

PENGEMBANGAN *WEB SERVICE* SISTEM INFORMASI SKRIPSI DENGAN METODE *SCRUM* UNTUK PROGRAM STUDI PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER UNIVERSITAS NEGERI JAKARTA

M. Syafiq Roikhan Maulana¹, Hamidillah Ajie², Widodo³

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

^{2,3} Dosen Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

¹msyafiq.roikhanm@gmail.com, ²hamidillah@unj.ac.id, ³widodo@unj.ac.id

Abstrak

Skripsi menjadi syarat kelulusan mahasiswa Program Studi Pendidikan Teknik Informatika dan Komputer (PTIK) Universitas Negeri Jakarta (UNJ). Skripsi sebagai sarana penelitian yang dilakukan mahasiswa dengan bimbingan dosen dengan harapan hasil penelitian memiliki kualitas tinggi sehingga dapat diterapkan di lingkungan Program Studi PTIK. Saat ini, pengelolaan semua tahapan administratif skripsi masih dilakukan secara manual oleh admin program studi. Belum adanya sistem terintegrasi yang membantu keseluruhan proses administratif pelaksanaan skripsi, sehingga muncul sejumlah kendala. Selain itu, kebutuhan informasi yang berkaitan dengan skripsi tidak hanya dibutuhkan untuk kelulusan mahasiswa saja, tetapi sewaktu-waktu dibutuhkan untuk sistem yang lain. Sehingga dibutuhkan perangkat lunak layanan sistem informasi skripsi yang dapat memberikan informasi ke berbagai sistem yang didukung. Tujuan dari penelitian ini adalah untuk menghasilkan Web Service yang dapat digunakan untuk mengembangkan Sistem Informasi Skripsi untuk Program Studi PTIK UNJ yang memiliki sifat adaptif terhadap perubahan yang perlu dilakukan di masa mendatang. Pengembangan dilakukan menggunakan Metode Scrum, bahasa pemrograman yang digunakan adalah javascript, dibantu dengan framework ExpressJS, dan menghasilkan 69 endpoint. Pengujian dilakukan pada seluruh endpoint dengan menggunakan metode Unit Testing dengan bantuan tools Postman. Hasil pengujian menunjukkan bahwa endpoint berfungsi dengan baik dan layak digunakan sebagai sisi backend dari Sistem Informasi Skripsi Program Studi PTIK UNJ.

Kata kunci : Sistem Informasi Skripsi, Web Service, Metode Scrum, Javascript, Unit Testing.

1. Pendahuluan

Karya akhir kuliah merupakan istilah bagi perguruan tinggi yang merujuk pada suatu karya ilmiah berupa paparan hasil penelitian yang membahas suatu permasalahan yang ditemukan dalam bidang tertentu dengan menggunakan kaidah-kaidah penelitian yang berlaku berdasarkan jenis pendidikan yang ditempuh atau gelar yang akan didapatkan. Istilah Skripsi biasanya digunakan sebagai salah satu syarat kelulusan untuk mendapatkan gelar Sarjana (S-1), sedangkan istilah Tugas Akhir (TA) digunakan sebagai salah satu syarat kelulusan untuk mendapatkan gelar Ahli Madya (D-3). Dalam penyusunan karya akhir, masing-masing fakultas dapat memiliki pedoman penyusunan karya akhir untuk memudahkan mahasiswanya melakukan penelitian sesuai bidang yang ada di fakultas. Pada umumnya tahapan penyusunan karya akhir mahasiswa sama, yang menjadi perbedaan di dalam pedoman masing-masing fakultas adalah struktur penulisan karya akhir yang tentunya disesuaikan dengan disiplin ilmu yang dipelajari di masing-masing fakultas. Perubahan dan perbaikan dalam pedoman seringkali dilakukan dimaksudkan untuk meningkatkan kualitas penelitian, tidak terkecuali Fakultas Teknik yang terus melakukan perbaikan pedoman penyusunan karya akhir, saat ini Fakultas Teknik memiliki pedoman penyusunan karya akhir yang terbaru dikeluarkan pada tahun 2019 dan pedoman tersebut diterapkan oleh Program Studi yang ada di Fakultas Teknik, salah satunya Program Studi Pendidikan Informatika dan Komputer (PTIK).

