

# ardi

*by* Yunda heningtyas

---

**Submission date:** 14-Feb-2022 11:26PM (UTC-0500)

**Submission ID:** 1756580132

**File name:** 2869-7180-1-PB.pdf (962.53K)

**Word count:** 6551

**Character count:** 39313

<sup>1</sup>  
**Web Service Sistem Informasi Terpadu (SIMIPA) Menggunakan REST API**

<sup>1</sup>Ardiansyah, <sup>2</sup>Didik Kurniawan, <sup>3</sup>Dwi Sakheti, <sup>4</sup>Bustomi & <sup>5</sup>Bambang Hermanto

<sup>1,2,3,4</sup>Jurusan Ilmu Komputer FMIPA Universitas Lampung, Bandar Lampung 35145

e-mail: <sup>1</sup>[ardiansyah@fmipa.unila.ac.id](mailto:ardiansyah@fmipa.unila.ac.id), <sup>2</sup>[didikunila@gmail.com](mailto:didikunila@gmail.com), <sup>3</sup>[dwijim@fmipa.unila.ac.id](mailto:dwijim@fmipa.unila.ac.id),  
<sup>4</sup>[tomblok.id@gmail.com](mailto:tomblok.id@gmail.com), <sup>5</sup>[bambang.hermanto@fmipa.unila.ac.id](mailto:bambang.hermanto@fmipa.unila.ac.id)

<sup>1</sup>  
**Abstract** — SIMIPA web service is a service that is used to provide services in the form of information to the Android mobile apps, is the Student module, the Lecturer module, and the Parent module. The service provided uses the REST API as a bridge between the server and the client. The goal is that the REST API can be accessed on the Android mobile apps. The development of web services in this study uses the Scrum development method which has 26 product backlogs with 9 iterations of sprints. Each sprint has several stages, namely sprint planning, daily sprint, sprint review, and sprint retrospective. Web services developed using the PHP programming language and MySQL database. The result of this research is the SIMIPA REST API which can be accessed on the Android front-end. Android accesses the REST API via the GET and POST request methods using a URL that generates a JSON response. Based on several tests, the SIMIPA REST API URL is said to have been successfully developed and can be accessed on Android mobile apps according to the expected response. Data security using the REST API is focused on the POST method using JSON Web Token (JWT).

**Keywords:** *Web service; REST API; Scrum; JSON web token; SIMIPA*

## 1. PENDAHULUAN

Pendidikan adalah upaya mengembangkan potensi peserta didik baik potensi akademik maupun non-akademik. Upaya tersebut berlangsung dalam proses pembelajaran antara peserta didik dan pendidik. Pada jenjang perguruan tinggi, peserta didik adalah mahasiswa dan pendidik adalah dosen. Saat ini proses pembelajaran antara pendidik dan peserta didik tidak lepas dari teknologi informasi dan sarana-sarana yang membantu dalam kegiatan pembelajaran.

Fakultas Matematika dan Pengetahuan Alam (FMIPA) Universitas Lampung merupakan salah satu sarana atau tempat untuk kegiatan pendidikan. FMIPA Universitas Lampung memiliki 5 jurusan yaitu Kimia, Matematika, Biologi, Fisika dan Ilmu Komputer. Setiap jurusan memiliki berbagai informasi yang dikemas dalam teknologi informasi berupa *web* [fmipa.unila.ac.id](http://fmipa.unila.ac.id) yang dapat diakses sesuai kebutuhan. *Web* FMIPA juga memiliki sistem informasi terpadu yang bernama SIMIPA (Sistem Informasi Terpadu FMIPA). Sistem Informasi tersebut dapat diakses oleh mahasiswa dan dosen serta staf yang memiliki kebutuhan informasi atau layanan bagi masing-masing selama proses pendidikan berlangsung.

Proses bisnis yang berjalan pada SIMIPA adalah mahasiswa dan dosen memilih kebutuhan informasi guna untuk kegiatan akademik maupun non-akademik. Kemudian staf bertugas memvalidasi dan melaporkan ke setiap bidang yang terdapat pada FMIPA sesuai kebutuhan informasi yang terdaftar. Sistem ini masih berbasis *web* menggunakan Wordpress.

*Scrum* adalah sebuah kerangka kerja di mana orang-orang dapat menyelesaikan permasalahan kompleks yang senantiasa berubah, di mana pada saat bersamaan menghasilkan produk dengan nilai setinggi mungkin secara kreatif dan produktif. *Scrum* bersifat ringan, mudah dipahami, serta sulit dikuasai.

Metode *scrum* memiliki 3 *roles* yaitu *scrum master*, *product owner*, dan *scrum team* yang memiliki peran masing-masing [1].

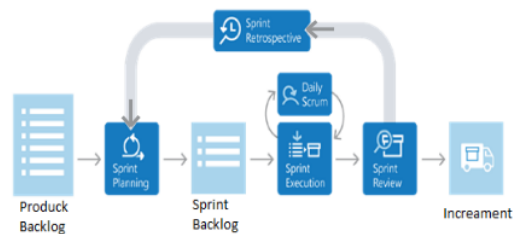
Berhubungan dengan latar belakang diatas pada penelitian ini akan dikembangkan *web service* untuk SIMIPA menggunakan REST API. Hasil penelitian ini diharapkan menjadi sebuah *service* untuk komunikasi antara *server* dengan *client* yang akan digunakan pada *front end* berbasis Android. REST API yang dikembangkan menggunakan *database* MySQL, karena memiliki struktur tabel yang *flexible*. Selain itu metode *scrum* dipilih sebagai kerangka kerja, dan memiliki 26 buah *product backlog* yang telah disetujui dengan tujuan mengembangkan *backend* SIMIPA menggunakan REST API yang memiliki *running time* yang baik dalam pengolahan dan pemrosesan data.

## 2. METODOLOGI PENELITIAN

*Web service* SIMIPA dikembangkan menggunakan REST API dengan metode penelitian *scrum*. Menurut Prakasa dan Setiawan *web service* adalah sistem perangkat lunak yang didesain dapat dioperasikan mesin ke mesin melalui jaringan. *web service* memungkinkan sarana operasi antar perangkat lunak menjadi standar pada berbagai *platform* yang berbeda [2]. REST API menurut webber adalah *Representational State Transfer* (REST) merupakan jenis arsitektur yang terdapat pada web untuk melayani suatu *service*. REST dapat digunakan sebagai *interface* dari *Application Programming Interface* (API) untuk mengakses suatu *resource*. Prinsip REST arsitektur memberikan kemudahan bagi *developer* untuk tidak perlu mengetahui bagaimana struktur dari API di dalam *server* [3]. Response REST API yang diberikan merupakan hasil dari sebuah *query* yang dijalankan pada *database* MySQL. Menurut Winamo MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan [4]. Menurut Kadir *database* merupakan suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi [5]. Keuntungan menyimpan data di *database* adalah kemudahannya dalam penyimpanan dan menampilkan data karena dalam bentuk table.

