

## PENGEMBANGAN *RESTFUL WEB SERVICE* PADA SISTEM KOLABORASI GURU MATA PELAJARAN DI WILAYAH JAKARTA TIMUR DENGAN METODE *SCRUM*

Swardiantara Silalahi<sup>1</sup>, Hamidillah Aje<sup>2</sup>, Widodo<sup>3</sup>

<sup>1</sup> Mahasiswa Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

<sup>2,3</sup> Dosen Prodi Pendidikan Teknik Informatika dan Komputer, Teknik Elektro, FT – UNJ

<sup>1</sup> swardyantara@gmail.com, <sup>2</sup> hamidillah@unj.ac.id, <sup>3</sup> widodo@unj.ac.id

---

### Abstrak

*Kehidupan pada masa kini hampir tidak dapat dipisahkan dari peran teknologi, salah satunya di bidang Pendidikan. Dominasi peran teknologi dapat dilihat dari banyaknya platform yang menyediakan layanan pendidikan. Jika dilihat, platform-platform yang muncul cenderung berfokus pada siswa. Produk yang berfokus pada guru dapat terbilang masih minim. Saat ini, guru bekerja sama dalam mengembangkan perangkat ajar dan bahan ajar hanya sebatas grup pada media sosial seperti Whatsapp, atau LMS buatan luar negeri. Sehingga dibutuhkan sebuah solusi untuk memberikan ruang kepada para guru agar dapat saling berkolaborasi dalam menjalankan tugasnya sebagai tenaga pendidik. Tujuan dari penelitian ini adalah untuk menghasilkan *restful web service* yang dapat digunakan untuk mengembangkan sistem kolaborasi guru mata pelajaran serta memiliki sifat adaptif terhadap perubahan yang perlu dilakukan di masa mendatang. Pengembangan dilakukan dengan menggunakan arsitektur *Service Oriented Architecture (SOA)*, metode *Scrum*, framework *ExpressJS* dan menghasilkan 77 endpoint. Pengujian dilakukan terhadap seluruh endpoint dengan menggunakan metode unit testing dan bantuan tools *Postman*. Dari hasil pengujian menunjukkan seluruh endpoint berfungsi dengan baik dan layak digunakan sebagai sisi backend dari sistem kolaborasi guru mata pelajaran.*

**Kata kunci :** Sistem Kolaborasi Guru, *Scrum*, *Unit Testing*, *Restful Web Service*, *Service Oriented Architecture*

---

### 1. Pendahuluan

Pada agenda tujuan pembangunan berkelanjutan (*SDGs*), pendidikan menjadi salah satu perhatian dari 17 tujuan yang ingin dicapai oleh masyarakat dunia melalui PBB. Lebih dari 265 juta anak-anak di dunia tidak bersekolah, 22% di antaranya adalah anak-anak pada usia sekolah dasar (United Nations, 2015). Indonesia, sebagai salah satu negara anggota PBB ikut mendukung *SDGs*. Untuk mencapai tujuan tersebut, guru menjadi ujung tombak perjuangan dalam upaya mencapai tujuan tersebut. Guru didefinisikan sebagai pendidik profesional dengan tugas utama mendidik, mengajar, membimbing, mengarahkan, melatih, menilai, dan mengevaluasi peserta didik (Presiden Republik Indonesia, 2017).

Fakta di lapangan menunjukkan bahwa kompetensi guru masih rendah (Leonard, 2015). Beban kerja guru yang meliputi mengajar, membuat perangkat pembelajaran, membuat bahan ajar. Bahkan beberapa sekolah yang kekurangan guru akan membebani guru melebihi jam mengajar sewajarnya, yaitu 37,5 jam seminggu (Kementerian Pendidikan dan Kebudayaan, 2018). Pemerintah Indonesia melalui Kementerian Pendidikan dan Kebudayaan melakukan berbagai upaya dalam meningkatkan kualitas tenaga pendidik. Namun, dari 96 guru bersertifikasi di wilayah Karesidenan Surakarta yang meliputi Kota Praja Surakarta, Kabupaten Karanganyar, Kabupaten Sragen, Kabupaten Wonogiri, Kabupaten Sukoharjo, Kabupaten Klaten dan Kabupaten Boyolali, hanya 37% yang dapat menyampaikan materi dengan jelas (Siswandari & Susilaningih, 2013). Setelah pelaksanaan PLPG, pemerintah mengeluarkan program baru bernama PPG (Pendidikan Profesi Guru). Pada pelaksanaannya, PPG masih belum cukup efektif dalam memenuhi kebutuhan akan guru profesional (Rambitan & Hardoko, 2016).