Dalam penyusunan skripsi, Program Studi PTIK mengikuti pedoman penyusunan skripsi yang dikeluarkan oleh Fakultas Teknik. Saat ini pengelolaan semua tahapan administratif skripsi tersebut masih dilakukan secara

manual oleh seorang admin program studi. Belum adanya sistem terintegrasi yang membantu keseluruhan proses administratif pelaksanaan skripsi, sehingga muncul sejumlah kendala, seperti; hilangnya dokumen-dokumen yang berhubungan dengan proses penyusunan skripsi, hilangnya dokumen skripsi yang sudah selesai, kesulitan mengatur jadwal seminar maupun sidang skripsi, dan tidak adanya arsip skripsi sehingga mahasiswa yang akan memulai penyusunan karya akhir sulit mencari referensi dari skripsi yang sudah disetujui sebelumnya. Selain itu, kebutuhan informasi yang berkaitan dengan skripsi tidak hanya dibutuhkan untuk kelulusan mahasiswa saja, tetapi informasi tersebut sewaktu-waktu dibutuhkan untuk sistem yang lain, seperti; pendataan mahasiswa bimbingan dosen, akreditasi program studi PTIK, dan remunerasi dosen.

Dalam proses pengembangan perangkat lunak, perangkat lunak yang akan dibuat semestinya memenuhi kebutuhan pengguna, hal ini perlu diketahui pada awal pengembangan. Tim pengembang melakukan wawancara terhadap pengguna untuk mendapatkan daftar kebutuhan perangkat lunak, yang nantinya akan didiskusikan oleh tim pengembang untuk mulai dilakukan pengembangan. Tetapi pada faktanya, kebutuhan yang sudah didapatkan saat wawancara seringkali berubah saat pengembangan, hal ini dapat berakibat pada banyak aspek, salah satunya adalah bertambahnya waktu pengerjaan jika terdapat perubahan kebutuhan dan penambahan kebutuhan.

Berdasarkan permasalahan yang ada, maka diperlukan sebuah perangkat lunak sistem informasi skripsi yang dapat memberikan informasi ke berbagai sistem yang didukung dan dalam proses pengembangannya adaptif terhadap perubahan.

2. Dasar Teori

2.1. Sistem Informasi

Menurut Khoirunnisa (2017) mendefinisikan sistem informasi sebagai suatu proses menemukan, memproses, menganalisis, dan menyebarkan suatu informasi dengan tujuan tertentu. Hal ini berarti bahwa sistem informasi merupakan alur atau langkah-langkah memproses informasi mulai dari menemukan informasi sampai menyebarkan informasi sesuai dengan kebutuhan pengguna.

2.2. Skripsi

Menurut Suwita (2020) skripsi merupakan istilah bagi perguruan tinggi untuk merujuk pada suatu karya ilmiah berupa paparan hasil penelitian dari sarjana strata satu (S1) yang membahas suatu permasalahan yang ditemukan dalam bidang tertentu dengan menggunakan kaidah-kaidah penelitian yang berlaku. Skripsi adalah jalur penyelesaian studi yang menuntut mahasiswa menyusun suatu karya ilmiah berdasarkan hasil penelitian yang dilakukan untuk memecahkan suatu permasalahan secara ilmiah.

2.3. Website

Menurut Sitinjak Daniel Dido Jantce TJ & Suwita (2020) *website* merupakan layanan yang dapat digunakan oleh pemakai komputer terhubung ke internet baik berupa teks, gambar, suara maupun video yang interaktif dan dapat menghubungkan antar halaman dan diakses menggunakan *browser*. *website* adalah sekumpulan halaman yang memuat informasi yang dapat diakses kapanpun dan dimanapun melalui jaringan internet dan menggunakan *browser*.

2.4. Backend

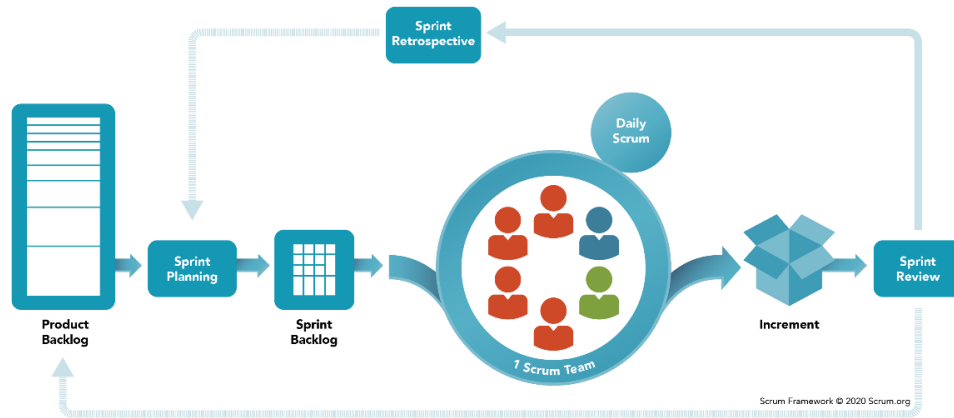
Pengembangan *website* saat ini, terdapat istilah *frontend* dan *backend*. istilah ini berkaitan dengan pemisahan dari segi interaksi dan tugas. Terjadinya pemisahan ini mempermudah pengembang dalam melaksanakan tugasnya sehingga lebih terspesialisasi. *Backend* adalah sebuah bagian dari aplikasi yang bertugas mengelola komunikasi data, logika, dan *server* dari sebuah aplikasi. *Backend* membangun aplikasi yang nantinya akan dipasangkan di *server* untuk mengirimkan data ke *browser* yang mengirimkan permintaan menggunakan protokol HTTP. Salah satu aplikasi *backend* yaitu *Web Service*.

Menurut Somya & Nathanael (2019), *Web Service* adalah suatu sistem perangkat lunak (*software*) yang dirancang untuk mendukung komunikasi dan *interoperabilitas* antar sistem dalam sebuah jaringan. Sedangkan menurut Bachriwindi *et al.* (2020), *Web Service* adalah antar muka yang mendeskripsikan layanan yang dapat diakses dalam jaringan menggunakan format standar untuk pertukaran pesan. Dari pendapat tersebut, dapat disimpulkan *Web Service* merupakan standar yang dapat digunakan untuk melakukan komunikasi atau pertukaran data antar aplikasi atau sistem.

2.5. Metodologi Pengembangan Agile

Metodologi pengembangan agile atau *agile software development* merupakan kumpulan metode pengembangan perangkat lunak yang berbasis perulangan pada proses pengembangannya atau sering disebut *iterative*. Menurut Pratasik & Rianto (2020), *agile* merupakan metodologi pengembangan perangkat lunak yang mengutamakan kesiapan untuk melakukan perubahan pada proses pengembangan perangkat lunak. Metode

pengembangan perangkat lunak yang bersifat *agile* antara lain; *Extreme Programming (XP)*, *Scrum*, *Feature Driven Development (FDD)*, dan masih banyak lagi. Menurut Al-Saqqa, dkk., (2020) Metode Agile lebih adaptif terhadap perubahan yang diminta, karena itu, selalu ada kolaborasi langsung antara pengembang dan pelanggan, metode ini juga lebih efisien untuk digunakan dalam proyek kecil dan menengah.



Gambar 2.1 Tahapan Pengembangan Scrum (Sumber: Scrum.org)

Scrum merupakan kerangka kerja dimana orang dapat mengatasi masalah adaptif yang kompleks, sementara secara produktif dan kreatif memberikan produk dengan nilai setinggi mungkin. *Scrum* memiliki lima tahapan dalam proses pengembangan, dimulai dari *Product Backlog*, *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*. Fokus utama dalam metode pengembangan *scrum* adalah *Sprint*. Dalam *Sprint*, *Scrum* team berusaha membuat kemajuan sebaik mungkin menuju tujuannya.

3. Metodologi

3.1. Tempat dan Waktu Penelitian

Penelitian dilakukan di Program Studi Teknik Informatika dan Komputer, Fakultas Teknik, Universitas Negeri Jakarta yang berlokasi di Gedung L1, Jalan Rawamangun Muka, RT.11 RW.14, Rawamangun, Pulo Gadung, Jakarta Timur, DKI Jakarta, 13220. Penelitian dilakukan pada bulan Desember 2021 hingga Juli 2022.

3.2. Alat dan Bahan Penelitian

Perangkat keras yang digunakan selama penelitian yaitu sebuah *notebook*. Perangkat lunak yang digunakan yaitu NodeJS versi 16.13.1 sebagai *web server*, Visual Studio Code versi 1.63.2 sebagai *text editor* dan *Postman* versi 9.9.0 sebagai *tools testing*.

