

## PENGEMBANGAN *WEB SERVICE* MODUL MAHASISWA PADA SISTEM INFORMASI AKADEMIK UNIVERSITAS NEGERI JAKARTA

Febrianto Widyoutomo<sup>1</sup>, Hamidillah Ajie<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> widyoutomof@gmail.com, <sup>2</sup> hamidillah@unj.ac.id, <sup>3</sup> widodo@unj.ac.id

---

### Abstrak

*Perkembangan teknologi sebagai bagian dari inovasi yang dapat memberikan banyak kemudahan dalam berbagai bidang. Penerapan teknologi dalam bidang pendidikan mendorong terciptanya sistem informasi yang dapat mengelola berbagai data, baik data akademik maupun non-akademik. Universitas Negeri Jakarta (UNJ) menjadi salah satu perguruan tinggi yang memanfaatkan sistem informasi sebagai sarana pengelola data akademik yang dinamakan SIAKAD UNJ. SIAKAD UNJ saat ini tidak memiliki arsitektur sistem yang maksimal, sehingga belum bersifat adaptif terhadap pengembangan dan pembaruan fitur di dalam sistem. Tujuan dari penelitian ini adalah untuk menghasilkan web service yang dapat digunakan dalam SIAKAD UNJ dan memiliki sifat adaptif terhadap perubahan-perubahan di masa mendatang. Pengembangan web service SIAKAD UNJ dengan metode spiral menggunakan framework Express.js dan menghasilkan 25 endpoint. Pengujian dengan metode unit testing terhadap 25 endpoint menggunakan aplikasi bernama Postman. Hasil pengujian menunjukkan bahwa 25 endpoint yang diuji berfungsi dengan baik, dan layak digunakan sebagai sisi backend yang akan digabungkan dengan sisi frontend untuk menjadi aplikasi SIAKAD UNJ yang utuh.*

**Kata kunci** : sistem informasi akademik, web service, metode spiral, express, endpoint, unit testing.

---

### 1. Pendahuluan

#### 1.1. Latar Belakang

Setiap inovasi hadir untuk memberikan manfaat bagi kehidupan manusia. Teknologi sebagai bagian dari inovasi yang dapat memberikan banyak kemudahan, serta sebagai penunjang hadirnya inovasi-inovasi lain. Oleh karenanya, teknologi menjadi hal yang tidak bisa dihindari dalam kehidupan ini. Saat ini, teknologi dimanfaatkan dalam berbagai bidang, seperti bidang pendidikan, administrasi dan lain-lain.

Teknologi yang berkembang mempunyai pengaruh terhadap bidang pendidikan, salah satunya yaitu perguruan tinggi. Penerapan teknologi pada perguruan tinggi berupa sistem untuk mengelola layanan informasi dan administrasi akademik, layanan informasi kurikulum, layanan penerimaan peserta didik baru dan banyak lainnya agar lebih terorganisir, efektif, dan efisien.

Universitas Negeri Jakarta (UNJ) menjadi salah satu perguruan tinggi negeri di Indonesia yang dalam perkembangan penyelenggaraan program pendidikan mulai dari jenjang Diploma, Sarjana, Magister hingga Doktor. Tidak bisa dipungkiri untuk dapat memberikan informasi dan layanan sebagai penunjang keberlangsungan akademik yang terorganisir, efektif dan efisien, Universitas Negeri

Jakarta memanfaatkan teknologi untuk mengatasi hal itu. Maka dibuatlah Sistem Informasi Akademik (SIAKAD) pada tahun 1980-an.

Berdasarkan hasil wawancara kepada pimpinan dan karyawan Unit Pelayanan Teknis Teknologi Informasi dan Komunikasi (UPT TIK) UNJ, SIAKAD UNJ saat itu berfungsi sebagai penyimpanan data akademik dalam bentuk digital. Seiring berjalannya waktu, pada tahun 2000 SIAKAD UNJ telah menggunakan sistem berbasis IBM AS400 yang fungsinya mampu mengelola data akademik, seperti pengisian jadwal kuliah, pengisian Kartu Rencana Studi (KRS) dan penyimpanan nilai akhir semester mahasiswa.

Pengembangan SIAKAD UNJ berbasis web yang memiliki banyak fitur untuk berbagai pengguna ini akan dilakukan secara bertahap, salah satunya membagi secara modular sesuai dengan tugas dan wewenang dari masing-masing pengguna, seperti modul mahasiswa, modul dosen, modul administrasi program studi dan lain sebagainya. Tujuan dibuatnya modul-modul ini adalah membuat pengembangan terfokus pada bagian-bagian kecil sehingga pengembangan menjadi lebih sederhana dan efisien serta yang paling utama untuk memudahkan pembuatan dokumentasi atas pengembangan-pengembangan yang terjadi.

Available at:

<http://journal.unj.ac.id/unj/index.php/pinter/article/view/23587>

Modul Mahasiswa menjadi komponen penting dalam SIAKAD UNJ, karena proses akademik ini menyangkut administrasi mahasiswa dalam pembelajaran tiap semesternya. Modul ini ada untuk mahasiswa memproses administrasi akademik secara digital. Adapun layanan yang diberikan di dalam modul mahasiswa, yaitu pengelolaan biodata mahasiswa, pengisian KRS, unduh KHS pada tiap semester yang telah diselesaikan dan bentuk administrasi akademik lainnya.

Desain arsitektur SIAKAD UNJ saat ini belum membagi sistem menjadi dua sisi, yaitu sisi *backend* dan sisi *frontend*. Desain semacam ini mempunyai tingkat adaptasi aplikasi web yang rendah, karena apabila ada pengembangan dan pembaruan terhadap fitur-fitur SIAKAD UNJ akan mempengaruhi aplikasi secara keseluruhan dikarenakan sisi *presentation* dan *business logic* ada pada satu kesatuan dan akan saling mempengaruhi. Atas dasar tersebut, maka terdapat beberapa pilihan alternatif untuk mengembangkan aplikasi web yang memiliki adaptasi terhadap perubahan dan pengembangan fitur-fitur di dalam SIAKAD UNJ salah satunya membuat arsitektur aplikasi menjadi tiga sisi, yaitu sisi *frontend*, sisi *backend*, dan sisi *database*.

*Web service* yang menjadi peran utama dalam *backend* adalah sistem pertukaran informasi dengan memanfaatkan jaringan internet untuk interaksi antar aplikasi dan perangkat dengan format standar seperti *text*, *eXtensible Markup Language* (XML), *JavaScript Object Notation* (JSON). *Web service* adalah sebuah sistem yang didesain untuk mendukung interoperabilitas interaksi mesin ke mesin melalui sebuah jaringan. *Web service* dapat diartikan juga sebuah metode pertukaran data tanpa memperhatikan di mana sebuah *database* ditanamkan, dalam bahasa apa sebuah aplikasi yang menggunakan data tersebut, dan di platform apa sebuah data digunakan.

Pemanfaatan *web service* pada SIAKAD UNJ akan membuat aplikasi web SIAKAD UNJ menjadi lebih adaptif terhadap pengembangan dan pembaruan fitur-fitur. Dengan menggunakan *web service* memungkinkan perbaikan akan sistem tidak akan memengaruhi sisi *frontend*, sehingga memudahkan dalam sisi perawatan sistem. *Web service* sebagai metode pertukaran data yang tidak memperhatikan di platform apa sebuah data digunakan, atau dalam kata lain dapat digunakan secara *cross platform*.

Berdasarkan pemahaman di atas, penulis mencoba mengangkat permasalahan tersebut ke dalam studi penelitian yang berjudul "Pengembangan Web Service Modul Mahasiswa Sistem Informasi Akademik Universitas Negeri Jakarta". Harapannya agar pengembangan web Sistem Informasi Akademik Universitas Negeri Jakarta memiliki tingkat adaptasi yang baik kedepannya.

## 1.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah dibahas sebelumnya, identifikasi masalah pada penelitian ini adalah :

1. Sistem Informasi Akademik Universitas Negeri Jakarta belum memiliki arsitektur sistem yang maksimal, sehingga belum bersifat adaptif terhadap pengembangan dan pembaruan fitur di dalam sistem.
2. Masih menyatunya semua sistem untuk berbagai *user* yang memungkinkan fokus akan terbagi dalam pengembangan aplikasi.
3. Sistem Informasi Akademik Universitas Negeri Jakarta masih menerapkan desain arsitektur yang hanya membagi sistem menjadi dua sisi, yaitu *interface* dan *database* dalam satu kesatuan sehingga akan mempersulit dalam pengembangan sistem jika ada pembaruan pada sistem.

## 1.3. Pembatasan Masalah

Melihat luasnya lingkup permasalahan, agar memperoleh hasil penelitian yang optimal dan terfokus, maka pembatasan masalah sangat penting untuk dilakukan. Penelitian dibatasi pada:

1. Pada penelitian ini tahapan yang akan menjadi pokok bahasan dari rangkaian Sistem Informasi Akademik Universitas Negeri Jakarta Modul Mahasiswa.
2. Penelitian ini tidak akan menghasilkan daftar kebutuhan perangkat lunak, namun menggunakan dokumen *requirement* milik UPT TIK UNJ.
3. Penelitian ini akan menghasilkan *web service* berupa *endpoint-endpoint* dan tidak menghasilkan aplikasi web SIAKAD yang siap digunakan dengan menggunakan basis data berdasarkan standar PDDikti.

## 1.4. Rumusan Masalah

Berdasarkan latar belakang masalah, identifikasi, dan pembatasan masalah, maka perumusan masalah yang akan dibahas pada penelitian ini adalah: Bagaimana mengembangkan *web service* yang dapat digunakan dalam pengembangan Modul Mahasiswa pada Sistem Informasi Akademik Universitas Negeri Jakarta?

## 1.5. Tujuan Penelitian

Tujuan yang diharapkan akan dicapai pada penelitian adalah untuk menghasilkan *web service* yang dapat digunakan dalam pengembangan aplikasi modul mahasiswa SIAKAD Universitas Negeri Jakarta dan memiliki tingkat adaptasi yang baik terhadap perubahan-perubahan di masa mendatang.

## 1.6. Manfaat Penelitian Tujuan Penelitian

Available at:

<http://journal.unj.ac.id/unj/index.php/pinter/article/view/23587>

1. Secara teoritis, hasil dari penelitian ini diharapkan dapat dimanfaatkan sebagai referensi dan masukan bagi perkembangan ilmu pengembangan aplikasi web khususnya web programming.

2. Secara praktis, hasil dari penelitian ini diharapkan dapat digunakan sebagai sebuah backend yang akan diimplementasikan dengan frontend sehingga menghasilkan aplikasi web Sistem Informasi Akademik Universitas Negeri Jakarta secara utuh.

## 2. Dasar Teori

### 2.1. Sistem Informasi Akademik Universitas Negeri Jakarta

Sistem Informasi Akademik Universitas Negeri Jakarta (SIKAD UNJ) adalah sistem aplikasi berbasis web milik Universitas Negeri Jakarta (UNJ) yang dijadikan alat untuk mengelola kegiatan akademik di lingkup universitas. SIKAD UNJ ada guna untuk membuat proses administrasi dan berbagai kebutuhan akademik di UNJ secara digital dan dapat diakses di mana pun.

SIKAD UNJ menjadi tempat penyimpanan informasi akademik bagi mahasiswa, dosen, administrator program studi seperti biodata mahasiswa, Kartu Rencana Studi (KRS), Kartu Hasil Studi (KHS), Daftar Hasil Studi (DHS), jadwal kuliah, pembayaran, pengisian nilai, dan sebagainya. SIKAD UNJ mengelola data akademik secara keseluruhan dan terintegrasi satu sama lain antara data mahasiswa, data dosen dan data program studi, sehingga setiap perubahan terjadi pada data akan berpengaruh terhadap data lainnya secara langsung. Data yang ada pada SIKAD akan menjadi informasi akademik pusat bagi mahasiswa, dosen, administrator program studi dan fakultas, serta pimpinan universitas, sehingga dengan adanya SIKAD ini dapat membuat data tersimpan secara rapi dan terstruktur.

### 2.2. Modul Mahasiswa SIKAD UNJ

Modul Mahasiswa menjadi komponen penting dalam SIKAD UNJ, karena proses akademik ini menyangkut administrasi mahasiswa dalam pembelajaran tiap semesternya. Modul ini ada untuk mahasiswa memproses administrasi akademik secara digital. Adapun layanan yang diberikan di dalam modul mahasiswa, yaitu: (a) pengelolaan biodata mahasiswa; (b) pengelolaan akun; (c) pengelolaan KRS, seperti pengisian pada awal semester, pantau dan unduh pada tiap semester yang diikuti; (d) pantau dan unduh KHS pada tiap semester yang telah diselesaikan; (e) pantau dan unduh DHS yang memperlihatkan daftar keseluruhan hasil dari seluruh aktivitas akademik selama menjadi mahasiswa UNJ; (f) pantau jadwal kuliah tiap semester; (h)

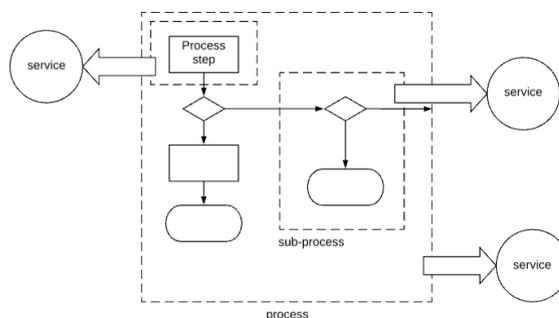
pendaftaran Praktik Keterampilan Mengajar (PKM); (i) pengisian evaluasi perkuliahan di akhir semester.

### 2.3. Sisi Backend

*Backend* adalah sisi yang bertolak belakang dengan sisi *frontend* yang berhadapan langsung dengan *client*, dimana *backend* merupakan *server side* yang mengelola data dan menjalankan *business logic* sebuah aplikasi. *Backend* bertindak sebagai arsitek yang merancang sistem dan teknologi web. *Backend* bertanggung jawab akan program yang bekerja pada server agar mengirimkan informasi ke browser pengguna setiap kali pengguna memintanya melalui HTTP *request*. Sisi *backend* juga yang berinteraksi langsung dengan *database*, sehingga frontend hanya akan menerima informasi yang bersumber dari database melalui *backend*.

### 2.4. Service Oriented Architecture (SOA)

*Service Oriented Architecture* (SOA) merupakan sekumpulan fungsi, prosedur atau proses yang akan memberi respons jika diminta oleh sebuah *client*. SOA adalah salah satu bentuk arsitektur teknologi yang mengikuti prinsip-prinsip *service computing platform* dengan membawa konsep, teknologi, dan tantangan baru. Menurut Thomas Erl konsep *service oriented* ini melakukan pendekatan dengan membagi masalah besar menjadi sekumpulan *service* kecil yang bertujuan untuk menyelesaikan permasalahan tertentu (Erl,2008).



Gambar 2.1 Enkapsulasi business process dengan service

### 2.5. Web Service Restful

*Web service* adalah sistem pertukaran informasi berbasis eXtensible Markup Language (XML) atau JavaScript Object Notation (JSON) dengan memanfaatkan jaringan internet untuk interaksi antar aplikasi dan perangkat. Teknologi ini menjadi standar yang diadopsi oleh banyak *vendor* perangkat lunak, karena memiliki standar terbuka. Standar terbuka ini yang memungkinkan aplikasi *web service* yang diimplementasi oleh *vendor* berbeda dapat berkomunikasi satu sama lain (Erl, 2008:56).

*Web service* memiliki dua bentuk, yaitu *Simple Object Access Protocol* (SOAP) dan

Available at:

<http://journal.unj.ac.id/unj/index.php/pinter/article/view/23587>

*Representational State Transfer* (REST). SOAP dan REST memiliki kesamaan di antaranya, sama-sama digunakan metode pertukaran pesan dan data dalam komunikasi dengan *web service* dan sama-sama menggunakan protokol web *Hypertext Transfer Protocol* (HTTP) dan *Hypertext Transfer Protocol Secure* (HTTPS). Namun, REST relatif lebih hemat dalam penggunaan *bandwidth*, karena *markup-markup* ekstra seperti XML dapat diganti dengan JSON, sehingga kini mulai beralih menggunakan REST.

## 2.6. Framework

Menurut Wardana (2016:3), *framework* adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu yang saling berinteraksi satu sama lain sehingga dalam pembuatan aplikasi kita harus mengikuti aturan dari *framework* tersebut. *Framework* yang dimaksud adalah kumpulan dari beberapa fungsi yang dibuat oleh orang atau perusahaan yang berguna untuk memudahkan pengembang dalam mengurangi waktu pembuatan sebuah aplikasi.

## 2.7. Express.js

Node.js adalah *platform* sisi *server* dibangun di mesin JavaScript V8 Google Chrome (*V8 Engine*) yang dikembangkan oleh Ryan Dahl pada tahun 2009. Node.js adalah aplikasi *open source* yang sepenuhnya gratis dan digunakan oleh ribuan *developer* di seluruh dunia untuk mengembangkan aplikasi sisi *server* dan jaringan. Node.js adalah sebuah *environment runtime* dari JavaScript. *Environment runtime* dari Node.js mencakup semua yang dibutuhkan pengembang web untuk menjalankan program yang ditulis dalam JavaScript (Node.js).

Express.js adalah *framework* yang bekerja pada aplikasi Node.js yang minimalis dan fleksibel. Express.js juga memiliki dokumentasi yang lengkap dan penggunaannya yang cukup mudah, dapat membuat kita mengembangkan berbagai produk seperti aplikasi web ataupun RESTful API. Express.js pun dapat digunakan menjadi pijakan untuk membangun web *framework* yang lebih kompleks seperti *sails.js*, MEAN (MongoDB, Express.js, Angular.js, Node.js) dan MERN (MongoDB, Express.js, React.js, Node.js).

Express.js memiliki beberapa keunggulan yang membuatnya populer kalangan *developer* web, seperti (a) mendukung pembuatan *middleware* yang berperan menyediakan mekanisme penyaringan HTTP request yang masuk ke aplikasi dengan kata lain setiap kali ada request masuk maka akan difilter oleh *Middleware*; (b) mendukung berbagai HTTP verb seperti POST, GET, PUT, DELETE, OPTION, HEAD, dan lainnya; (c) *express.js* memberikan layanan untuk dapat memilih *template engine* yang

diperlukan; (d) *express.js* dapat manajemen file statik seperti CSS dan Javascript; (e) *express.js* mempunyai sifat fleksibel sehingga sangat bebas untuk dikostumisasi. Contoh potongan program dari *express.js* ditunjukkan pada Gambar 2.2.

```
const express = require('express') // 4.16.4
const app = express()
const port = 3000

app.get('/', (req, res) => res.send('Hello World!'))

app.listen(port, () => console.log(`Example app listening on port ${port}!`))
```

Gambar 2.2 Potongan program *express.js*

## 2.8. JSON

*JavaScript Object Notation* (JSON) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dan lain-lain. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data ([json.org](http://json.org)).

## 2.9. Pengembangan Web Service dengan Metode Spiral

Dalam mengembangkan perangkat lunak terdapat beberapa metodologi yang dapat digunakan, salah satunya yaitu metode Spiral yang akan digunakan dalam mengembangkan *web service* pada penelitian ini. Metode Spiral ini dikemukakan pertama oleh Barry Boehm pada tahun 1988 dengan jurnalnya yang berjudul *A Spiral Model of Software Development and Enhancement*. Metode Spiral adalah metode yang menggabungkan antara sifat *iterative* metode *Prototyping* dan sifat linear metode *Waterfall* dengan penekanan yang tinggi pada analisis risiko tiap tahapannya. Dalam tiap iterasinya, proses pengembangan perangkat lunak mengikuti tahap-tahap fase linear di mana perangkat lunak akan dibuat *prototype (incomplete model)* dan pada akhir tiap fasenya, pengembang akan melakukan evaluasi bersama dengan *user*.

## 3. Metodologi

Penelitian ini menggunakan satu buah *personal computer* dan perangkat lunak untuk pengembangan *web service*, dengan spesifikasi perangkat keras dan perangkat lunak yang dapat dilihat pada Tabel 3.1 dan Tabel 3.2. Penelitian ini dilakukan dengan menggunakan hasil wawancara pimpinan dan karyawan UPT TIK UNJ. Sehingga didapat dokumentasi berupa daftar *requirement*.

Available at:

<http://journal.unj.ac.id/unj/index.php/pinter/article/view/23587>

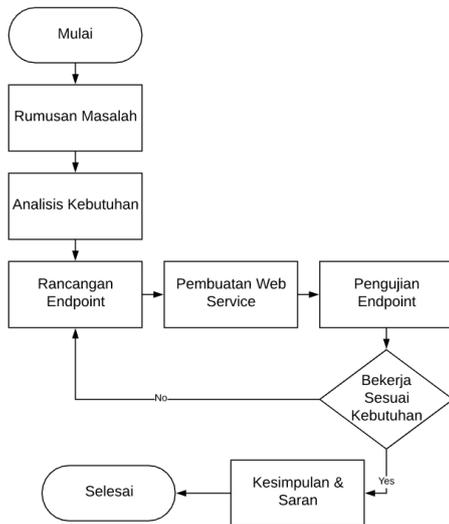
**Tabel 3.1 Daftar Perangkat Keras**

Perangkat Keras	Spesifikasi
Prosesor	Intel Core i5-7500 berkecepatan 3.4 GHz.
RAM	DDR4 8GB
Harddisk	1 TB
Display	NVIDIA GeForce GTX 1050 Ti

**Tabel 3.2 Daftar Perangkat Lunak**

Perangkat Lunak	Fungsi
Windows 10 64-bit	Sistem Operasi
Microsoft Office 365	Pengolah Kata
Visual Studio Code Version: 1.33.1	Text Editor
Node.js Version: 10.2.0	Node.js sebagai web server
Postman Version: 6.7.3	Aplikasi Postman Native dari Google sebagai media testing

Penelitian ini dilakukan dengan menggunakan metode Spiral yang merupakan gabungan antara sifat *iterative* metode *Prototyping* dan sifat linear metode Waterfall dengan penekanan yang tinggi pada analisis risiko tiap tahapannya. Secara garis besar, metode penelitian yang akan dilaksanakan seperti diagram alir di bawah ini:



**Gambar 2. Alur Penelitian**

Proses pertama dalam penelitian ini adalah merumuskan masalah yang dipaparkan pada bahasan sebelumnya. Setelah itu, melakukan analisis kebutuhan berdasarkan dokumentasi yang telah dibuat oleh tim UPT TIK, Hafidz Arief Raharjo untuk melihat kebutuhan data yang ada dan mengelompokkan sesuai kebutuhan yang membutuhkan *endpoint* dan tidak membutuhkan *endpoint*.

Setelah dilakukan analisis kebutuhan, selanjutnya adalah melakukan perancangan *web service* sesuai dengan kebutuhan dari sistem yang dibuat menjadi *endpoint-endpoint*. Setelah merancang, selanjutnya membangun *web service* sesuai dengan hasil rancangannya dan lakukan pengujian menggunakan Postman untuk *endpoint-endpoint* yang telah dibuat. Jika masih memiliki kesalahan, maka dilihat kembali rancangannya dan hasil *script* koding untuk ditelusuri kesalahannya, karena kesalahan bisa terjadi pada rancangan yang tidak sesuai atau logika yang kurang tepat. Setelah semua selesai dan *web service* dinyatakan berhasil dan layak, selanjutnya dilakukan penarikan kesimpulan dan saran.

Pengujian menggunakan metode unit testing yaitu pengujian yang dilakukan pada masing-masing *endpoint* yang dihasilkan pada saat proses pengembangan *web service*. Pengujian ini dilakukan untuk mengetahui apakah *web service* berfungsi dengan baik sesuai dengan *requirement* yang telah ditentukan secara spesifik satu persatu fungsi dari *endpoint-endpoint* yang telah dibuat.

**Tabel 3.1 Daftar Endpoint dengan Menggunakan URL**

<http://localhost/mahasiswa/siakad/public/api/>

No	Requirement	Endpoint	Type Request
1	Login	/user/login	POST
2	Menampilkan pertanyaan keamanan ( <i>captcha</i> )	/get-captcha	GET
3	Menampilkan daftar nilai beserta bobot nilai	/ref-bobot/	GET
4	Menampilkan daftar provinsi	/ref-provinsi/	GET
5	Menampilkan daftar kota/kabupaten	/ref-kota/	GET
6	Menampilkan daftar kecamatan	/ref-kecamatan/	GET
7	Menampilkan daftar pekerjaan	/ref-pekerjaan/	GET
8	Menampilkan daftar penghasilan	/ref-penghasilan/	GET
9	Menampilkan daftar pendidikan	/ref-pendidikan	GET

Available at:

<http://journal.unj.ac.id/unj/index.php/pinter/article/view/23587>



4.	Tidak	Status: 500.	/user/login	Internal Server Error
Waktu proses: 80 ms				

### 4.3 Pembahasan

Pengujian dilakukan oleh penguji ahli, yaitu seorang *programmer backend* yang juga sebagai *staff* di Unit Pelayanan Teknis Teknologi Informasi dan Komunikasi (UPT TIK) bernama Muhammad Nurmansyah Santosa, S.Pd. Berdasarkan hasil pengujian pada Tabel 4.1 hingga Tabel 4.25 yang dievaluasi oleh penguji ahli, dapat dilihat bahwa hasil pengujian berjalan dengan lancar tanpa adanya kesalahan pada sisi fungsional sistem. Dengan spesifikasi *personal computer* yang berbeda dan dengan banyaknya aplikasi lain yang dibuka di saat pengujian, disimpulkan bahwa *personal computer* penelitian dengan aplikasi yang beroperasi hanya yang berkaitan dengan pengujian memiliki waktu respons lebih cepat dibandingkan dengan *personal computer* penguji ahli saat pengujian dengan aplikasi lain yang beroperasi selain aplikasi penelitian sebanyak 4 aplikasi. Aplikasi tersebut di antaranya Google Chrome, I Series Navigator, AnyDesk, MEmu android emulator.

Penguji ahli memberikan masukan akan urutan data keluaran yang dihasilkan agar disusun lebih teratur, agar memudahkan *programmer frontend* dalam mengembangkan aplikasi menjadi aplikasi yang utuh. Sehingga dapat diimplementasikan dalam pengembangan aplikasi SIAKAD yang siap digunakan oleh *end user* nantinya.

### 4.4 Aplikasi Hasil Penelitian

Hasil dari penelitian ini menghasilkan beberapa *endpoint* berdasarkan *requirement* milik UPT TIK dan penambahan *requirement* berdasarkan pada kebutuhan sistem yang disetujui oleh UPT TIK. Hasil dapat digunakan untuk *web service* yang akan digabungkan dengan sisi *frontend* untuk menghasilkan sebuah web yang utuh. Sehingga dapat memberikan acuan untuk membangun aplikasi Sistem Akademik Universitas Negeri Jakarta di masa mendatang.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

*Web service* dikembangkan dengan menggunakan *framework* Express.js dan daftar *requirement* milik UPT TIK sebagai acuan. Daftar *requirement* digunakan untuk memetakan *endpoint*, satu buah *requirement* dapat menghasilkan satu atau lebih *endpoint* dan juga beberapa *requirement* dapat dibuat menjadi satu *endpoint*.

Dalam pengembangan *web service* modul mahasiswa, dihasilkan 25 buah *endpoint* yang dihasilkan berdasarkan pada *requirement*. Kemudian masing-masing *endpoint* diuji dengan metode unit

testing untuk mengetahui apakah *endpoint* bekerja dengan baik dan memberikan *output* yang sesuai dengan *requirement*.

Hasil pengujian fungsional dengan metode unit testing menunjukkan bahwa *endpoint* yang telah dibuat berfungsi dengan baik sesuai dengan skenario proses dan layak digunakan sebagai *web service* sebuah aplikasi.

### 5.2 Saran

Untuk pengembangan penelitian lebih lanjut disarankan untuk melakukan beberapa hal berikut:

1. Membandingkan teknologi baru yang sesuai dengan perkembangan teknologi *web service*.
2. Penyesuaian terhadap *requirement* yang mungkin saja mengalami perubahan setiap waktunya.
3. Hasil *web service* dapat dikembangkan dengan menerapkan prinsip *frontend-backend*

### Daftar Pustaka:

- Auth0. Introduction to JSON Web Tokens. Diambil dari <https://jwt.io/introduction/>. Diakses pada tanggal 28 April 2019 Pukul 16.15 WIB.
- Boehm, Barry W. (1988). A Spiral Model of Software Development and Enhancement. *Computer*, 21(5):61-72.
- Djojodihardjo, