

PENGEMBANGAN WEB SERVICE APLIKASI UJIAN ESSAY ONLINE PADA PRODI PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER UNIVERSITAS NEGERI JAKARTA

Muhammad Azhar Hermawan¹, 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

¹ MuhammadAzharHermawan_1512617038@mhs.unj.ac.id, ² hamidillah@unj.ac.id,

³ widodo@unj.ac.id

Abstrak

Pelaksanaan ujian pada beberapa mata kuliah di Pendidikan Teknik Informatika dan Komputer (PTIK) Universitas Negeri Jakarta dilaksanakan secara online. adapun instrumen pelaksanaannya menggunakan aplikasi dari pihak ketiga yang bervariasi dan sesuai dengan pilihan tiap dosen. Banyaknya variasi dari aplikasi yang digunakan memungkinkan data pelaksanaan ujian ataupun data hasil ujian pada setiap ujian tidak terintegrasi secara maksimal. Oleh karena itu dibutuhkan sebuah solusi berupa aplikasi ujian essay yang dikembangkan sesuai dengan kebutuhan ujian pada Prodi PTIK dan dapat diintegrasikan dengan sistem lain seperti Sistem Informasi Akademik (SIKAD). Tujuan dari penelitian ini adalah untuk menghasilkan web service yang dapat digunakan untuk mengembangkan aplikasi ujian essay yang memiliki sifat adaptif terhadap perubahan kebutuhan di masa mendatang. Pengembangan dilakukan dengan menggunakan konsep RESTful API, metode Spiral, framework ExpressJS dan menghasilkan 229 endpoint. Pengujian web service dilakukan terhadap seluruh endpoint dengan menggunakan metode unit testing dan bantuan tools Postman. Berdasarkan hasil pengujian dapat disimpulkan bahwa seluruh endpoint telah berfungsi dengan baik dan layak digunakan sebagai sisi back-end pada aplikasi ujian essay online.

Kata kunci : Aplikasi Ujian Essay Online, Spiral, Unit Testing, Web Service, Endpoint

1. Pendahuluan

Pada program studi (Prodi) Pendidikan Teknik Informatika dan Komputer di Universitas Negeri Jakarta, penggunaan ujian *essay* sebagai media evaluasi sudah umum digunakan oleh tenaga pendidik di sana. Penggunaan ujian *essay* berbasis teks biasanya digunakan pada bidang mata kuliah yang sifatnya melihat terhadap pandangan pribadi dan pikiran relatif peserta didik atau disebut subjektif, seperti filsafat, kewarganegaraan, pengantar sistem dan teknologi informasi. ujian *essay* digunakan juga untuk mata kuliah yang sifatnya melihat terhadap keadaan yang sebenarnya tanpa dipengaruhi pendapat atau pandangan pribadi atau disebut objektif.

Pada pelaksanaannya ujian yang diselenggarakan secara konvensional memerlukan kertas yang digunakan untuk pengadaan soal ujian, penilaian ujian, dan lain sebagainya. hal ini dapat menyebabkan borosnya penggunaan kertas dan kurangnya efisiensi biaya yang diperlukan untuk mempersiapkan dokumen ujian. Selain itu terkadang lembar soal yang dicetak tidak mencukupi sehingga menghambat proses pengerjaan ujian. waktu yang diperlukan untuk mengoreksi ujian secara konvensional memakan waktu yang lama, bahkan terkadang proses 2 pengoreksian jawaban harus dilakukan di luar jam kerja. dapat dikatakan, waktu merupakan faktor yang perlu diperhitungkan di sini. adapun banyak dari dosen Prodi Pendidikan Teknik Informatika dan Komputer sebelum masa *work from home*, masih menggunakan sistem ujian secara manual, sehingga memerlukan proses pengoreksian dan penilaian yang lebih lama.

Penggunaan aplikasi berbasis web pada implementasi ujian *essay online* memberikan banyak manfaat terhadap kesuksesan pengembangannya. Aplikasi ujian *online* berbasis web disebut juga dengan *e-Exam*, atau *electronic exam*. salah satu keunggulan penggunaan *website* dalam penerapan *e-Exam* antara lain adalah faktor portabilitasnya. *portabilitas* di sini berarti aplikasi yang dikembangkan dengan basis web mudah untuk diakses melalui perangkat apa saja asalkan ada koneksi internet. *web-based application* atau secara singkat *web-app* adalah istilah yang diberikan untuk aplikasi tersebut. selebihnya aplikasi berbasis web menjanjikan waktu

pengembangan aplikasi yang terlampaui lebih cepat dibandingkan dengan aplikasi *native*.