Penelitian ini menggunakan bahan berupa hasil wawancara kepada Admin Program Studi PTIK dan Koordinator Program Studi PTIK serta studi literatur pada Buku Panduan Skripsi Fakultas Teknik tahun 2019. Sumber data tersebut kemudian dianalisis untuk menyusun kebutuhan pengguna terhadap perangkat lunak yang akan dikembangkan.

3.3. Teknik Pengumpulan Data

Pada penelitian ini, teknik pengumpulan data dilakukan dengan cara wawancara dan studi literatur. Pengumpulan data dilakukan dengan wawancara admin Program Studi PTIK yaitu pada bulan Januari 2022 dan Koordinator Program Studi PTIK pada bulan April 2022. Selanjutnya adalah melakukan studi literatur dengan melakukan pencarian dan pengambilan data terhadap referensi yang berkaitan dengan *Web Service* dan Sistem Informasi Skripsi yang bertujuan untuk memenuhi kebutuhan data dan informasi yang dibutuhkan peneliti. Setelah wawancara dan studi pustaka dilakukan maka data-data yang diperoleh diolah kembali untuk menentukan arah pengembangan *web* yang akan dibuat.

3.4. Teknik Analisis Data

3.4.1. Unit Testing

Unit Testing adalah pengujian yang dilakukan oleh pengembang terhadap setiap bagian perangkat lunak yang telah selesai dibuat dengan memastikan antara *input* dengan *output* agar sesuai dengan kebutuhan perangkat lunak. Bagian kecil yang dimaksud adalah *endpoint* yang merepresentasikan satu fungsi pada kebutuhan perangkat lunak.

Pengujian dilakukan pada tahapan *Sprint*, tepatnya setelah suatu *endpoint* telah selesai dibuat. Selama melakukan pengujian, akan digunakan sebuah tabel yang memetakan hasil dari pengujian *Web Service*. Tabel tersebut nantinya akan menjadi dokumentasi dari *Web Service* yang dihasilkan. Pengujian dilakukan dengan dua skenario yaitu skenario sukses dan skenario gagal.

4. Hasil dan Analisis

4.1. Deskripsi Hasil Penelitian

Penelitian ini menghasilkan *Web Service* sebanyak 69 buah *endpoint*, mengacu pada kebutuhan pengguna terhadap perangkat lunak. Pengembangan *endpoint* dilakukan menggunakan Metode *Scrum*.

4.2. Analisis Data Penelitian

4.2.1. Tahap Discovery Session (*Product Backlog*)

Data penelitian ini, selanjutnya dianalisis sehingga menghasilkan daftar kebutuhan pengguna yang berjumlah 69 buah. Kemudian, dilakukan perencanaan *endpoint* secara keseluruhan untuk merealisasikan daftar kebutuhan pengguna. Setelah itu didiskusikan kembali untuk menghasilkan *product backlog*. Sampel *product backlog* dapat dilihat pada tabel 4.1.

Tabel 4.1 *Product Backlog*

Kode	<i>Product Backlog Item</i>	Waktu (Jam)	Prioritas
R001	<i>Login</i>	5	Tinggi
R002	<i>Logout</i>	4	Tinggi

4.2.2. Tahap *Sprint*

4.2.2.1. Tahap *Sprint Planning*

Pada kegiatan *Sprint planning*, dilakukan proses identifikasi terhadap pekerjaan yang akan dilakukan selama satu *Sprint*. *Sprint planning* dilaksanakan internal oleh tim pengembang. Luaran yang dihasilkan oleh kegiatan *Sprint planning* berupa *Sprint backlog*. *Sprint backlog* ini yang akan menjadi acuan tim pengembang untuk melakukan satu *Sprint* kedepan.

Tabel 4.2 *Sprint Backlog*

Kode	<i>Product Backlog Item</i>	Waktu (Jam)	Prioritas
R001	<i>Login</i>	5	Tinggi
	Perancangan <i>endpoint</i>	1	
	Implementasi <i>endpoint</i>	2	
	Pengujian <i>endpoint</i>	1	
	Mendokumentasikan <i>endpoint</i>	1	
R002	<i>Logout</i>	4	Tinggi
	Perancangan <i>endpoint</i>	1	
	Implementasi <i>endpoint</i>	1	
	Pengujian <i>endpoint</i>	1	
	Mendokumentasikan <i>endpoint</i>	1	

4.2.2.2. *Daily Scrum*

Pada kegiatan *daily scrum*, tim pengembang melakukan pertemuan singkat setiap harinya untuk membahas progres dan kendala yang dialami pada pekerjaan pada hari sebelumnya dan pekerjaan yang akan dilakukan hari ini. Pertemuan dilaksanakan tidak lebih dari 15 menit.