### 2.1 Metode Scrum

Metode *scrum* digunakan karena mampu mengatasi masalah yang kompleks. Metode ini memprioritaskan tugas-tugas individu, lalu mendelegasikan tugas-tugas tersebut kepada anggota tim yang dianggap sesuai untuk menyelesaikan tugas tersebut [6]. Metode *scrum* memiliki beberapa tahapan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Metode Scrum [6].

#### 2.1.1 Product Backlog

Tahap awal dari metode *scrum* adalah penyusunan *product backlog*. *Product backlog* adalah daftar kebutuhan atau fitur yang memberikan nilai bisnis klien, dan produk backlog yang dibuat dapat bertambah [7]. Daftar *backlog* yang didapatkan dijalankan atau dikerjakan sesuai *sprint* yang telah ditentukan pada Table 1.

Tabel 1. *Sprint Backlog web service* pada pengembangan SIMIPA

No.	Product Backlog	Estimasi (Jam)	Tingkat Kesulitan	Sprint ke-
1.	Analisis <i>database</i> SIMIPA	4	Tinggi	1
2.	Analisis proses bisnis SIMIPA	6	Tinggi	1
3.	Diagram class SIMIPA	6	Sedang	1
4.	<i>Entity Relationship</i> Diagram SIMIPA	6	Sedang	1
5.	Perancangan <i>database</i> SIMIPA	8	Tinggi	1

Tabel 1. *Sprint Backlog web service* pada pengembangan SIMIPA (Lanjutan)

No.	Product Backlog	Estimasi (Jam)	Tingkat Kesulitan	Sprint ke-
6.	Implementasi <i>database</i> SIMIPA	10	Tinggi	2
7.	Pengembangan fungsi <i>register, login</i> dan <i>verification</i>	10	Tinggi	2
8.	Pengembangan fungsi tambah mahasiswa ( <i>Apps Parent</i> )	6	Sedang	2
13.	Pengembangan fungsi <i>READ</i> prestasi	5	Sedang	4
14.	Pengembangan fungsi <i>READ</i> beasiswa	5	Sedang	4
15.	Pengembangan fungsi CRUD kehadiran seminar	15	Tinggi	5
16.	Pengembangan fungsi rekapitulasi seminar	10	Sedang	5
17.	Pengembangan fungsi cek <i>update</i>	8	Sedang	6
18.	Pengembangan Fungsi <i>READ</i> berita	8	Sedang	6
19.	Pengembangan Fungsi Tanggal Penting ( <i>Apps Parent</i> )	5	Sedang	7
20.	Pengembangan Fungsi Validasi Orang tua ( <i>Apps Student</i> )	15	Tinggi	7
21.	Pengembangan Fungsi Validasi Kehadiran Seminar ( <i>Apps Lecturer</i> )	10	Tinggi	7
22.	Pengembangan Fungsi Notifikasi ( <i>Apps Parent</i> )	15	Tinggi	8
23.	Pengembangan Fungsi Pengajuan Beasiswa yang tersedia ( <i>Apps Student</i> )	10	Tinggi	8
24.	Pengembangan Fungsi Pengajuan Prestasi Mahasiswa ( <i>Apps Student</i> )	15	Tinggi	8
25.	Pengembangan Fungsi Pengesahan Tema Skripsi ( <i>Apps Lecturer</i> )	15	Tinggi	9
26.	<i>Security POST</i> Menggunakan JWT	15	Tinggi	9

### 2.1.2 *Sprint planning*

*Sprint planning* dilakukan oleh *product owner* untuk menjelaskan *product backlog* kepada *development team*. *Product backlog* pada *sprint planning* akan dibagi lagi menjadi *sprint goal* dan *sprint backlog* [8].

### 2.1.3 Daily Scrum

Tahapan ini dilakukan oleh seluruh tim SIMIPA untuk mengerjakan *sprint backlog* masing-masing. *Daily scrum* berguna untuk meningkatkan kemungkinan tim *scrum* untuk mencapai *sprint goal* sesuai waktu yang telah ditentukan. Tahapan ini juga bertujuan untuk melakukan pencocokan setiap pekerjaan yang telah dilakukan oleh setiap tim *scrum* [9].

### 2.1.4 Sprint review

*Sprint review* dilakukan oleh seluruh tim untuk mempresentasikan hasil pekerjaan yang telah dilakukan. Tujuan dari *sprint review* adalah untuk memeriksa dan mengadaptasi produk yang sedang kembangkan [10].

### 2.1.5 Sprint Retrospective

Pada tahapan ini, *Scrum Master* akan memberikan masukan kepada tim *scrum* agar melakukan peningkatan pada *sprint* yang akan dilakukan berikutnya, serta *scrum master* akan mengedukasi tim untuk selalu bekerja di bawah batas waktu yang telah ditentukan, agar *sprint backlog* dapat berjalan dengan baik tanpa ada masalah sedikitpun [11].

## 2.2 Pengujian Sistem

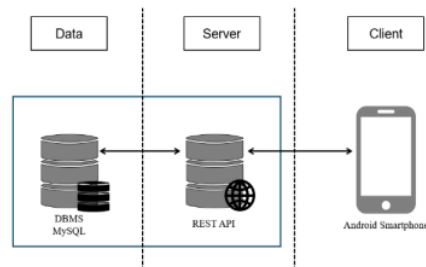
Dalam penelitian ini, akan digunakan pengujian sistem dengan menggunakan metode *black box testing*. *Black box testing* merupakan teknik pengujian perangkat lunak yang menguji *requirements* dan fungsional dari perangkat lunak [12]. Input yang diberikan berupa sebuah *string*, *number*, dan *array*, sedangkan output adalah *Javascript Object Notation* (JSON). JSON adalah salah satu bahasa *markup* yang dapat melakukan pertukaran data dimana JSON ini dibuat berdasarkan *javascript* [13]. Pertukaran data menggunakan API melalui perintah HTTP yang bisa digunakan dalam REST adalah GET, POST, PUT dan DELETE [14]. Berikut skenario pengujian REST API SIMIPA yang dapat dilihat pada Tabel 2.

Tabel 2. Skenario pengujian URL REST API SIMIPA

Nama Pengujian	Request method	REST API URL	Param	Status Pengujian	Running Time
Pengujian 1	POST, GET	/pengujian1	Pengujian1	[√][X]	ms
Pengujian 2	POST, GET	/pengujian2	Pengujian2	[√][X]	ms
Pengujian 3	POST, GET	/pengujian3	Pengujian3	[√][X]	ms

## 3. HASIL DAN PEMBAHASAN

Arsitektur penelitian pengembangan *backend* SIMIPA yang digunakan terdiri atas 3 bagian, yaitu data, server, dan client. Database relasional MySQL digunakan pada bagian data. Sedangkan bagian server, *backend* berfungsi sebagai penghubung antara database dan *front end*. Bahasa pemrograman yang digunakan untuk mengembangkan *backend* adalah PHP dengan WordPress 5.3.2. Bagian *client* terdiri dari 3 modul Android, yaitu modul *student* (Mahasiswa), modul *lecturer* (Dosen) dan modul *parent* (Orang tua/wali) yang dikerjakan menggunakan *software* Android Studio. Arsitektur awal penelitian dapat dilihat pada Gambar 2.



