuai fungsinya atau tidak. Hasil pengujian fungsional dengan metode *unit testing* menunjukkan bahwa *endpoint* yang telah dibuat berfungsi dengan baik 100%, sesuai dengan skenario proses yang telah dijabarkan di Tabel 3.19 – Tabel 3.60 dan layak digunakan untuk membangun sebuah aplikasi manajemen aset UPT TIK UNJ. Untuk pengembangan penelitian lebih lanjut, disarankan untuk melakukan beberapa hal berikut.

1. Membandingkan teknologi terbaru seperti bahasa pemrograman beserta *frameworknya* yang sesuai dengan perkembangan teknologi *web service*.
2. Penyesuaian terhadap *requirement* yang mungkin saja mengalami perubahan setiap waktunya.
3. Hasil *web service* dapat diimplementasikan menjadi satu kesatuan aplikasi dengan menggabungkan pengembangan di sisi *front-end*.

Daftar Pustaka:

- Anonymous. Pengenalan json. <https://json.org/json-id.html>. Diakses 18 Agustus 2020.
- Ahmad, I., Suwarni, E., Borman, R. I., Rossi, F., & Jusman, Y. (2021, October). Implementation of *Restful API Web services* Architecture in Takeaway Application Development. In 2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS) (pp. 132-137). IEEE.
- Duskarnaen, M. F., Praditya, H., & Adjie, H. (2019). Pengembangan *web service* penerimaan mahasiswa baru pascasarjana Universitas Negeri Jakarta tahun ajar 2017/2018. *Jurnal PINTER*, 3(1):38-44.
- Elu, A. M. (2013). Rancang Bangun Aplikasi Pendeteksian Vulnerability Structured Query Language (Sql) Injection Untuk Keamanan Website. *Jurnal Teknologi Informasi*, 7(22):111-124.
- Fajrin, R. (2017). Pengembangan sistem informasi geografis berbasis *node.js* untuk pemetaan mesin dan tracking engineer dengan pemanfaatan geolocation pada PT IBM Indonesia. *Jurnal Informatika*, 11(2):40-47.
- Indrawan, M. E., & Adil, A. (2017). Implementasi *Restful Web service One Chip Multi-Client* Untuk Mengoptimalkan Penjualan Pulsa All Operator. *Jurnal Matrik*, 16(2):11-16.
- Kristanto, T. (2016). Enterprise architecture planning untuk proses pengelolaan manajemen aset dengan *zachman framework*. *Jurnal Ilmiah Teknologi Sistem Informasi*, 2(2):98-104.
- Limbong, N. I. E. (2016). Manajemen Aset Daerah Kota Pekanbaru (Studi Kasus Mobil Dinas). *JOM FISIP*, 3(2):1-14.
- Mediana, D., & Nurhidayat, A. I. (2018). Rancang bangun aplikasi helpdesk (a-desk) berbasis web menggunakan *framework laravel* (studi kasus di PDAM Surya Sembada Kota Surabaya). *Jurnal Manajemen Informatika*, 8(2):75-81.
- Lolong, S., Egerly, W., & Harimisa, R. (2016). Sistem berbasis digital untuk penelitian di Universitas Klabat. *Cogito Smart Journal*, 2(2):264271.
- Nirgudkar, N., & Singh, P. (2017). The mean stack. *International Research Journal of Engineering and Technology (IRJET)*, 4(5):3237-3239.
- Nugroho, M. A., & Katardi, R. (2016). Analisis Kinerja Penerapan Container Untuk Load Balancing Web Server Pada Raspberry Pi. *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, 1(2):7-15.
- Nurhayati, A. N., Josi, A., & Hutagalung, N. A. (2017). Rancang bangun aplikasi penjualan dan pembelian barang pada Koperasi Kartika Samara Grawira Prambumulih. *Jurnal Teknologi dan Informasi*, 7(2):13-23.
- Paryati, R., & Wahyu, A. P. (2019). Sistem Inventori Manajemen Aset Berbasis Teknologi Informasi. *TEDC*, 13(3):276-282.
- Putra, M. G. L., & Putera, M. I. A. (2019). Analisis perbandingan metode soap dan rest yang digunakan pada *framework flask* untuk membangun *web service*. *SCAN*, 14(2):1-7.
- Rais, M., & Sirad, M. A. H. (2019). Inventory Information System Of Goods Using Codeigniter Framework. *Patria Artha Technological Journal*, 3(1):23-30.
- Siregar, H. F., Siregar, Y. H., & Melani. (2018). Perancangan aplikasi komik hadist berbasis multimedia. *Jurnal Teknologi Informasi*, 2(2):113-121.
- Styawan, B. A. (2018). Rancang bangun sistem informasi tempat pariwisata di Provinsi Jawa Timur berbasis

android. TEKNOLOGI, x(y):1-6.

- Syam, A., & Arie, J. S. (2018). Aplikasi rekam medis berbasis web (studi kasus praktek Dr. M. Najib Djalaluddin). *Prosiding Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, 7(1):28-36.
- Utomo, A. N., & Alfaridzi, M. (2018). *Perancangan sistem informasi pada percetakan CV Citra Kencana Jakarta Timur berbasis web*. *Jurnal Rekayasa Informasi*, 7(1):43-47.
- Utomo, T. A., Yuwono, B. D., & Amarrohman, F. J. (2017). *Aplikasi Sistem Informasi Geografis Berbasis Web dan Android Untuk Pemilihan Jalur Alternatif Menuju Tempatpariwisata (Studi Kasus: Kota Wisata Cibubur dan Jungleland, Kabupaten Bogor)*. *Jurnal Geodesi UNDIP*, 6(2):1-11.
- Widyoutomo, F. (2019). *Pengembangan web service modul mahasiswa pada sistem informasi akademik Universitas Negeri Jakarta [skripsi]*. Jakarta: Fakultas Teknik, Universitas Negeri Jakarta.