Web service mempunyai artian layanan web, sesuai dengan namanya, *web service* memberikan layanan pertukaran data antara *client-side* dan *server-side*. *Web service* dapat diimplementasikan menggunakan arsitektur REST (*Representational State Transfer*) yang pertama kali diperkenalkan oleh Fielding dalam disertasinya mengenai arsitektur perangkat lunak berbasis jaringan (Fielding, 2000). Pendekatan ini menangani struktur data dan keadaan transfernya secara efisien melalui protokol HTTP (Battle & Benson, 2008; Richardson & Ruby, 2007). Penggunaan *web service* berbasis REST (RESTful) sangat cocok sebagai *backend* aplikasi karena format data JSON yang ringan (Fauziah, 2014; Masse, 2011). *Web service* juga dapat diimplementasikan hanya di *server* saja, yang mana *webservice* dapat istilahnya “mendengarkan” *request* dari berbagai *client* melalui *port* yang ada di suatu jaringan, dan secara langsung melayani permintaan *client* tanpa implementasi servisnya secara langsung. *Web-service* menggunakan prinsip modularitas, di mana komponen-komponen perangkat lunaknya dapat diimplementasikan oleh aplikasi lain. *Web service* berhubungan langsung dengan *database* dan *business-logic*, *database* adalah sebuah penyimpanan informasi atau data yang terorganisasi dan terstruktur, yang biasanya disimpan di dalam sebuah sistem komputer. sementara, *business-logic* adalah serangkaian peraturan atau algoritma yang menangani pertukaran informasi antara *client* dan *database*. dengan kedua komponen tersebut, *webservice* dapat menjalankan serangkaian *business-logic* pada tingkatan *database* yang hasilnya bisa ditampilkan ke *front-end* atau pada *client* yang memintanya.

Penelitian ini mempertimbangkan tenaga pendidik yang menggunakan ujian *essay* sebagai media evaluasinya di kampus Universitas Negeri Jakarta terutama di Prodi Pendidikan Teknik Informatika dan Komputer sebagai narasumbernya. dengan menggunakan informasi yang didapat dari dosen di Prodi Pendidikan Teknik Informatika dan Komputer, diharapkan dapat tercapai beberapa kriteria pembuatan *web-service* aplikasi ini yang sesuai dengan kebutuhan dosen Prodi Pendidikan Teknik Informatika dan Komputer. banyak dari *web service* atau API untuk aplikasi ujian *essay online* yang ada sekarang ditujukan untuk kalangan akademisi pada umumnya. akan lebih bagus lagi jika *web-service* tersebut sudah disesuaikan dengan kebutuhan konsumen utamanya, dalam hal ini yaitu sesuai dengan kebutuhan pengguna pada Prodi Pendidikan Teknik Informatika dan Komputer.

Dengan demikian, berdasarkan pentingnya kegunaan sebuah aplikasi ujian *online essay* yang menyediakan berbagai fitur dengan lengkap, terlebih di era *study from home* seperti saat ini, Maka pengembangan aplikasi ujian *essay online* yang memiliki fitur lengkap, ringan dalam hal memori, mudah dioperasikan, dan sesuai dengan kebutuhan Prodi Pendidikan Teknik Informatika dan Komputer perlu dikembangkan agar mempermudah proses evaluasi pembelajaran, dan pengembangan *web service* dengan menggunakan metode pengembangan spiral dan terpisah dari keseluruhan aplikasi berbasis web tersebut menjadi faktor terbaik dalam mempercepat proses pengembangan. Maka, pada skripsi ini diangkatlah judul “*Pengembangan Web Service Aplikasi Ujian Essay Online Pada Prodi Pendidikan Teknik Informatika Dan Komputer Universitas Negeri Jakarta*“. harapannya agar *web service* tersebut dapat berguna bagi pihak-pihak yang ingin mengembangkan aplikasi ujian terutama ujian *essay online* berbasis web.

2. Dasar Teori

2.1. Ujian Essay

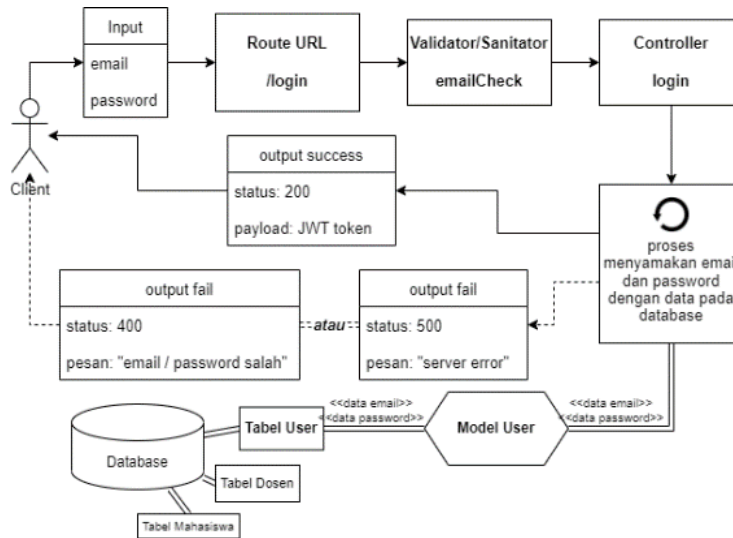
Ujian adalah cara atau prosedur yang perlu ditempuh dalam rangka pengukuran dan penilaian di bidang pendidikan, yang berbentuk pemberian tugas atau serangkaian tugas baik berupa pertanyaan – pertanyaan (yang harus dijawab) oleh tester, sehingga dapat menghasilkan nilai yang melambangkan tingkah laku atau prestasi tester; Sudijono, (2010) diacu dalam Nasril & Saputra, (2016). sementara *essay* menurut Kemdikbud, (2021) adalah karya tulis atau karangan ilmiah yang dalamnya memaparkan tentang masalah tertentu dari sudut pandang pribadi penulis dengan disertai data yang valid. Dari sana dapat disimpulkan bahwasanya ujian *essay* merupakan instrumen yang dipakai untuk menguji atau mengukur kemampuan seseorang dengan format karya tulis atau karangan ilmiah sebagai dasar evaluasi masa pembelajaran.

2.2. Aplikasi Web

Menurut Rerung, (2018) “Aplikasi Web merupakan aplikasi yang berjalan di berbagai web browser yang memberikan kemudahan-kemudahan seperti pada akses informasi dan setup server”.

2.3. Web Service

Web-service adalah sekumpulan fungsionalitas yang dapat diakses melalui *internet protocol (IP) standard* dan banyak digunakan oleh aplikasi sistem yang terdistribusi secara heterogen baik dari sistem operasi maupun model objek menurut Fauziah, (2014). Berikut alur *web service endpoint/login* dapat dilihat pada gambar 2.1.



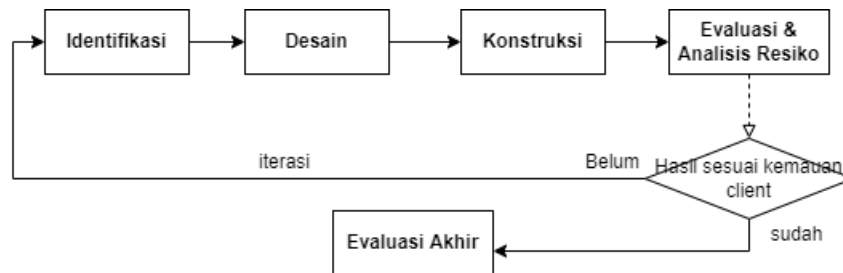
Gambar 2.1 Alur Web Service endpoint /login

2.4. RESTful Web Service

RESTful web service adalah web service yang menggunakan standar protokol dan method dari HTTP yang dapat diakses dengan mudah dengan web-browser melalui URL atau disebut juga endpoint menurut (Kurniawan, 2014).

2.5. Metode Pengembangan Spiral

Metode pengembangan spiral merupakan salah satu metode pengembangan perangkat lunak yang menggabungkan sifat iterative prototyping dengan aspek yang terkontrol dan sistematis dari model waterfall. (Pressman, 2011). Alur pengembangan spiral dapat dilihat pada gambar 2.2 berikut.

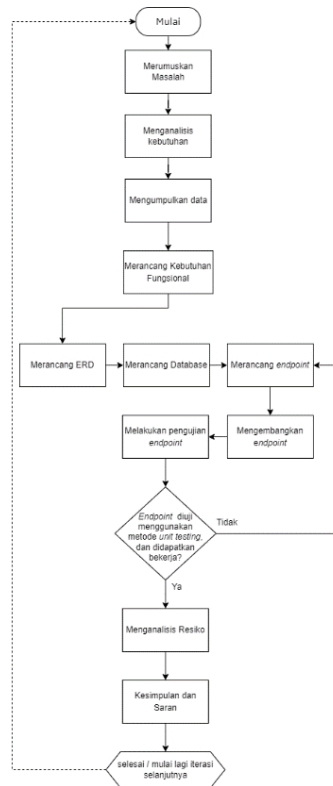


Gambar 2.2 Alur Pengembangan Spiral (modifikasi dari Spiral R. Pressman)

3. Metodologi

3.1. Diagram Alir Penelitian

Metodologi Penelitian ini dilakukan dengan menggunakan metode Spiral yang menggabungkan sifat iterative prototyping dengan aspek terkontrol dan sistematis dari model waterfall (Boehm & Hansen, 2001; Pressman, 2011; Sommerville, 2011). Tahapan analisis dan perancangan sistem mengacu pada standar pemodelan perangkat lunak modern untuk memastikan kebutuhan pengguna terpenuhi (Satzinger dkk., 2000). Untuk pemodelan basis data dan alur sistem, digunakan diagram UML yang sesuai dengan kaidah rekayasa perangkat lunak terstruktur (Rosa & Shalahudin, 2014). Pengembangan kode program dilakukan menggunakan teknologi Node.js dengan framework ExpressJS yang terbukti efisien untuk pengembangan API (Doglio, 2015). Maka metode penelitian alir yang disesuaikan dengan metode tersebut dapat dilihat pada gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.2. Instrumen Pengumpulan Data

Proses penelitian diawali dengan memberikan kuesioner kepada empat dosen program studi PTIK UNJ secara acak. Jenis pertanyaan terbuka ditawarkan sebagai bahan pengumpulan informasi awal. Kuesioner ini digunakan pada tahap analisis dengan tujuan untuk mendapatkan karakteristik tiap pengguna.

Kuesioner selanjutnya dilakukan untuk menentukan kebutuhan fungsional aplikasi ujian esai berbasis web. Kuesioner diberikan kepada tiga dosen program studi PTIK UNJ untuk mendapatkan informasi yang komprehensif tentang fitur-fitur pada aplikasi ujian *essay* pada umumnya atau yang diharapkan ada. Pertanyaan kuesioner tersebut terkait dengan kebiasaan dosen saat melakukan proses ujian esai, dan komentar serta ekspektasi terhadap aplikasi ujian yang akan dibangun.

3.3. Teknik Analisis Data

Analisis data dalam pengujian ini dilakukan dengan menggunakan metode pengembangan spiral. Data yang didapat dari kuesioner awal dan lanjutan diolah dalam proses tahapan pertama yaitu tahapan identifikasi. pada tahapan identifikasi data hasil kuesioner dianalisis menjadi sebuah daftar kebutuhan fungsional yang nantinya menjadi data primer bagi tahapan rancangan.

Tahapan rancangan meliputi perancangan *user story*, ERD (*Entity Relationship Diagram*), *Database*, dan *Endpoint*. Tahapan selanjutnya adalah tahap konstruksi, yaitu pembuatan prototipe *web service*, menggunakan format MVC (*Model View Controller*) sebagai basisnya untuk proyek penelitian ini. Dan terakhir melakukan evaluasi terhadap iterasi yang sedang berjalan dan menganalisis risiko pada iterasi tersebut.

Iterasi dinyatakan selesai jika hasil prototipe memenuhi kebutuhan dan kemauan *client*. untuk *client* dari proyek ini adalah gabungan dari dosen Prodi Teknik Informatika dan Komputer Universitas Negeri Jakarta, kolega peneliti, dan peneliti sendiri.

4. Hasil dan Analisis

4.1. Deskripsi Hasil Penelitian

Penelitian ini dimulai dari bulan Juli 2021 – Mei 2022 dan dilaksanakan secara *online*. Dari hasil kuesioner penelitian awal dan lanjutan yang telah dilakukan sebelumnya kepada beberapa dosen, dilakukan analisis untuk membuat daftar kebutuhan fungsional awal yang selanjutnya digunakan pada iterasi pertama spiral. selanjutnya, dikembangkanlah 3 prototipe *web service* aplikasi ujian *online*. Produk prototipe akhir yang dikembangkan menghasilkan 229 buah *endpoint*.

4.2. Analisis Data Penelitian

Terdapat beberapa tahapan dalam proses analisis data. Tahapan-tahapan tersebut mengacu kepada tahapan pengembangan perangkat lunak pada metode pengembangan spiral. Terdapat 3 iterasi pada tiap tahapan, di mana iterasi final menghasilkan produk jadi dan tidak perlu dilakukan lagi analisis risiko pada tahapan akhir iterasi terakhir. *Workflow spiral* pengembangan *web service* dari tahun 2021-2022 dapat dilihat pada tabel 4.1

Tabel 4.1 *Workflow Spiral* pengembangan *web service* 2021 – 2022

Iterasi	Bulan	Januari	Februari	Maret	April	Mei	Juni	Juli	Agustus	September	Oktober	November	Desember	Tahun
Pertama-Kedua													2021	
Kedua														2022
Ketiga														

4.2.1. Tahap Identifikasi

Data pada tahap ini bersumber dari hasil pengumpulan data berupa kuesioner awal dan lanjutan.

4.2.1.1. Iterasi Pertama

Fitur-fitur yang teridentifikasi pada tahap identifikasi pertama berjumlah total 65 fitur. Dibagi berdasarkan dengan *role* tiap *user*. Dengan jumlah fitur untuk *user* admin sejumlah 15, fitur untuk *role All User* (semua *role user* bisa menggunakan fitur-fitur tersebut) sejumlah 10, fitur untuk *user* dosen sejumlah 25, dan untuk *user* mahasiswa sejumlah 15.

4.2.1.2. Iterasi Kedua

Pada tahap identifikasi iterasi ke-2 memiliki total 77 fitur. dibagi berdasarkan dengan *role* tiap *user* dan juga pada tahap ini diidentifikasi fitur non-fungsional (fitur keseluruhan sistem). Dengan jumlah fitur untuk *user* admin sejumlah 25, fitur untuk *role All User*(semua *role user* bisa menggunakan fitur-fitur tersebut) sejumlah 15, fitur untuk *user* dosen sejumlah 22, dan untuk *user* mahasiswa sejumlah 12, dan fitur non-fungsional sejumlah 3.

4.2.1.3. Iterasi Ketiga

Pada tahap identifikasi iterasi ke-3 memiliki total 101 fitur. tabel kebutuhan dapat dilihat pada Lampiran 10. dibagi berdasarkan dengan *role* tiap *user*. Dengan jumlah fitur untuk *user* admin sejumlah 32, fitur untuk *role All User* (semua *role user* bisa menggunakan fitur-fitur tersebut) sejumlah 14, fitur untuk *user* dosen sejumlah 22, dan untuk *user* mahasiswa sejumlah 12, dan terdapat tambahan *role* sistem yang bertujuan untuk fitur automasi, memiliki sebanyak 6 *endpoint*, *role* super-admin bertujuan untuk memonitor kinerja admin, sebanyak 9 *endpoint*. Kebutuhan fungsional iterasi ketiga dapat dilihat pada tabel 4.2 berikut.

Tabel 4.2 Kebutuhan Fungsional iterasi ketiga

No.	Kebutuhan	User	Jenis Kebutuhan
1.	Mendaftarkan <i>user</i>		
2.	Menampilkan daftar <i>user</i>		
3.	Menampilkan data <i>user</i> (termasuk admin)	Admin	Kebutuhan Fungsional
4.	Mengubah data <i>user</i>		
5.	Menghapus data <i>user</i>		

4.2.2. Tahap Desain

4.2.2.1. Iterasi Pertama

Iterasi pertama tahap desain menghasilkan sebanyak 43 skenario *user story* yang terbagi ke dalam 4 *role user* yaitu *All User*, Dosen, Mahasiswa, dan Admin. *user story* merupakan suatu cara pengembang aplikasi memperjelas fitur-fitur yang seorang *user* aplikasi gunakan dari sudut pandang *role user* tersebut. Rancangan *endpoint* iterasi pertama menghasilkan sebanyak 75 *endpoint* yang terbagi ke dalam 4 *role user*. Rancangan ERD (*Entity Relationship Diagram*) menghasilkan sebanyak 13 Tabel *database*. Rancangan *database* menghasilkan desain sebanyak 13 tabel, dengan jumlah tabel utama sebanyak 9 tabel dan tabel irisan (*many-to-many*) yang diawali dengan Rel_... sebanyak 4 tabel.

4.2.2.2. Iterasi Kedua

Rancangan *user story* iterasi kedua menghasilkan sebanyak 53 skenario, semuanya terbagi ke dalam 5 *role user* (+*role* sistem) dari iterasi sebelumnya. Rancangan *endpoint* iterasi kedua menghasilkan sebanyak 165

10	updated_at	DATE	Index	Null
----	------------	------	-------	------

4.2.3. Tahap Konstruksi

4.2.3.1. Iterasi Pertama

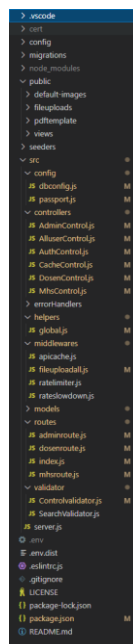
Project ini memakai format MVC (*Model View Controller*). *Model* merupakan representasi dari tabel pada *database* dalam proyek program. *Controller* merupakan tempat logika program dikembangkan, sesuai dengan kriteria data pada *Model*. *File* yang dihasilkan pada iterasi pertama untuk *model* sejumlah 23 *file* ekstensi .js, dan *controller* sejumlah 5 *file* untuk logika Admin, Dosen, Mahasiswa, *Authentication*, dan Index, yang berekstensi .js. *View* tidak digunakan karena fokus pengembangan hanya pada *backend* saja atau pada *web service* saja.

4.2.3.2. Iterasi Kedua

Iterasi kedua menghasilkan sejumlah 28 *file model* berekstensi .js. sejumlah 5 *file controller* berekstensi .js, jumlah *controller* sama seperti iterasi sebelumnya.

4.2.3.3. Iterasi Ketiga

Iterasi ketiga menghasilkan sejumlah 28 *file model* berekstensi .js, sama seperti iterasi sebelumnya, dapat dilihat pada gambar 4.2 berikut. Sejumlah 6 *file controller* berekstensi .js, bertambah 1 untuk *controller caching* aplikasi. Untuk kode program lengkapnya dapat dilihat pada tautan berikut: [Github-AplikasiUjianEssayPTIK](#)



Gambar 4.2 rancangan ERD iterasi ketiga

4.2.4. Tahap Evaluasi dan Analisis Risiko

4.2.4.1. Iterasi Pertama

Iterasi pertama menghasilkan *prototipe web service* dengan *database* sejumlah 13 tabel, *endpoint* sejumlah 75 *endpoint*, dan *file* kode sejumlah 44 berekstensi .js dan 1 .json

Analisis Risiko memunculkan hasil yang tinggi terhadap 6 *endpoint*, 2 terkait fitur Kerja sama dengan *frontend*, 1 terkait penilaian otomatis, 1 terkait penambahan gambar pada *excel*, dan 2 terkait pemasukan gambar ke aplikasi oleh *user*.

4.2.4.2. Iterasi Kedua

Iterasi pertama menghasilkan *prototype web service* dengan *database* sejumlah 13 tabel, *endpoint* sejumlah 75 *endpoint*, dan *file* kode sejumlah 44 berekstensi .js dan 1 .json

Analisis Risiko memunculkan hasil yang tinggi terhadap 6 *endpoint*, 2 terkait fitur Kerja sama dengan *frontend*, 1 terkait penilaian otomatis, 1 terkait penambahan gambar pada *excel*, dan 2 terkait pemasukan gambar ke aplikasi oleh *user*. Dapat dilihat pada tabel 4.6 dan 4.7 berikut.

Tabel 4.6 Risk Severity iterasi ke-2

No.	Risiko	Kategori	Kategori	Risk
-----	--------	----------	----------	------

<i>Endpoint</i>		<i>likelihood</i>	<i>impact</i>	<i>Severity</i>
5.	Merefresh <i>token</i> , implementasi jenis <i>refresh</i> sesuai <i>standard OAuth</i> , apakah menggunakan <i>refresh</i> biasa atau <i>token rotation</i> .	<i>High</i>	<i>High</i>	<i>High</i>

Tabel 4.7 Risk Respons iterasi ke-2

No. <i>Endpoint</i>	Risiko	Risk Severity	Risk Respond	Kesimpulan
5.	Merefresh <i>token</i> , implementasi jenis <i>refresh</i> sesuai <i>standard OAuth</i> , apakah menggunakan <i>refresh</i> biasa atau <i>token rotation</i> . Bagaimana cara menyatukan logika <i>token rotation oauth</i> dengan <i>passport.js</i>	<i>High</i>	<i>Accept</i> , perlu mempelajari lebih lanjut tentang sistem <i>token rotation</i> dan <i>local strategy passport.js</i>	Mempelajari sistem <i>token rotation</i> dan implementasi pada <i>local strategy passport.js</i>

4.2.4.3. Iterasi Ketiga

Iterasi Ketiga atau iterasi final, menghasilkan sejumlah 229 *endpoint* yang ke semuanya diuji menggunakan metode *unit testing*, dan didapatkan ke semuanya berjalan dengan baik dan semestinya. Dapat dilihat pada tabel 4.8 berikut.

Tabel 4.8 unit testing pada *endpoint* “/v1/login” ber-method GET, untuk role All User

Skenario	Respons	Sesuai Kebutuhan	Waktu Proses
Sukses	<pre>{ "success": true, "msg": "Login Berhasil", "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJheGhhckgiLCJlbWVpbCI6ImF6cjEyM0BnbWFpbC5jb20iLCJpYXQiOiJlOTNTI3Njg1OTAsImV4cCI6MTY1Mjc2OTE5MH0.1XhTMrWh_O863J4yR5z8PAypzooWyNTLDQgG1fKyGaU", "refreshToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJheGhhckgiLCJlbWVpbCI6ImF6cjEyM0BnbWFpbC5jb20iLCJpYXQiOiJlOTNTI3Njg1OTB9.K3mGb fEGdBR6oJgRVoPGPnP5meRZWHZ28RI-H4erft4" }</pre>	Ya	461 ms
	HTTP response code: 200 Ok		
Gagal	<pre>{ "success": false, "msg": "Username tidak terdaftar!" }</pre>	Ya	234 ms
	HTTP response code: 400 Bad Request		
Gagal	<pre>{ "success": false, "msg": "password salah!" }</pre>	Ya	241 ms
	HTTP response code: 400 Bad Request		
Gagal	<pre>{ success: false, msg: 'something went wrong on our side...' }</pre>	Ya	40 ms
	HTTP response code: 500 Internal Server Error		

4.3. Pembahasan

Web service ini dikembangkan dengan menggunakan metode spiral, yang mana menghasilkan *workflow* iterasi total selama 12 bulan. Iterasi dibagi menjadi iterasi awal (iterasi pertama) selama 5 bulan pertama, yaitu dari Juni – Oktober 2021. iterasi pertama membahas tentang kebutuhan sistem, fitur-fitur, desain awal, dan *coding* prototipe awal yang menghasilkan sebanyak 75 *endpoint* dan 12 *point risk assessment*. Iterasi kedua berjalan selama 3 bulan dari November 2021 – Januari 2022, membahas tentang pengembangan lanjutan dari tahap pertama, pengembangan tersebut meliputi penambahan dan penyesuaian kebutuhan sistem, pembaruan desain *database*, dan pengembangan prototipe kedua. Iterasi kedua menghasilkan sebanyak 169 *endpoint* dan 3 *point risk assessment*.

Iterasi ketiga berjalan selama 4 bulan dari Februari – Juni 2022. Iterasi ketiga membahas tentang pengembangan akhir atau finalisasi *web service*, dengan penambahan fitur-fitur *non-fungsional* yang bertujuan membantu sistem secara keseluruhan, seperti fitur *caching*, dan fitur TLS, Iterasi ketiga menghasilkan 229 *endpoint*.

Pengujian *endpoint* dilakukan oleh penulis dengan menggunakan metode *Unit Testing*. Berdasarkan hasil pengujian terhadap 229 *endpoint* yang terbagi dalam 6 *role* pengguna sistem dapat dilihat pada tautan berikut: [DOKUMENTASI-webservice-AplikasiUjianEssayPTIK](#). Dan bahwasanya keseluruhan *endpoint* berjalan

dengan baik dan sesuai dengan kebutuhan sistem dan sesuai dengan format pada metode *unit testing*, di mana terdapat skenario keberhasilan dan kegagalan yang diberikan dengan jelas dan mudah dipahami bertujuan untuk membantu pengguna akhir, yaitu pengembang *front-end*.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Web Service aplikasi ujian *essay* PTIK untuk keseluruhan 229 *endpoint* telah melalui uji *unit testing*, dan dapat ditarik kesimpulan sebagai berikut:

1. Pengembangan *web service* menghasilkan sebanyak 229 *endpoint*.
2. Pengembangan *web service* dilakukan dengan metode spiral.
3. Implementasi metode spiral menghasilkan sebanyak 3 iterasi pengembangan yang berjalan dari Juli 2021 – Mei 2022.
4. Data – data untuk membuat tabel rancangan kebutuhan fungsional didapat dari kuesioner awal, kuesioner lanjutan, dan studi literatur.
5. Secara keseluruhan dapat disimpulkan, keseluruhan 229 *endpoint* berjalan dengan baik dan sesuai dengan kebutuhan

5.2. Saran

1. Melibatkan pengguna akhir *endpoint* (dalam kasus ini *front-end developer*) secara lebih banyak pada saat pengembangan aplikasi, serta mengikuti kaidah pengembangan aplikasi *backend* untuk menghindari perubahan yang tidak diinginkan.
2. Mengelola *web service* secara jangka panjang, salah satunya melalui pembaruan teknologi-teknologi (*library*) yang digunakan dalam pengembangan aplikasi.
3. Mendokumentasikan kegunaan *endpoint* secara lebih merinci untuk menghindari kerancuan dalam persepsi fungsi suatu *endpoint*.
4. Menambahkan data-data, dan fitur-fitur yang relevan jika ingin lebih menyepesifikan *target user*, dalam hal ini untuk *user* Program Studi Pendidikan Teknik Informatika dan Komputer (PTIK) Universitas Negeri Jakarta (UNJ).

Daftar Pustaka:

- Battle, R., & Benson, E. (2008). Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST). *Journal of Web Semantics*, 6(1), 61–69. <https://doi.org/10.1016/j.websem.2007.11.002>
- Boehm, B., & Hansen, W. (2001). The spiral model as a tool for evolutionary acquisition. *CrossTalk*, 14(5), 4–11.
- Doglio, F. (2015). *Pro REST API Development with Node.js*. New York: Apress. <https://doi.org/10.1007/978-1-4842-0917-2>
- Fauziah, Y. (2014). Aplikasi iklan baris online menggunakan arsitektur REST web service. *Telematika: Jurnal Informatika dan Teknologi Informasi*, 9(2), 75–80.
- Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures* (Doctoral dissertation). University of California, Irvine.
- Kementerian Pendidikan dan Kebudayaan. (2021). *Kamus Besar Bahasa Indonesia (KBBI) Daring*. Jakarta: Badan Pengembangan dan Pembinaan Bahasa. Diakses dari <https://kbbi.kemdikbud.go.id>
- Kurniawan, E. (2014). Implementasi REST web service untuk sales order dan sales tracking berbasis mobile. *Jurnal Eksplorasi Karya Sistem Informasi dan Sains*, 7(1), 1–12.
- Masse, M. (2011). *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. Sebastopol: O'Reilly Media, Inc.
- Nasril, & Saputra, A. Y. (2016). Rancang bangun sistem informasi ujian online. *Jurnal Lentera ICT*, 3(1), 47–53.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). New York: McGraw-Hill Higher Education.
- Rerung, R. R. (2018). *Pemrograman Web Dasar*. Yogyakarta: Deepublish.
- Richardson, L., & Ruby, S. (2007). *RESTful Web Services*. Sebastopol: O'Reilly Media, Inc.
- Rosa, A. S., & Shalahuddin, M. (2014). *Rekayasa Perangkat Lunak: Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- Satzinger, J. W., Jackson, R. B., & Burd, S. D. (2012). *Systems Analysis and Design in a Changing World* (6th ed.). Boston: Cengage Learning.

Sommerville, I. (2011). *Software Engineering* (9th ed.). Boston: Addison-Wesley.
Sudijono, A. (2010). *Pengantar Statistik Pendidikan*. Jakarta: Rajawali Pers.