Gambar 2. Arsitektur Awal Penelitian

1  
3.1. *Sprint ke-1*

1  
3.1.1 *Sprint Planning*

*Sprint planning* ke-1 berfokus pada analisis kebutuhan sistem serta pemodelan. Analisis dan pemodelan pada tahapan ini menjadi acuan kerja tim dalam pengembangan *web service* SIMIPA, sehingga pada *sprint* selanjutnya tidak akan terjadi banyak perubahan.

1  
3.1.2 *Daily Scrum*

*Product backlog* pertama adalah analisis *database* SIMIPA. Analisis yang telah dilakukan lebih menekankan pada struktur dari tabel yang akan digunakan, seperti penggunaan *type* data, panjang data, *primary key*, serta *foreign key* yang digunakan sebagai penghubung atau relasi antar tabel. Normalisasi *database* perlu digunakan untuk menghilangkan dan mengurangi redundansi data dan memastikan dependensi data (data berada pada tabel yang tepat).

Analisis proses bisnis SIMIPA yang telah dilakukan menghasilkan penambahan 20 fungsi umum baru yang akan dikembangkan pada *web service* SIMIPA. Fungsi baru tersebut yaitu, *register* orang tua/wali, *login*, *verification*, menambah mahasiswa, melihat *splashscreen*, melihat jadwal, melihat kalender akademik, melihat layanan, melihat prestasi, melihat beasiswa, CRUD kehadiran, melihat rekapitulasi kehadiran, melihat kemajuan studi, dan melihat *update* aplikasi *mobile*. Fungsi tersebut akan digunakan pada bagian *front end* Android 3 modul yaitu Modul *Student*, Modul *Lecturer*, dan Modul *Parent*.

Diagram *class* SIMIPA menunjukkan *class* dalam sistem dan asosiasi antara kelas-kelas. Kelas-kelas ini dianggap sebagai definisi umum dari satu jenis objek pada SIMIPA.

Perancangan ERD SIMIPA dikerjakan oleh seluruh tim pengembang. ERD dirancang dengan mengidentifikasi, menetapkan seluruh himpunan entitas yang terlibat, dan menentukan atribut *key* dari masing-masing entitas. Selanjutnya ERD dapat digunakan untuk pengembangan *database* SIMIPA. Berikut ERD SIMIPA dapat dilihat pada Gambar 3.



*verification*. Pada pengembangan fungsi *register*, *login* dan *verification* ini terdiri dari 6 URL yang kemudian akan dipanggil pada bagian *front end* Android.

URL pertama yaitu */register-parent.php* yang digunakan untuk mendaftarkan user modul *parent*. Pada fungsi ini menggunakan *method POST* dengan parameter *no\_hp*, *nama*, *password*, dan *npm* (mahasiswa yang akan di monitoring). Sebelum *user parent* terdaftar, pada fungsi *register parent* mengecek terlebih dahulu apakah *npm* yang didaftarkan *parent* terdaftar di FMIPA, jika *npm* terdaftar maka *user* akan otomatis terdaftar. Jika *npm* tidak terdaftar maka akan mengeluarkan pesan bahwa *npm* yang di inputkan tidak terdaftar pada FMIPA.

URL kedua yaitu */login-parent.php* yang digunakan untuk masuk aplikasi *mobile* SIMIPA modul *parent*. Pada fungsi ini menggunakan *method POST* dengan parameter *nohp* sebagai *username*, *pass* (*password*), *imei* (IMEI *smartphone* yang digunakan saat menggunakan *apps* SIMIPA) dan *ip* (IP jaringan yang digunakan saat login). Pada *method POST* selain digunakan untuk mengecek apakah parameter *nohp* terdaftar dan *password* yang dimasukan benar, pada URL ini sekaligus mengirim data *nohp*, *IMEI* dan *IP* ke *database* sebagai tanda bahwa akun telah digunakan pada perangkat tersebut.

URL ketiga yaitu */login.php* yang digunakan untuk modul *student* dan *lecturer* dimana data *user*nya diambil dari *web* *fmipa.unila.ac.id* sehingga mahasiswa dan dosen tidak perlu mendaftar lagi pada aplikasi *mobile* Android. Pada fungsi ini menggunakan *method GET* dan *POST*. *Method GET* digunakan untuk membedakan penggunaan URL login yaitu dengan parameter *status*, dimana untuk modul *student* menggunakan *status* Mahasiswa sedangkan modul *lecturer* menggunakan *status* Dosen. Kemudian *method POST* yang digunakan pada kedua modul yaitu dengan parameter *user* (*NPM* dari mahasiswa atau *NIP* dari dosen), *pass* (*password* yang telah dibuat sebelumnya pada *web*), *imei* dan *ip*. Pada URL ini juga sama seperti URL sebelumnya yaitu mengirim data ke *database* sebagai tanda bahwa akun telah digunakan pada perangkat tersebut.

URL keempat yaitu */profile.php* yang digunakan untuk modul *student*. Pada URL ini merupakan *verification* dari pengguna aplikasi *mobile* modul *student* setelah login, dengan mengecek *npm* dan menampilkan hasil berupa data diri dari pengguna yang telah masuk. *Method* yang digunakan pada fungsi ini adalah *GET* dengan parameter *npm*. Selain digunakan pada modul *student*, fungsi ini juga digunakan pada modul *parent* untuk mengecek data *npm* yang didaftarkan untuk dimonitoring oleh *parent* tersebut.

URL kelima yaitu */profile-dosen.php* yang digunakan untuk modul *student*. Pada URL ini merupakan *verification* dari pengguna aplikasi *mobile* modul *Lecturer* setelah login, dengan mengecek *nip* dan menampilkan hasil berupa data diri dari pengguna yang telah masuk. *Method* yang digunakan pada fungsi ini adalah *GET* dengan parameter *nip*.

URL keenam yaitu */profile-parent.php* yang digunakan untuk modul *student*. Pada URL ini merupakan *verification* dari pengguna aplikasi *mobile* modul *student* setelah login, dengan mengecek *no\_hp* dan menampilkan hasil berupa data diri dari pengguna yang telah masuk. *Method* yang digunakan pada fungsi ini adalah *GET* dengan parameter *no\_hp*.

Selanjutnya *product backlog* ketiga yaitu pengembangan fungsi tambah mahasiswa (modul *parent*). Fungsi ini digunakan menambahkan mahasiswa baru pada aplikasi *mobile* modul *parent* untuk dimonitoring oleh orang tua/wali. URL yang digunakan untuk memanggil fungsi ini adalah *parent-create-student.php*. *Method* yang digunakan adalah *method POST* dengan parameter *npm* dan *no\_hp*.