Peran Teknologi Informasi dewasa ini memberikan dampak yang signifikan di berbagai sendi kehidupan manusia. Salah satu keuntungan yang didapat dari kolaborasi daring dari perspektif guru adalah dapat meningkatkan pertukaran ide, informasi dan pengalaman, mewadahi debat dan diskusi, serta menghimpun pendapat dari berbagai sudut pandang (Romeu, Guitert, & Sangra, 2015).

Available at:

<http://journal.unj.ac.id/unj/index.php/pinter/article/view/38675>

Sistem kolaborasi guru merupakan salah satu bentuk solusi konkret dari adanya Teknologi Informasi dalam hal meringankan pekerjaan manusia. Sistem kolaborasi guru dirancang untuk memungkinkan kolaborasi dapat terjadi dengan mudah tanpa kendala jarak dan waktu. Sistem ini dapat meningkatkan pengalaman tentang pengembangan profesionalitas guru, memperoleh dan meningkatkan keterampilan, berbagi dan bertukar perangkat pembelajaran dan bahan ajar, serta bersosialisasi (Kabilan, Adlina, & Embi, 2011). Sistem yang akan dibangun berbasis *web*, dikarenakan oleh halaman web terkoneksi selalu ke internet, sehingga memudahkan untuk diakses dari mana dan kapan saja. Tidak hanya itu, kita dapat mengolah informasi yang didapat secara langsung saat itu juga, dikarenakan data yang di akses bersifat *real-time*. Berdasarkan hasil wawancara, didapat data bahwa sudah ada kegiatan kolaborasi secara daring yang dilakukan oleh narasumber guru, namun masih terbatas melalui sosial media grup seperti *Whatsapp*.

Dalam proses pengembangan perangkat lunak, semua kebutuhan yang dirasa perlu ada dan disediakan oleh perangkat lunak didefinisikan pada tahap awal pengembangan. Daftar kebutuhan tersebut didapat melalui wawancara kepada calon pengguna, lalu didiskusikan oleh tim pengembang untuk dibuat dalam bentuk yang sudah terstandar. Pada faktanya, pada saat proses pengembangan, kebutuhan tersebut dapat berubah sesuai kondisi di lapangan. Hal ini dapat berakibat pada banyak aspek, terutama waktu pengerjaan yang semakin panjang jika kebutuhan bertambah atau berubah. Untuk itu, dibutuhkan metode pengembangan yang adaptif terhadap perubahan yang mungkin terjadi pada saat proses pengembangan sudah dimulai. *Scrum* merupakan salah satu kerangka kerja yang dapat digunakan dalam pengembangan perangkat lunak. Sifatnya yang fleksibel menjadikan *Scrum* menjadi salah satu solusi kerangka kerja pengembangan perangkat lunak dengan tingkat kelincihan (*agility*) yang tinggi.

Untuk mendukung proses pengembangan yang adaptif, sistem dirancang khusus agar dapat menyesuaikan kondisi atau perubahan yang mungkin terjadi di kemudian hari. Dalam konsep pengembangan *web*, bisa dilihat menjadi 2 sisi, yaitu *frontend* dan *backend*. Dengan membagi menjadi dua bagian, diharapkan pada saat proses pengembangan dapat menghasilkan produk yang sesuai dengan kebutuhan, tingkat kesalahan yang kecil dan fleksibel terhadap perubahan dimasa mendatang. Sehingga ketika ada perubahan di sisi tampilan, maka cukup memperbaiki kode di bagian tampilan saja, tanpa menyentuh kode bagian pemrosesan data (*backend*).