Tabel 4.3 *Daily Scrum*

Hari/Tanggal	Evaluasi Pekerjaan	Perencanaan Pekerjaan
Senin, 13 Juni 2022	Penulisan nama <i>migration</i> menggunakan nama yang mendefinisikan fungsi dari <i>migration</i> tersebut agar mudah dipahami	Melakukan perencanaan, implementasi basis data, dan mempersiapkan data <i>dummy</i> untuk pengembangan
		Membuat <i>migration</i> basis data untuk mengimplementasikan basis data

Penggunaan <i>library javascript</i> yang sesuai dan lengkap dokumentasinya selama pengembangan	Melakukan perencanaan <i>endpoint login dan logout</i>
	Melakukan implementasi <i>endpoint login dan logout</i>
Pengujian dilakukan lebih detail untuk meminimalisir error yang tidak tertangani	Melakukan pengujian <i>endpoint login dan logout</i>
	Melakukan dokumentasi <i>endpoint login dan logout</i>

4.2.2.3. Perancangan *Endpoint*

Sebelum melakukan implementasi *endpoint*, diperlukan perancangan *endpoint* agar *endpoint* yang dihasilkan sesuai dengan kebutuhan perangkat lunak. Adapun yang perlu diperhatikan dalam perancangan *endpoint* yaitu HTTP *method* dan URI yang akan digunakan harus sesuai dengan *endpoint* yang akan dibuat. Hasil perancangan *endpoint* dapat dilihat pada tabel 4.4.

Tabel 4.4 Perancangan *Endpoint*

Kode	Kebutuhan Fungsional	<i>Endpoint</i>	HTTP <i>Method</i>
R001	<i>Login</i>	/auth/login	POST
R002	<i>Logout</i>	/auth/logout	POST

4.2.2.4. Implementasi *Endpoint*

Setelah melakukan perancangan *endpoint*, dilanjutkan proses implementasi *endpoint* menggunakan *framework* ExpressJS dan menggunakan JSON sebagai format pertukaran data. Dalam Implementasi *endpoint*, setiap *endpoint* direpresentasikan melalui *route*, yaitu kode yang menghubungkan antara URL dengan *controller*. Kode program dibuat menggunakan VSCode. Cuplikan kode *route* dapat dilihat pada gambar 4.1.

```

1  const router = require("express").Router();
2  const { check } = require("express-validator");
3  const AuthController = require("../controllers/auth.controller");
4  const ValidationMiddleware = require("../middlewares/validation.middleware");
5
6  router.post(
7    "/login",
8    [
9      check("email", "Email tidak boleh kosong").notEmpty(),
10     check("password", "Password tidak boleh kosong").notEmpty(),
11   ],
12   ValidationMiddleware.result,
13   AuthController.login
14 );
15 router.post(
16   "/logout",
17   [check("email", "Email tidak boleh kosong").notEmpty()],
18   ValidationMiddleware.result,
19   AuthController.logout
20 );
21
22 module.exports = router;

```

Gambar 4.1 Kode *AuthRoute*

4.2.2.5. Pengujian *Endpoint*

Pada pengujian *endpoint*, setiap *endpoint* yang sudah selesai dibuat akan diuji dengan dua skenario, yaitu; skenario berhasil dan skenario gagal. Kedua kondisi tersebut merupakan kondisi yang sesuai dengan kebutuhan perangkat lunak. Pengujian dilakukan menggunakan *tools* Postman. Cuplikan hasil pengujian dapat dilihat pada tabel 4.5.

Tabel 4.5 Hasil Pengujian *Endpoint*

Kode	Skenario	Respon	Waktu Proses	Sesuai Kebutuhan
R001	Sukses	{ "code": 200, "status": "OK", "message": "Berhasil Login", "data": {	687ms	Ya

	<pre> "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1dWlkIjoiaWkiNThiMzVkMTctOTkyZS00MzIxLTliYmMtNzE3MTljMGFjMzZmIiwiaWF0IjoiNjUyZWZpcVJvaWtoYW5NYXVsYW5hXzE1MTI2MTgwMDFAbWhzLnVuai5hYy5pZCIsImhhdCI6MTY1NTEuMDY1OCwiZXhwIjozNjU1MTk3MDE0fQ.sQJC1yRSs9LYUe-c7gysdAXGJHFwPj2Nb_h_BtA_ha0" } } </pre>		
Gagal	<pre> { "code": 401, "status": "Unauthorized Request", "message": "Email atau password salah" } </pre>	154ms	Ya

4.2.2.6. Dokumentasi Endpoint

Proses dokumentasi *endpoint* merupakan proses yang sangat penting setelah proses pembuatan *endpoint*. Dokumentasi ini bermanfaat untuk tim pengembang dan pengguna *Web Service* nantinya. Bagi pengembang, dokumentasi membantu dalam melakukan pembaharuan *endpoint* nantinya. Bagi pengguna, dokumentasi membantu dalam penggunaan *endpoint* dalam integrasi dengan aplikasi lain. Dokumentasi *endpoint* dapat dilakukan dengan menyimpan setiap *respon* ketika pengujian *endpoint*. Hasil dokumentasi *endpoint* selengkapnya dapat dilihat pada tautan [Thesis Information System \(getpostman.com\)](https://getpostman.com).

4.2.2.7. Sprint Review

Pada kegiatan *Sprint review* ini, dilakukan peninjauan terhadap tujuan *Sprint*, *product backlog item* yang sudah dikerjakan, demonstrasi produk yang sudah dikerjakan, serta menerima umpan balik dari *stakeholders*. Semua poin yang telah direncanakan pada *Sprint planning* selesai tepat waktu dan sesuai dengan kebutuhan pengguna. Adapun untuk *Sprint* berikutnya diharapkan tetap tepat waktu dalam menyelesaikan *Sprint backlog* karena *framework* dan *library* sebagian besar sudah dikerjakan pada *Sprint* pertama dengan begitu akan memudahkan proses pengembangan pada *Sprint* berikutnya. Kemudian jikalau diperlukan adanya perubahan yang menyangkut produk hasil *Sprint* pertama, harus dipastikan perubahan tersebut tidak menyebabkan gagal sistem pada produk hasil *Sprint* pertama.

4.2.2.8. Sprint Retrospective

Selama *Sprint* pertama ini, proses pengembangan berjalan lancar hanya saja memerlukan waktu lebih dibanding perencanaan sebelumnya karena perlu mempersiapkan *framework*, *library*, *service*, dan *helper* selama proses pengembangan. Hal ini diperlukan agar kode dapat digunakan kembali (*reusable*) dan mudah dibaca. Namun, *Sprint backlog* dapat diselesaikan seluruhnya. Pada *Sprint* berikutnya, perlu mengoptimasi waktu dengan menyesuaikan *Sprint backlog* yang akan dikerjakan agar beban kerja seimbang setiap harinya.

4.2.2.9. Product Increment

Selama *Sprint* pertama ini, dihasilkan produk yang sesuai dengan tujuan ketika *Sprint planning*. Semua *Sprint backlog item* telah selesai dibuat semua.

4.3. Pembahasan

Pengembangan *Web Service* ini dilakukan dengan menggunakan metode pengembangan *scrum*. Selama *Sprint*, tim pengembang merealisasikan daftar kebutuhan perangkat lunak yang ada pada *product backlog*.

Pengujian dilakukan oleh penulis sendiri menggunakan metode *Unit Testing*. Berdasarkan hasil pengujian pada tabel 4.5, dapat dilihat bahwa *endpoint* yang dihasilkan sudah berjalan baik dan sesuai kebutuhan. Kemungkinan kesalahan (*error*) sudah ditangani dengan baik dengan membuat pesan kesalahan yang merepresentasikan kesalahan yang terjadi sehingga dapat memudahkan pengguna dalam menangani kesalahan yang dihadapi.

Proses pengembangan dilakukan menggunakan alat dan bahan yang sudah dijelaskan sebelumnya. Sehingga hasil pengujian dengan menggunakan alat dan bahan yang berbeda akan menghasilkan hasil pengujian yang berbeda secara performa, namun tetap sama dari segi fungsional.

4.4. Aplikasi Hasil Penelitian

Hasil dari penelitian ini merupakan *Web Service* yang terdiri dari 69 *endpoint* sesuai dengan kebutuhan yang dapat digunakan untuk mengembangkan Sistem Informasi Skripsi untuk Program Studi PTIK UNJ. *Web Service* dari hasil penelitian ini dapat diintegrasikan dengan *User Interface* untuk mengembangkan Sistem Informasi Skripsi untuk Program Studi PTIK UNJ secara utuh.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Pengembangan *Web Service* Sistem Informasi Skripsi untuk Program Studi PTIK UNJ dilakukan dengan metode pengembangan *scrum*. Bahasa pemrograman yang digunakan yaitu *javascript* dengan bantuan *runtime* NodeJS yang bertindak sebagai server untuk mengeksekusi dan *framework* ExpressJS. Proses pengujian dan dokumentasi dilakukan dengan menggunakan *too backend ls Postman*.

Metode *Scrum* yang digunakan sebagai metode pengembangan dinilai cocok untuk pengembangan *Web Service*. Hal ini dikarenakan metode *scrum* memiliki sifat yang adaptif terhadap perubahan di tengah proses pengembangan. Kebutuhan perangkat lunak direpresentasikan ke dalam bentuk *product backlog* dan digunakan sebagai acuan dalam perencanaan dan implementasi *endpoint* yang akan dikembangkan. Selama proses pengembangan berlangsung dilakukan pengujian terhadap *endpoint* yang telah selesai dikembangkan untuk mengetahui apakah *endpoint* sudah berfungsi dengan baik serta memberikan respon yang sesuai. Setelah itu *endpoint* didokumentasikan untuk mempermudah pengguna menggunakan *endpoint*.

Pengujian dilakukan dengan metode *Unit Testing*, yaitu menguji unit-unit kecil *Web Service* secara fungsional dengan memperhatikan kondisi-kondisi yang memungkinkan akan terjadi. Unit-unit kecil yang dimaksud adalah *endpoint*. Hasil pengujian dapat dilihat di tabel 4.5. Dari tabel pengujian dapat dilihat bahwa keseluruhan *endpoint* sudah berfungsi dengan baik.

5.2 Saran

Saran yang dapat penulis sampaikan dari penelitian ini sebagai berikut:

1. Mengikuti tahapan yang ada pada metode pengembangan *scrum* agar perubahan dapat ditangani dengan baik.
2. Membaca dokumentasi dari bahasa pemrograman dan *framework* yang digunakan agar tidak terjadi kesalahan sintaks saat mengimplementasikan suatu produk.
3. Implementasikan metode *clean code* agar kode yang dibuat mudah dibaca oleh pengembang sendiri di masa depan atau orang lain yang akan mengembangkan produk yang dikembangkan oleh pengembang berikutnya.
4. Dokumentasikan seluruh *endpoint* dengan baik agar memudahkan pengguna dalam menggunakannya.
5. Menjaga pembaruan teknologi yang digunakan pada *Web Service* dengan mencari informasi terhadap teknologi yang berkaitan dengan pengembangan *Web Service*.

Daftar Pustaka:

- Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). *Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11)*.
- Bachriwindi, A. et al. (2020) 'Pengembangan *Web Service* Sistem Informasi Boarding School', *Jurasik (Jurnal Riset Sistem Informasi dan Teknik Informatika)*, 5(1), p. 36. doi: 10.30645/jurasik.v5i1.167.
- Khoirunnisa (2017) 'Jurnal Manajemen Pendidikan Jurnal Manajemen Pendidikan', *Jurnal Manajemen Pendidikan*, 3, pp. 479–487.
- Pratasik, S. and Rianto, I. (2020) 'Pengembangan Aplikasi E-DUK Dalam Pengelolaan SDM Menggunakan Metode Agile Development', *CogITO Smart Journal*, 6(2), p. 204. doi: 10.31154/cogito.v6i2.267.204-216.
- Sitinjak Daniel Dido Jantce TJ, M. and Suwita, J. (2020) 'Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang', *Ipsikom*, 8(1), pp. 1–19.
- Somya, R. and Nathanael, T. M. E. (2019) 'Pengembangan Sistem Informasi Pelatihan Berbasis *Web* Menggunakan Teknologi *Web Service* Dan *Framework Laravel*', *Jurnal Techno Nusa Mandiri*, 16(1), pp. 51–58. doi: 10.33480/techno.v16i1.164.
- Suwita, F. S. (2020) 'Pengembangan Sistem Informasi Tugas Akhir dan Skripsi (SIMITA)', *Jurnal Teknologi dan Informasi (JATI)*, 10, pp. 71–82. doi: 10.34010/jati.v10i1.