Kemudian *product backlog* keempat yaitu pengembangan fungsi *read update splashscreen*. Fungsi ini digunakan untuk menampilkan gambar *spalshscreen* berdasarkan *event* yang ada di Universitas Lampung atau *event* nasional. *Method* yang digunakan adalah *method GET* dengan parameter *flag*. URL untuk memanggil fungsi ini yaitu *read-update-splashscreen.php*.



### 3.1.3 Sprint Review

Pada tahap ini seluruh tim berkumpul dan berdiskusi untuk membahas hal yang telah dilakukan pada *sprint* 2. Tim *development* mempresentasikan hasil yang sudah dikerjakan selama *sprint* 2. Pengembangan REST API pada URL pertama `/register.php` terdapat masukan dari tim aplikasi *mobile* modul *parent* yaitu untuk menambahkan parameter foto dan fungsi unggah foto untuk menambahkan foto profil pengguna, sehingga parameter yang digunakan pada URL ini antara lain `no_hp`, `nama`, `password`, `npm` dan `foto`.

Kemudian tim aplikasi *mobile* modul *parent* juga meminta untuk menambahkan fungsi *read* daftar mahasiswa. Fungsi ini digunakan untuk melihat daftar mahasiswa yang ditambahkan oleh orang tua/wali dari mahasiswa tersebut. *Method* yang digunakan pada fungsi *read* mahasiswa adalah *method GET* dengan parameter `no_hp` dari orang tua/wali yang terdaftar pada sistem dengan URL yang dipanggil adalah `parent-read-students.php`.

### 3.1.4 Sprint Retrospective

Pada *sprint retrospective* seluruh anggota tim berkumpul dan berdiskusi membahas masalah dan kendala yang dihadapi dan menyampaikan solusinya masing-masing pada *sprint* 2. Masalah yang dihadapi pada *sprint* 2 yaitu kesulitan dalam mengimplementasikan fungsi *include* Wordpress 5.3.2. Solusi yang dilakukan yaitu lebih rajin membaca dokumentasi tentang Wordpress.

## 3.3. Sprint ke-3

### 3.1.5 Sprint Planning

*Sprint planning* ke – 3 berfokus pada fitur jadwal kegiatan pada jurusan seperti jadwal kuliah, praktikum, seminar, maupun jadwal kalender akademik Universitas Lampung. Pengembangan fungsi pada *sprint* ini bertujuan agar mahasiswa, dosen atau orang tua/wali dapat melihat jadwal – jadwal yang akan dilaksanakan selama masa belajar berlangsung.

### 3.1.6 Daily Scrum

Pada *sprint* ke – 3 *product backlog* pertama yaitu pengembangan fungsi *read* jadwal. Pada fungsi ini memiliki beberapa jadwal yang harus dibaca pada aplikasi *mobile* Android modul *student*, modul *lecturer*, dan modul *parent* dengan parameter yang berbeda. Total keseluruhan URL pada *product backlog* ini adalah 9 URL. Beberapa URL dapat digunakan bersamaan di setiap modul aplikasi *mobile* dan juga hanya dapat digunakan pada satu modul aplikasi *mobile*.

URL pertama yaitu `/read-jadwal-mhs.php` yang digunakan pada aplikasi *mobile* modul *student* untuk membaca jadwal kuliah mahasiswa. Pada URL ini menggunakan *method GET* dengan parameter `hari`, `kodeMK`, `tahun_akademik`, `semester`, dan `type`.

URL kedua yaitu `read-seminar.php` digunakan pada aplikasi *mobile* modul *student* untuk membaca jadwal seminar yang ada pada jurusan masing–masing mahasiswa. Kehadiran seminar mahasiswa lain pada Fakultas MIPA Universitas Lampung digunakan untuk persyaratan untuk melakukan seminar proposal ataupun seminar hasil dari penelitian mahasiswa tersebut. Pada URL ini menggunakan *method GET* dengan parameter `jenis`, `jurusan`, dan `tgl`.

URL ketiga yaitu `read-daftar-dosen.php` digunakan pada aplikasi *mobile* modul *student* untuk membaca daftar dosen jurusan masing–masing dari setiap mahasiswa. Pada URL ini menggunakan *method GET* dengan parameter `jurusan`.

URL keempat yaitu `read-list-jadwal-dsn.php` digunakan pada aplikasi *mobile* modul *student* untuk membaca daftar jadwal dosen yang ingin dilihat. Pada URL ini menggunakan *method GET* dengan parameter `nip`, `tahun_akademik`, dan `semester`.

URL kelima yaitu *read-seminar-dsn.php* digunakan pada aplikasi *mobile* modul *lecturer* untuk membaca jadwal seminar mahasiswa yang akan dihadiri dosen sebagai pembimbing ataupun membahas dari mahasiswa tersebut. Pada URL ini menggunakan *method GET* dengan parameter *tgl* dan *nip*.

URL keenam yaitu *read-jadwal-dsn.php* digunakan pada aplikasi *mobile* modul *lecturer* untuk membaca jadwal kuliah dosen. Pada URL ini menggunakan *method GET* dengan parameter *hari*, *nip*, *tahun\_akademik*, *semester*, dan *type*.

URL ketujuh yaitu *read-jadwal-mhs-parent.php* digunakan pada aplikasi *mobile* modul *parent* untuk membaca jadwal kuliah mahasiswa yang telah didaftarkan oleh orang tua tersebut. Pada URL ini menggunakan *method GET* dengan parameter *hari*, *npm*, *tahun\_akademik*, dan *semester*.

URL kedelapan yaitu *read-daftar-ruangan.php* digunakan pada aplikasi *mobile* modul *student* dan *lecturer* untuk membaca daftar ruangan yang sesuai jurusan masing-masing mahasiswa maupun dosen. Pada URL ini menggunakan *method GET* dengan parameter jurusan.

URL kesembilan yaitu *read-jadwal-ruang.php* digunakan pada aplikasi *mobile* modul *student* dan *lecturer* untuk membaca jadwal ruangan yang ada di setiap jurusan masing-masing mahasiswa maupun dosen. Pada URL ini menggunakan *method GET* dengan parameter *ruang*, *hari*, *semester*, dan *tahun\_akademik*.

Selanjutnya *product backlog* kedua dari *sprint* ke – 3 adalah Pengembangan fungsi *READ* kalender akademik. Pada fungsi ini dapat digunakan untuk semua modul aplikasi *mobile* SIMIPA. URL pada fungsi ini yaitu *read-kalender-akademik.php* yang digunakan untuk membaca kalender akademik yang sedang aktif di Universitas Lampung. *Method* yang digunakan adalah *method GET* dengan parameter *semester* dan *tahun\_akademik*.

### 3.1.7 Sprint Review

Pada tahap ini seluruh tim berkumpul dan berdiskusi untuk membahas hal yang telah dilakukan pada *sprint* 3. Tim *development* mempresentasikan hasil yang sudah dikerjakan selama *sprint* 3. Tim *web service* menyampaikan kepada tim *development mobile* agar lebih banyak komunikasi agar data ataupun *response* yang terima sesuai yang dibutuhkan oleh *mobile apps* Android, sehingga tidak banyak yang harus diperbaiki selanjutnya. *Review* yang telah disampaikan tim *web service* diterima oleh seluruh *role scrum* yang terlibat pada *sprint* 3. Pada *sprint* ke 3 tidak ada penambahan data ataupun fungsi REST API.

### 3.1.8 Sprint Retrospective

Pada *sprint retrospective* seluruh anggota tim berkumpul dan berdiskusi membahas masalah dan kendala yang dihadapi dan menyampaikan solusinya masing-masing pada *sprint* 3. *Sprint* 3 dikatakan selesai karena setelah *sprint review* dilakukan, tim pengembang, *scrum master*, ataupun *product owner* telah menyetujui hasil *product backlog* yang telah dikerjakan, sehingga dapat melanjutkan pada *sprint* ke – 4.

## 3.4. Sprint ke-4

### 3.1.9 Sprint Planning

Pada *Web* *fmipa.unila.ac.id* memiliki sebuah layanan yang digunakan untuk mahasiswa Fakultas MIPA, antara lain layanan akademik, kemahasiswaan, dan umum & keuangan, serta memiliki *form* untuk mengajukan beasiswa dan prestasi yang dimiliki mahasiswa tersebut. *Sprint planning* ke – 4 ini berfokus pada monitoring layanan yang telah diajukan pada masing-masing bidang melalui *smartphone* Android mahasiswa, serta orang tua dapat memonitoring beasiswa atau prestasi apa yang dimiliki anaknya.

### 3.1.10 Daily Scrum

Pada *sprint* ke – 4 URL pada masing–masing fungsi memiliki 1 URL, sehingga total URL pada *sprint* ke – 4 adalah 3. Fungsi pertama yaitu fungsi *read* layanan, URL yang digunakan adalah *read-layanan.php*. URL ini digunakan untuk membaca atau memonitoring layanan yang telah mahasiswa ajukan. *Method* yang digunakan yaitu *method GET* dengan parameter *npm* dan jenis.

Fungsi kedua yaitu fungsi *read* prestasi, URL yang digunakan adalah *read-prestasi.php*. URL ini digunakan untuk membaca prestasi yang dimiliki mahasiswa pada aplikasi *mobile* modul *student* dan modul *parent*. *Method* yang digunakan yaitu *method GET* dengan parameter *npm* dan kategori. Fungsi ketiga yaitu fungsi *read* beasiswa, URL yang digunakan adalah *read-beasiswa.php*. URL ini digunakan untuk membaca beasiswa yang dimiliki mahasiswa pada aplikasi *mobile* modul *student* dan modul *parent*. *Method* yang digunakan yaitu *method GET* dengan parameter *npm* dan kategori.

### 3.1.11 Sprint Review

Pada tahap ini seluruh tim berkumpul dan berdiskusi untuk membahas hal yang telah dilakukan pada *sprint* 4. Tim *development* mempresentasikan hasil yang sudah dikerjakan selama *sprint* 4. Tim *web service* menyampaikan 3 hasil REST API yang akan digunakan pada *mobile apps*. REST API tersebut disetujui sehingga tidak ada *retrospective* dari tim dan *scrum master*.

### 3.1.12 Sprint Retrospective

Pada *sprint retrospective* seluruh anggota tim berkumpul dan berdiskusi membahas masalah dan kendala yang dihadapi dan menyampaikan solusinya masing-masing pada *sprint* 4. Tidak ada masalah yang dihadapi pada *sprint* 4, sehingga dapat melanjutkan *sprint backlog* selanjutnya.

## 3.5. Sprint ke-5

### 3.1.13 Sprint Planning

FMIPA Universitas Lampung terdapat suatu proses bisnis yang harus diselesaikan mahasiswa agar dapat melanjutkan tahap selanjutnya dalam menyelesaikan studi. Proses bisnis itu merupakan menghadiri suatu seminar kerja praktek, proposal, dan hasil. Sehingga dalam *sprint planning* ke – 5 berfokus pada proses bisnis tersebut untuk mendapatkan rekapitulasi kehadiran seminar.

### 3.1.14 Daily Scrum

Pengembangan fungsi CRUD kehadiran seminar memiliki 4 URL yang digunakan. URL pertama yaitu */read-list-seminar.php* digunakan untuk membaca seminar yang akan berlangsung yang ingin dihadiri oleh mahasiswa. Menggunakan *method GET* dengan parameter *tgl*. URL kedua */create-peserta-seminar.php* digunakan untuk menambah data kehadiran mahasiswa ke seminar yang ingin dihadiri. Menggunakan *method POST* dengan parameter *npm* dan *id\_seminar*. URL ketiga */read-cek-hadir-seminar.php* digunakan untuk mengecek apakah mahasiswa sudah memilih hadir atau tidak. Menggunakan *method GET* dengan parameter *npm*. Kemudian URL ke-4 */delete-peserta-seminar.php* digunakan untuk membatalkan hadir mahasiswa dalam seminar yang sebelumnya ingin dihadiri sebelum divalidasi oleh dosen. Menggunakan *method POST* dengan parameter *npm* dan *id\_seminar*.

Pengembangan fungsi rekapitulasi seminar memiliki 2 URL yang akan digunakan yaitu, pertama */read-rekapitulasi-seminar-mhs.php* digunakan untuk membaca seminar mahasiswa yang pernah dihadiri dan telah divalidasi oleh dosen. Kedua */read-jumlah-seminar.php* digunakan untuk membaca berapa jumlah seminar yang telah dihadiri oleh mahasiswa. *Method* yang digunakan pada kedua URL diatas adalah *method GET*, URL pertama dengan parameter *npm*, kemudian URL kedua dengan parameter *npm* dan jenis.

### 3.1.15 Sprint Review

Pada tahap ini seluruh tim berkumpul dan berdiskusi untuk membahas hal yang telah dilakukan pada *sprint* 5. Tim *development* mempresentasikan hasil yang sudah dikerjakan selama *sprint* 5. Pada *review* ini *scrum master* meminta untuk menambahkan rekapitulasi seminar mahasiswa untuk dosen pembimbing akademik dan pembimbing skripsi dalam bentuk grafik, sehingga dijadikan *sprint retrospective* pada tim *web service*.

### 3.1.16 Sprint Retrospective

Pada *sprint retrospective* seluruh anggota tim berkumpul dan berdiskusi membahas masalah dan kendala yang dihadapi dan menyampaikan solusinya masing-masing pada *sprint* 5. Hasil dari penambahan rekapitulasi seminar mahasiswa untuk dosen pembimbing akademik dan pembimbing skripsi disampaikan pada tahap ini. Terdapat 5 URL dalam penambahan fungsi rekapitulasi. Pertama yaitu */read-daftar-mhs.php* digunakan untuk membaca daftar mahasiswa yang terdaftar pada dosen pembimbing akademik sedangkan URL kedua untuk pembimbing skripsi menggunakan URL */read-mahasiswa-bimbingan-skripsi.php*. Kedua URL menggunakan *method GET* dengan parameter *nip*.

Kemudian URL ketiga adalah untuk menampilkan rekapitulasi seminar mahasiswa yang telah dilakukan oleh mahasiswa bimbingan PA atau Skripsi. URL yang digunakan yaitu */read-rekapitulasi-seminar.php* dengan *method GET* dengan parameter *npm*.

URL keempat adalah */read-rekapitulasi-grafik-pa.php* digunakan untuk membaca data yang diperlukan untuk menampilkan grafik seminar mahasiswa bimbingan PA pada bagian *front end*. Sedangkan untuk grafik seminar mahasiswa bimbingan skripsi menggunakan URL kelima yaitu */read-rekapitulasi-grafik-skripsi.php*. kedua URL diatas menggunakan *method GET* dengan parameter *nip*. Pada fungsi grafik ini menggunakan *rowCount()* dari fungsi *read* rekapitulasi seminar mahasiswa sehingga mendapatkan jumlah data yang ada. Setelah *retrospective* telah disetujui oleh *scrum master* dan tim pengembang, tim *web service* melanjutkan ke *sprint* selanjutnya yaitu *sprint* ke – 6.

## 3.6. Sprint ke-6

### 3.1.17 Sprint Planning

*Sprint planning* ke – 6 mengembangkan fungsi cek *update* dan fungsi *read* berita. Fungsi cek *update* ini digunakan untuk mengecek apakah aplikasi yang terinstal merupakan versi terbaru yang telah terdaftar pada *database*. Kemudian fungsi *read* berita digunakan untuk membaca berita yang dipost pada web *fmipa.unila.ac.id*.

### 3.1.18 Daily Scrum

Pengembangan fungsi cek *update* menghasilkan URL */read-cek-update.php*. *Method* yang digunakan adalah *method GET* dengan parameter versi, tahun, kode dan *apps*.

Pengembangan fungsi *read* berita memiliki 2 URL, yaitu URL pertama */read-berita.php* digunakan untuk membaca berita yang aktif dan URL kedua */read-berita-foto.php* digunakan untuk membaca foto atau poster dalam berita yang aktif dengan menggunakan *method GET* dengan parameter *id* dari berita yang telah dibaca.

### 3.1.19 Sprint Review

Pada tahap ini seluruh tim berkumpul dan berdiskusi untuk membahas hal yang telah dilakukan pada *sprint* 6. Tim *development* mempresentasikan hasil yang sudah dikerjakan selama *sprint* 6. Pada *sprint review* kali ini *product owner* meminta penambahan 8 *product backlog* yang dibagi menjadi 3 *sprint* untuk dikerjakan tim *web service*.

### 3.1.20 Sprint Retrospective

Pada *sprint retrospective* seluruh anggota tim berkumpul dan berdiskusi membahas masalah dan kendala yang dihadapi dan menyampaikan solusinya masing-masing pada *sprint 6*. Tidak ada masalah yang dihadapi pada *sprint 6*, sehingga dapat melanjutkan *sprint backlog* yang telah ditambahkan oleh *product owner*

## 3.7. Sprint ke-7

### 3.1.21 Sprint Planning

*Sprint planning* ke – 7 membahas penambahan *product backlog* yang telah disepakati.

### 3.1.22 Daily Scrum

Pengembangan fungsi tanggal penting pada modul *parent* menghasilkan URL */read-tanggal-penting.php* digunakan untuk membaca tanggal penting yang akan datang dari tanggal aplikasi sedang digunakan. *Method* yang digunakan adalah *method GET* dengan parameter semester, tahun\_akademik, dan tgl.

Selanjutnya pengembangan fungsi validasi orang tua modul *student* terdapat 4 URL untuk menampilkan daftar, memvalidasi, menolak, dan menghapus orang tua. URL pertama adalah */read-list-parent.php* digunakan untuk menampilkan daftar orang tua yang mendaftarkan npm. *Method* yang digunakan adalah *method GET* dengan parameter npm. Selanjutnya URL kedua adalah */update-validasi-parent.php* digunakan untuk memvalidasi orang tua. URL ketiga */update-tolak-parent.php* digunakan untuk menolak orang tua. URL keempat */update-delete-parent.php* digunakan untuk menghapus orang tua. URL kedua, ketiga, dan keempat menggunakan *method POST* dengan parameter id dan *user*.

Selanjutnya pengembangan fungsi validasi kehadiran terdapat 2 URL yang digunakan untuk membaca daftar mahasiswa yang telah hadir dan digunakan untuk memvalidasi daftar hadir mahasiswa tersebut. URL pertama adalah */read-list-peserta-seminar.php* menggunakan *method GET* dengan parameter id\_seminar. Sedangkan URL kedua adalah */update-kehadiran-seminar.php* menggunakan *method POST* dengan parameter id\_seminar dan npm.

### 3.1.23 Sprint Review

Pada tahap ini seluruh tim berkumpul dan berdiskusi untuk membahas hal yang telah dilakukan pada *sprint 7*. Tim *development* mempresentasikan hasil yang sudah dikerjakan selama *sprint 7*. *Product owner* menyetujui hasil yang dikerjakan dan tidak terdapat perubahan dari *sprint backlog* yang telah dikerjakan.

### 3.1.24 Sprint Retrospective

Pada *sprint retrospective* seluruh anggota tim berkumpul dan berdiskusi membahas masalah dan kendala yang dihadapi dan menyampaikan solusinya masing-masing pada *sprint 7*. Tidak terdapat masalah dari *sprint backlog* yang telah dikerjakan, sehingga dapat melanjutkan ke *sprint backlog* selanjutnya

## 3.8. Sprint ke-8

### 3.1.25 Sprint Planning

*Sprint planning* ke – 8 membahas beberapa sisa *product backlog* baru yang diminta oleh *product owner* apakah fitur-fitur baru tersebut perlu diterapkan ke versi *mobile apps* atau tidak.

### 3.1.26 Daily Scrum

Pengembangan fungsi token notifikasi memiliki 2 URL yaitu, pertama */read-token.php* digunakan untuk membaca token dari *user* yang ingin dikirimkan notifikasi. *Method GET* digunakan untuk

parameter *user* dan *imei*. Kemudian URL kedua adalah `/update-token.php` digunakan untuk mengupdate token ketika *user* masuk aplikasi. URL ini menggunakan *method POST* dengan parameter *user*, *imei*, dan *token*.

Kemudian pengembangan fungsi pengajuan beasiswa menghasilkan URL `/create-beasiswa.php` digunakan untuk memasukkan data beasiswa mahasiswa ke SIMIPA. *Method* yang digunakan adalah *method POST* dengan parameter *npm*, *semester*, *tahun\_beasiswa*, *penyelenggara* *nama\_beasiswa*, dan *file*.

Pengembangan fungsi pengajuan prestasi mahasiswa merupakan turunan fitur dari pengajuan prestasi yang sudah ada di *web* `fmipa.unila.ac.id`. Untuk aplikasi *mobile* SIMIPA modul *student* fitur yang diterapkan pada pengajuan prestasi yaitu untuk individu mahasiswa tersebut, belum mencakup fitur pengajuan prestasi kelompok. URL yang digunakan adalah `/create-prestasi-mhs.php` dengan *method POST* menggunakan parameter *nama\_kegiatan*, *penyelenggara*, *kategori*, *tingkat*, *prestasi*, *tahun\_pelaksanaan*, *jenis\_peserta*, dan *sertifikat*.

### 3.1.27 Sprint Review

Pada tahap ini seluruh tim berkumpul dan berdiskusi untuk membahas hal yang telah dilakukan pada *sprint* 8. Tim *development* mempresentasikan hasil yang sudah dikerjakan selama *sprint* 8. Terdapat beberapa *review* yang diberikan oleh *product owner*, *scrum master*, dan tim pengembang pada *sprint* 8 yaitu menggabungkan fungsi update token pada URL login mahasiswa, dosen atau *parent* dan menambah URL untuk menampilkan daftar beasiswa yang tersedia agar mahasiswa memilih beasiswa sesuai daftar beasiswa tersebut.

### 3.1.28 Sprint Retrospective

Pada *sprint retrospective* seluruh anggota tim berkumpul dan berdiskusi membahas masalah dan kendala yang dihadapi dan menyampaikan solusinya masing-masing pada *sprint* 8. Tidak ada masalah yang terjadi pada *sprint* 8 namun memiliki beberapa perbaikan yang disampaikan pada *sprint review* sebelumnya. Fungsi update token sudah menjadi satu dengan URL fungsi *login*, sehingga pada saat *user* masuk aplikasi menggunakan *user* dan *password* yang terdaftar maka akan mengupdate *token* sesuai *device* yang digunakan. Kemudian untuk membaca daftar beasiswa yang tersedia menggunakan URL `/read-daftar-beasiswa-aktif.php` menggunakan *method GET* dengan parameter tanggal.

## 3.9. Sprint ke-9

### 3.1.29 Sprint Planning

*Sprint planning* ke – 9 membahas fitur atau *product backlog* terakhir dalam pengembangan *web service* SIMIPA. *Product backlog* tersebut adalah pengembangan fungsi pengesahan tema skripsi mahasiswa (*apps lecturer*) dan menambahkan keamanan dengan JSON Web Token (JWT).

### 3.1.30 Daily Scrum

Pengembangan fungsi pengesahan tema skripsi (*apps lecturer*) menghasilkan 3 URL. URL pertama yaitu `/read-tema-mahasiswa.php` digunakan untuk membaca tema skripsi mahasiswa untuk dosen pembimbing satu. *Method* yang digunakan adalah *method GET* dengan parameter *nip*. URL kedua yaitu `/read-tema-mahasiswa-pa.php` digunakan untuk membaca tema skripsi mahasiswa untuk dosen pembimbing akademik. *Method* yang digunakan adalah *method GET* dengan parameter *nip*.

URL terakhir yaitu `/validasi-tema-mahasiswa.php` digunakan untuk membaca tema skripsi mahasiswa untuk dosen pembimbing satu atau pembimbing akademik. *Method* yang digunakan adalah *method POST* dengan parameter *id*, *nip*, *validasi*, dan *dosen*.

Kemudian security POST menggunakan JSON Web Token (JWT) ditambahkan pada REST API agar sistem melakukan autentikasi pertukaran data pada parameter HTTP POST.

3.1.31 *Sprint Review*

Pada tahap ini seluruh tim berkumpul dan berdiskusi untuk membahas hal yang telah dilakukan pada *sprint* 9. Tim *development* mempresentasikan hasil yang sudah dikerjakan selama *sprint* 9. Review yang dilakukan membahas *sprint* 9 sekaligus semua *product backlog* yang telah dikerjakan agar semua dapat berjalan sesuai kebutuhan, sehingga tim *web service* dapat melanjutkan tahapan berikutnya.

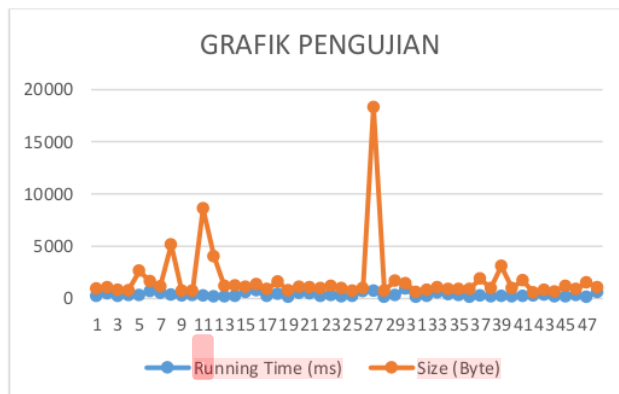
*Sprint review* selesai dan tidak ada argumen dari tim *scrum* terkait perubahan, penambahan ataupun menghapus *product backlog* yang telah dikerjakan, sehingga tim *web service* telah menyelesaikan semua *product backlog* yang ditentukan.

3.1.32 *Sprint Retrospective*

Pada *sprint retrospective* seluruh anggota tim berkumpul dan berdiskusi membahas masalah dan kendala yang dihadapi dan menyampaikan solusinya masing-masing pada *sprint* 9. Tidak terdapat masalah pada *sprint* 9 sehingga pengembangan *web service* SIMIPA telah selesai dilakukan. Kemudian akan dilakukan pengujian menggunakan metode *Black Box Testing*.

3.10. Pengujian *Black Box Testing*

Skenario pengujian penelitian ini dengan mengeksekusi *request* API menggunakan *software* POSTMAN melalui metode pengujian *black box testing*. *Rules* yang digunakan adalah melakukan 10 kali *request* API untuk menentukan dan mengetahui keseluruhan fungsional yang dikerjakan pada *service* berjalan sesuai keinginan atau diperlukan adanya perubahan agar REST API yang dikerjakan lebih maksimal saat digunakan pada bagian *mobile apps* Android.



Gambar 6. Grafik Pengujian REST API SIMIPA.

Pada Gambar 6 dapat dilihat bahwa rata – rata *running time* REST API SIMIPA stabil dengan rata – rata yang didapat paling rendah 155 ms pada pengujian REST API logout dan paling tinggi 965,8 ms yaitu pada REST API pengajuan beasiswa. Pada pengajuan beasiswa upload *file* gambar untuk di upload ke penyimpanan dan *database* SIMIPA, sehingga memerlukan waktu yang lebih lama. Kemudian untuk *size* yang diterima saat melakukan pengujian tergantung data yang tersedia pada *database* SIMIPA, jika dilihat pada grafik data yang memiliki *size* paling besar yaitu 1750 Byte pada REST API untuk membaca berita, sedangkan *size* yang paling kecil yaitu 287,1 Byte terdapat pada pengujian REST API untuk membaca absen seminar. Pada Gambar 6 dapat dilihat bahwa rata – rata *running time* REST API SIMIPA stabil dengan rata – rata yang didapat paling rendah 155 ms pada pengujian REST API logout dan paling tinggi 965,8 ms yaitu pada REST API pengajuan beasiswa. Pada pengajuan beasiswa upload *file* gambar untuk di upload ke penyimpanan dan *database* SIMIPA, sehingga memerlukan waktu yang lebih lama. Kemudian untuk *size* yang diterima saat melakukan pengujian tergantung data yang tersedia pada *database* SIMIPA, jika dilihat pada grafik data yang memiliki *size* paling besar yaitu 1750 Byte

©2021 Ilmu Komputer Unila Publishing Network All Rights Reserved

pada REST API untuk membaca berita, sedangkan *size* yang paling kecil yaitu 287,1 Byte terdapat pada pengujian REST API untuk membaca absen seminar. Pada Gambar 6 dapat dilihat bahwa rata – rata *running time* REST API SIMIPA stabil dengan rata – rata yang didapat paling rendah 155 ms pada pengujian REST API logout dan paling tinggi 965,8 ms yaitu pada REST API pengajuan beasiswa. Pada pengajuan beasiswa upload *file* gambar untuk di upload ke penyimpanan dan *database* SIMIPA, sehingga memerlukan waktu yang lebih lama. Kemudian untuk *size* yang diterima saat melakukan pengujian tergantung data yang tersedia pada *database* SIMIPA, jika dilihat pada grafik data yang memiliki *size* paling besar yaitu 1750 Byte pada REST API untuk membaca berita, sedangkan *size* yang paling kecil yaitu 287,1 Byte terdapat pada pengujian REST API untuk membaca absen seminar.

#### 4. SIMPULAN

Pengembangan *web service* menggunakan metode *scrum* dengan *sprint* sebanyak 9 iterasi. Jumlah *product backlog* pada tim *web service* yang dibuat oleh *product owner* terdapat 26 item. Hasil dari penelitian ini berupa sebuah REST API yang digunakan sebagai jembatan antara *server* yaitu FMIPA dan *client* yaitu *mobile apps* Android dengan menggunakan URL. Pengujian REST API SIMIPA untuk mendapatkan kecepatan akses yang diharapkan dilakukan 10 kali percobaan dengan hasil rata - rata *running time* yang didapat adalah 155 ms yang paling rendah dan 965,8 ms yang paling tinggi. REST API SIMIPA sudah digunakan dan di uji pada *mobile apps* Android SIMIPA Modul *Student*, SIMIPA Modul *Lecturer* dan SIMIPA Modul *Parent*.

#### DAFTAR PUSTAKA

- [1] Sommerville, I. (2016). *Software Engineering Edisi 9*. London: Pearson Education.
- [2] Prakasa, M Iqbal dan Setiawan, Eko Budi. (2018). Pengembangan *Web Service* Data Masyarakat Menggunakan REST API dengan *Access Token*. *ULTIMA Computing*. Vol. X. No.1.
- [3] Webber, J., Parastatidis, S. dan Robinson, I. (2010). *REST in Practice Hypermedia And Systems Architecture*. Gravenstein Highway North, Sebastopol, CA(USA): O'Reilly Media.Inc.
- [4] Winarno, E. dan Ali, Z. S. (2014). *Pemrograman Web Berbasis HTML5, PHP, dan JavaScript*. Jakarta: PT Elex Media Komputindo.
- [5] Kadir, A. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. Yogyakarta: Andi.
- [6] Ardiansyah, H. I. (2017). Pengembangan REST API Sistem Informasi Marketplace Pariwisata Indoexplore.id. *Central Library of Bogor Agricultural University*.
- [7] M. A. Dewi dan R. Andriani, "Implementasi Scrum Model Development Pada Monitoring Inventory Control Cleaning Equipment dan Chemical PT. Explore Global Solution," *ULTIMA Infosys*, Vol. VIII, No. 2, pp. 112-117, 2017.
- [8] J. Arka, A. H. Brata dan K. C. Brata, "Pengembangan Aplikasi Mobile Manajemen Keuangan Dengan Metode Scrum (Studi Kasus Mahasiswa FILKOM UB)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Vol. 3, No. 2, pp. 1199-1207, 2019.
- [9] K. Bhavsar, V. Shah dan S. Gopalan, "Scrum: An Agile Process Reengineering In Software Engineering," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, Vol. 9, No. 3, pp. 840-848, 2020.
- [10] B. Septian, I. K. Jayadi, M. Holil dan I. Handriani, "Sistem Human Capital Management Menggunakan Metode Scrum," *JUST IT: Jurnal Sistem Informasi, Teknologi Informatika dan Komputer*, Vol. 11, No. 1, pp. 1-16, 2020.
- [11] P. A. G. Permana "Scrum Method Implementation in a Software Development Project Management," *International Journal of Advanced Computer Science and Applications*, Vol. 6, No. 9, pp. 198-204, 2015.



©2021 Ilmu Komputer Unila Publishing Network All Rights Reserved

- [12] S. R. Jan, S. T. U. Shah, Z. U. Johar, Y. Shah, dan F. Khan, "An Innovative Approach to Investigate Various Software Testing Techniques and Strategies," *IJRSET*, Vol. 2, No. 2, pp. 682–689, 2016.
- [13] Sudirman. (2016). Analisis Komunikasi Data Dengan Xml Dan Json Pada Webservice. *CESS Journal of Computer Engineering, System and Science*, 1(2), 2502–7131.
- [14] M.A. Rahman, I. Kuswardayan dan R. R. Hariadi. "Perancangan dan Implementasi RESTful Web Service untuk Game Sosial Food Merchant Saga pada Perangkat Android," *Jurnal Teknik POMITS* Vol. 2, No. 1, pp. 1-4, 2013

ardi

---

ORIGINALITY REPORT

---

96%

SIMILARITY INDEX

96%

INTERNET SOURCES

6%

PUBLICATIONS

11%

STUDENT PAPERS

---

PRIMARY SOURCES

---

1

[jurnal.fmipa.unila.ac.id](http://jurnal.fmipa.unila.ac.id)

Internet Source

96%

---

2

[etheses.uin-malang.ac.id](http://etheses.uin-malang.ac.id)

Internet Source

<1%

---

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off