Salah satu teknologi yang banyak digunakan untuk pemrosesan data berbasis web adalah *web service*. *Web service* merupakan suatu standar terdefinisi yang memungkinkan dua atau lebih aplikasi dapat saling berkomunikasi, atau saling bertukar informasi / data yang direpresentasikan dalam bentuk *text*, *JSON*, atau *XML*. *Web service* mendukung komunikasi antar aplikasi tanpa memperhatikan teknologi apa yang digunakan dibalik aplikasi tersebut. Hal ini membuat *web service* mendukung interoperabilitas interaksi dua sistem melalui jaringan komputer. Dalam teknologi *web service*, sebuah data dapat diidentifikasi dari sebuah URI (*Uniform Resource Identifier*). Jika sebuah aplikasi membutuhkan data tertentu, maka dapat dilakukan permintaan kepada sebuah URI *web service* yang akan merespons permintaan tersebut dengan teks berformat *JSON/XML* yang berisi informasi yang dibutuhkan oleh aplikasi atau sistem yang mengirimkan permintaan tersebut.

Pada perkembangannya, teknologi *web service* dibagi menjadi dua arsitektur umum, yaitu *Simple Object Access Protocol* (SOAP) dan *Representational State Transfer* (REST). *Web service* yang menggunakan SOAP relatif lebih berat pada saat proses pengiriman data, dikarenakan format pertukaran data yang digunakan dan informasi konfigurasi yang terdapat pada *header* data. Sementara REST yang mengimplementasikan format data *JSON* relatif lebih ringan. Data yang dikirimkan langsung dideskripsikan dalam bentuk *object JavaScript*.

## 2. Dasar Teori

### 2.1. Sistem Kolaborasi Guru

Sistem Kolaborasi merupakan sebuah sistem yang memungkinkan guru untuk bekerja sama dalam proses pembelajaran. Dari mulai merancang pembelajaran, membuat bahan ajar, membuat perangkat pembelajaran sampai membuat butir soal untuk evaluasi pembelajaran.

### 2.2. Website

*Website* adalah keseluruhan halaman-halaman pada web yang dapat diakses melalui sebuah domain yang mengandung informasi tertentu. *Website* dapat digunakan untuk berbagai keperluan, seperti portal berita, blog, sistem informasi dan lain-lain. Menurut perubahan kontennya, *website* dibagi menjadi dua, yaitu *website* statis dan dinamis. *Website* statis adalah *website* yang menampilkan informasi statis, tidak ada perubahan konten dalam waktu tertentu. Sedangkan *website* dinamis memiliki konten yang relatif berubah-ubah seiring digunakan oleh penggunanya (Deitel, Deitel, & Deitel, 2011).

### 2.3. Backend

*Backend* meliputi semua teknologi yang dibutuhkan untuk memproses permintaan yang datang ke server, membuat dan mengirimkan respons ke klien. Teknologi tersebut di antaranya, server sebagai komputer yang akan

menerima permintaan dari klien. Aplikasi *backend*, sebagai aplikasi yang berjalan di sebuah server yang menerima permintaan dari klien, mengambil informasi dari basis data dan mengirimkan respons ke klien. Basis data, digunakan untuk menyimpan dan mengolah data secara berkelanjutan.

#### 2.4. *Service Oriented Architecture*

Ide mendasar dari arsitektur berorientasi layanan terletak pada distribusi layanan. Mari analogikan ada sebuah kota yang memiliki banyak pusat bisnis. Bisnis-bisnis tersebut sudah berorientasi layanan. Masing-masing perusahaan berorientasi pada layanan yang masing-masing menyediakan layanan berbeda yang dapat digunakan oleh banyak konsumen. Secara kolektif, bisnis ini terdiri dari komunitas bisnis. Wajar apabila komunitas bisnis tidak melayani hanya pada satu gerai bisnis yang menyediakan semua layanan. Dengan menguraikan komunitas menjadi gerai khusus, dapat dicapai lingkungan di mana gerai ini dapat didistribusikan. Ketika digabungkan dengan kata “arsitektur”, orientasi layanan memiliki konotasi teknis (Erl, 2017).

Arsitektur berorientasi layanan (SOA) adalah istilah yang mewakili model logika otomatisasi didekomposisi menjadi unit-unit logika yang lebih kecil dan berbeda. Secara kolektif, unit-unit ini merepresentasikan logika bisnis. Secara individual, unit-unit ini dapat didistribusikan (Erl, 2017). Terdapat beberapa prinsip kunci pada SOA, yaitu sebagai berikut.

1. *Loose Coupling*, yaitu bahwa *service* tersebut mempertahankan sebuah hubungan yang meminimalkan ketergantungan dan mereka hanya perlu menjaga kesadaran antar satu sama lain.
2. *Service Contract*, *service* melekat dan taat pada sebuah kesepakatan komunikasi, yang didefinisikan secara kolektif oleh satu atau lebih deskripsi *service* dan dokumen yang relevan.
3. *Autonomy*, *service* memiliki kendali atas logika yang di kerjakan.
4. *Abstraction*, di luar dari apa yang didefinisikan pada *service contract*, *services* menyembunyikan logika dari dunia luar.
5. *Reusability*, logika dibagi menjadi layanan-layanan dengan tujuan agar dapat digunakan kembali.
6. *Composability*, kumpulan dari layanan dapat dikoordinasikan dan dihimpun untuk membentuk layanan yang berbeda. Suatu layanan dapat menjadi bagian dari layanan lainnya.
7. *Statelessness*, *services* meminimalkan penyimpanan informasi khusus untuk suatu kegiatan.
8. *Discoverability*, *services* dirancang untuk bersifat deskriptif luar, sehingga dapat ditemukan dan dinilai melalui mekanisme penemuan yang tersedia.

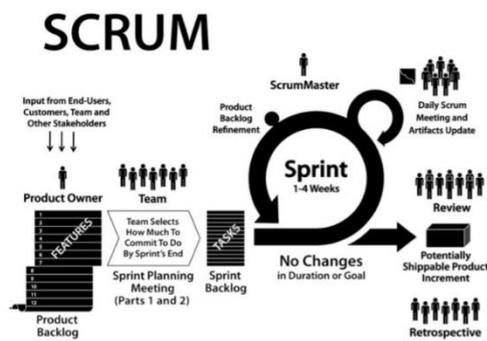
#### 2.5. *Restful Web Service*

*Web service* adalah kumpulan mekanisme dan standar yang digunakan oleh aplikasi atau sistem untuk bertukar data secara terbuka. Artinya terbuka adalah dapat saling dipahami oleh aplikasi atau sistem yang menggunakannya. Perangkat lunak yang ditulis dalam berbagai bahasa pemrograman dan berjalan pada berbagai platform dapat menggunakan *web service* untuk bertukar data melalui jaringan komputer seperti internet dengan cara yang sama seperti komunikasi antar proses dalam sebuah komputer (Petrie, 2016). Hal ini disebabkan oleh penggunaan standar terbuka (*open standard*).

#### 2.6. *Agile Software Development*

*Agile* adalah kemampuan untuk membuat dan merespons perubahan. Ini adalah cara untuk menghadapi, dan akhirnya berhasil dalam, lingkungan yang tidak pasti dan bergolak. Berasal dari kata “*agility*” dalam bahasa Inggris, yang berarti lincah. Pengembangan perangkat lunak *Agile* adalah istilah umum untuk serangkaian *framework* dan praktik berdasarkan nilai-nilai dan prinsip-prinsip yang dinyatakan dalam Manifesto untuk pengembangan perangkat lunak *Agile* dan 12 prinsip di belakangnya. Metode pengembangan perangkat lunak terbagi menjadi dua, yaitu konvensional dan *Agile*. Salah satu contoh dari metode konvensional adalah *Waterfall*. Sedangkan beberapa metode pengembangan yang bersifat *Agile* diantaranya yaitu: *Scrum*, *Extreme Programming* (XP), *Feature Driven Development* (FDD), Kanban, dan masih banyak lagi.

Meskipun *Scrum* sering disebut sebagai metode yang menerapkan prinsip *Agile*, namun sebenarnya *Scrum* lebih mendekati *framework* yang menerapkan prinsip yang ada pada *Agile*, yang merupakan sebuah *framework* di mana orang dapat mengatasi masalah adaptif yang kompleks, sementara secara produktif dan kreatif memberikan produk dengan nilai setinggi mungkin. *Scrum* itu ringan, mudah dimengerti namun sulit untuk dikuasai. *Scrum* adalah kerangka proses yang telah digunakan untuk mengelola pekerjaan pada produk kompleks sejak awal 1990-an. *Scrum* bukanlah proses, teknik atau metode definitif. Sebaliknya, ini adalah *framework* di mana kita dapat menggunakan berbagai proses dan teknik. *Scrum* memperjelas efektivitas relatif dari manajemen produk dan teknik kerja sehingga dapat terus meningkatkan produk, tim, dan lingkungan kerja (Schwaber & Sutherland, 2017). Berikut ilustrasi proses pengembangan perangkat lunak dengan menggunakan *Scrum*.



Gambar 1. Tahapan Pengembangan Scrum (Sutherland, 2011)

### 3. Metodologi

#### 3.1 Alat dan Bahan Penelitian

Penelitian ini akan menggunakan satu buah laptop dan perangkat lunak untuk perancangan, pembuatan dan pengujian *web service*, dengan spesifikasi perangkat keras dan perangkat lunak dapat dilihat pada Tabel 1.

Tabel 1. Daftar Perangkat Keras (*Hardware*)

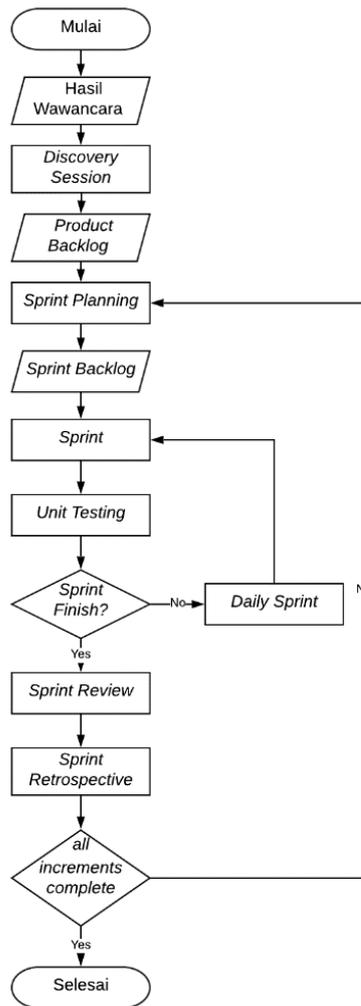
Perangkat Keras	Spesifikasi
Processor	Intel Core i7-8550U berkecepatan 1.8GHz.
RAM	DDR4 8G dengan frekuensi 2400MHz.
SSD	128 GB
Display	Intel(R) UHD Graphics 620 (1920x1080)

Penelitian ini dilakukan dengan menggunakan hasil analisis kebutuhan dari sejumlah calon pengguna. Sehingga didapat data berupa daftar kebutuhan perangkat lunak dalam bentuk *user story*. Kegiatan analisis kebutuhan perangkat lunak dilakukan dengan mengumpulkan data melalui wawancara ke beberapa responden yang merupakan guru-guru di wilayah Jakarta Timur sebagai calon pengguna dari sistem yang akan dikembangkan. Data hasil analisis kebutuhan merupakan data primer dalam penelitian ini. Pembuatan sistem akan menggunakan metode *Scrum*, di mana kebutuhan perangkat lunak akan didefinisikan ke dalam sebuah artefak bernama *product backlog*.

#### 3.2 Diagram Alir Penelitian

Penelitian ini dilakukan dengan menggunakan salah satu *framework Agile*, yaitu *Scrum* yang mengutamakan sifat adaptif dan iteratif dalam proses pengembangan perangkat lunak. Tahap pertama adalah membuat *product backlog*. Langkah ini dilakukan dengan membuat poin-poin *Product Backlog Item* (PBI) berpatokan pada daftar kebutuhan perangkat lunak yang sudah disusun ketika melakukan wawancara. Tahap kedua, dilakukan perencanaan untuk melaksanakan kegiatan sprint berdasarkan urutan prioritas yang sudah diuraikan pada *product backlog*. Pada tahap ini tim pengembang juga mendiskusikan cara untuk mengerjakan setiap *increment* agar bisa selesai tepat waktu. Selanjutnya, setiap anggota tim bekerja sesuai dengan *jobdesc* dan tanggung jawabnya masing-masing berdasarkan apa yang sudah tertulis pada *sprint backlog*. Tahapan ini adalah tahap di mana dilakukan perancangan, implementasi dan pengujian komponen perangkat lunak yang menyusun sebuah *Product Backlog Item*. Setelah itu dilakukan *Sprint Review* dan *Sprint Retrospective*, yaitu peninjauan ulang serta refleksi terhadap pekerjaan yang sudah dilakukan.

Secara garis besar, metode penelitian yang akan dilaksanakan seperti diagram alir berikut ini.



Gambar 2. Alur Penelitian

### 3.3 Teknik dan Prosedur Pengumpulan Data

Penelitian ini menggunakan teknik pengumpulan data dengan cara wawancara dan studi literatur. Instrumen wawancara berisi pertanyaan-pertanyaan yang diajukan kepada calon pengguna, yaitu guru-guru SMK di wilayah Jakarta Timur mengenai kesulitan-kesulitan yang dialami dalam melakukan pekerjaan sebagai seorang guru serta kemudahan-kemudahan yang akan dirasakan jika dibangun sistem kolaborasi guru untuk menunjang kinerja guru dan meningkatkan kualitas pembelajaran. Dari hasil wawancara, ditemukan bahwa guru merasa beban kerja yang diemban tergolong berat, terlebih bagi mereka yang mendapat tugas tambahan seperti menjadi kepala kompetensi keahlian. Setelah melakukan wawancara, selanjutnya dilakukan studi literatur untuk mengumpulkan, memilah, menganalisis data sekunder dari penelitian lain yang sesuai dengan kebutuhan penelitian ini. Setelah data-data terkumpul, dilakukan pengujian dengan menggunakan metode *unit testing*, yaitu dengan menguji masing-masing *endpoint* yang telah dibuat. Hasil pengujian akan dijadikan dokumentasi yang dapat digunakan untuk membangun sistem kolaborasi guru secara utuh.

### 3.4 Teknik Analisis Data

Pengujian menggunakan metode *unit testing*, yaitu pengujian yang dilakukan terhadap setiap *endpoint* yang telah dibuat pada proses pengembangan *restful web service*. Unit testing berfokus pada upaya verifikasi unit terkecil dari sebuah perangkat lunak, yaitu komponen atau modul yang dalam penelitian ini, disebut *endpoint*. Untuk melakukan hal tersebut, pengujian akan berfokus pada logika pemrosesan internal dan struktur data yang digunakan dalam setiap *endpoint*. Selama melakukan pengujian, dapat dilakukan secara paralel untuk beberapa *endpoint* sekaligus (Pressman & Maxim, 2015).

Tabel 2. Pengujian fungsional pada *endpoint*

Skenario	Respons	Sesuai Kebutuhan	Waktu Proses
Sukses	Data responss kondisi sukses		
Gagal	Data responss kondisi gagal 1		
	Data responss kondisi gagal 2		

#### 4. Hasil dan Analisis

##### 4.1 Hasil Penelitian

Penelitian ini menghasilkan *web service* sebanyak 77 *endpoint*. Setiap *endpoint* merupakan luaran dari kegiatan *sprint*. Dalam proses pengembangan, terdapat 3 tahapan yang dilakukan yaitu perancangan, implementasi dan pengujian. Pada tahap perancangan, dilakukan pemilihan *HTTP Method* yang digunakan, menentukan apa saja data masukan yang dibutuhkan serta format dan strukturnya, lalu menentukan apa saja data yang dihasilkan sebagai luaran serta format dan strukturnya. Dengan menggunakan *Scrum* sebagai metode pengembangan, proses pengembangan *lancer*, tanpa memakan waktu lama pada tahap perancangan. Setiap perancangan bersifat dinamis, dikarenakan *Scrum* bersifat iteratif, maka hasil perancangan pada *sprint* sebelumnya dapat diperbaiki pada *sprint* selanjutnya. Pada tahap implementasi, dilakukan penulisan kode sesuai dengan spesifikasi hasil dari perancangan *endpoint* dengan menggunakan arsitektur SOA. Setelah *endpoint* selesai dikerjakan, maka tahap terakhir yaitu pengujian. Pengujian dilakukan menggunakan *tools* Postman dan dengan metode *Unit Testing*. Pengujian dilakukan untuk melihat apakah *endpoint* yang dibuat sudah sesuai dengan kebutuhan.

##### 4.2 Analisis Data Penelitian

Pada tahap pengujian, setiap *endpoint* yang sudah dibuat akan dilakukan percobaan menggunakan 2 skenario, yaitu berhasil dan gagal. Kedua skenario tersebut merupakan kondisi sesuai kebutuhan. Pengujian dilakukan menggunakan *tools* Postman, dengan mendokumentasikan responss dari setiap API berupa data dan waktu proses. Salah satu hasil dari pengujian *endpoint* pada *sprint* pertama dapat dilihat pada Tabel 3 berikut ini.

Tabel.3. Hasil Pengujian Endpoint R001

Skenario	Respons	Sesuai Kebutuhan	Waktu Proses
Sukses	{ "code": 200, "message": "Email belum digunakan" }	Ya	7ms
Gagal	{ "code": 400, "message": "Email sudah digunakan" }	Ya	8ms

##### 4.3 Pembahasan

Pengembangan *restful web service* ini dilakukan dengan menggunakan *Scrum* sebagai metode pengembangan. Selama pengerjaan *sprint*, tim *Scrum* bekerja sama untuk merealisasikan setiap daftar kebutuhan yang ada pada *product backlog*. Dari tiga *sprint* yang sudah dikerjakan, *sprint* pertama memakan waktu paling singkat, yaitu tiga hari untuk mengerjakan 6 *endpoint*. Pada *sprint* kedua, terdapat 30 *endpoint* yang dihasilkan dari 9 *Product Backlog Item* dan memakan waktu selama 7 hari. Kemudian *sprint* ketiga dikerjakan selama 8 hari untuk mengerjakan 41 *endpoint* yang dihasilkan dari 13 *Product Backlog Item*. Pengujian dilakukan oleh penulis sendiri dengan menggunakan metode *Unit Testing*. Berdasarkan hasil pengujian dapat dilihat bahwa semua *endpoint* dapat berjalan dengan baik dan sudah sesuai dengan kebutuhan. Semua kemungkinan kesalahan sudah ditangani dengan baik serta pesan kesalahan dibuat sedemikian sehingga merepresentasikan permasalahan yang sebenarnya untuk memudahkan penanganan kesalahan pada sisi pengguna.

Perangkat yang digunakan dari tahap perancangan, pengembangan sampai pengujian menggunakan perangkat dengan spesifikasi pada Tabel 3.1. Sehingga hasil pengujian dengan menggunakan perangkat yang

memiliki spesifikasi lebih atau kurang, dapat menghasilkan hasil pengujian yang berbeda secara performa, namun tetap sama dari sisi fungsional. Oleh karena itu, hasil pengujian hanya terbatas pada pengujian fungsional, dan tidak menyajikan pengujian performa ataupun pengujian beban.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Pengembangan *restful web service* dilakukan dengan menggunakan metode Scrum, bahasa pemrograman JavaScript dengan bantuan NodeJS agar kode dapat dieksekusi di sisi *server*, dan ExpressJS sebagai *framework*. Sistem basis data relasional yang digunakan adalah MySQL untuk menyimpan data-data yang digunakan dalam sistem. Proses pengujian dilakukan dengan bantuan *tools* Postman. Metode *Scrum* yang digunakan sebagai metode pengembangan dinilai cocok untuk mengembangkan *web service*. Kebutuhan perangkat lunak direpresentasikan dalam bentuk *Product Backlog* dan digunakan sebagai acuan dalam perancangan dan implementasi *endpoint* yang akan dikembangkan. Sehingga dihasilkan sebanyak 77 *endpoint* dari 27 *Product Backlog Item*. Kemudian dilakukan pengujian terhadap semua *endpoint* yang sudah dikembangkan untuk mengetahui apakah *endpoint* yang dikembangkan sudah sesuai berfungsi dengan baik serta memberikan responss yang sesuai dengan kebutuhan pengguna.

### 5.2 Saran

Beberapa saran dari penelitian ini yaitu sebagai berikut.

1. Mengikuti prosedur pengembangan *web service* atau aplikasi sisi *backend* lainnya untuk menghindari perubahan-perubahan yang tidak diperlukan dengan melibatkan pengguna pada saat melakukan perancangan *endpoint*.
2. Menjaga keterbaruan teknologi yang digunakan dengan selalu melakukan pengecekan terhadap teknologi yang ada pada bidang pengembangan *web service*.
3. Seluruh *endpoint* yang dibuat memiliki dokumentasi yang baik agar pengguna dapat dengan mudah menggunakannya.
4. Pengujian yang dilakukan terbatas pada pengujian fungsional, sehingga tidak memberikan gambaran tentang hasil pengujian lain seperti uji performa dan uji beban.
5. Daftar kebutuhan yang digunakan terbatas hanya pada kebutuhan fungsional, sehingga tidak memperhatikan kebutuhan sistem dari sisi non-fungsional seperti kebutuhan kapasitas penyimpanan, keamanan, *deployment*, dll.

### Daftar Pustaka:

- Deitel, P., Deitel, H., & Deitel, A. (2011). *Internet & World Wide Web: How to Program* (5th ed.). Boston: Prentice Hall.
- Erl, T. (2017). *Service Oriented Architecture: Analysis and Design for Services and Microservices*. Boston, Massachusetts, Amerika Serikat: Prentice Hall.
- Kabilan, M. K., Adlina, W. F., & Embi, M. A. (2011). Online Collaboration of English Language Teachers for Meaningful Professional Development Experiences. *English Teaching: Practice and Critique*, 10(4), pp. 94-115.
- Kementerian Pendidikan dan Kebudayaan. (2018). Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 15 Tahun 2018 Tentang Pemenuhan Beban Kerja Guru, Kepala Sekolah dan Pengawas Sekolah. Jakarta: Direktur Jenderal Peraturan Perundang-undangan Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.
- Leonard. (2015). Kompetensi Tenaga Pendidik di Indonesia: Analisis Dampak Rendahnya Kualitas SDM Guru dan Solusi Perbaikannya. *Jurnal Formatif*, 192-201.
- Petrie, C. J. (2016). *Web Service Compositioin*. Bastrop, Texas, Amerika Serikat: Stanford Logic Group.
- Presiden Republik Indonesia. (2017). Peraturan Pemerintah Republik Indonesia Nomor 19 Tahun 2017 Tentang Perubahan Atas Peraturan Pemerintah Nomor 74 Tahun 2008 Tentang Guru. Jakarta, Indonesia: Kementrian Hukum dan Hak Asasi Manusia Republik Indonesia.
- Pressman, R. S., & Maxim, B. R. (2015). *Software Engineering: A Practitioner's Approach* (8 ed.). New York: Mc Graw Hill Education.
- Rambitan, V. M., & Hardoko. (2016). Pengembangan Desain Manajemen untuk Efektivitas Implementasi Pelaksanaan Program Pendidikan Profesi Guru (PPG) di Daerah Perbatasan Provinsi Kalimantan Timur dan Utara. *Proceeding Biology Education Conference*(13), 388-398.

- Romeu, T., Guitert, M., & Sangra, A. (2015, Maret 30). Teacher Collaboration Network in Higher Education: Reflective Visions from Praxis. *Innovations in Education and Teaching International*, 53(6), 592-604. doi:10.1080/14703297.2015.1025807
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*. USA: Attribution Share-Alike license of Creative Commons.
- Siswandari, & Susilaningih. (2013). *Dampak Sertifikasi Guru Terhadap Peningkatan Kualitas Pembelajaran Peserta Didik*. *Jurnal Pendidikan dan Kebudayaan*(19), 487-498.
- Sutherland, J. (2011). *The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework*. Paris: Scrum Inc.
- United Nations. (2015). *Goals: 4. Quality Education*. Retrieved from Sustainable Development Goals: <https://sustainabledevelopment.un.org/sdg4>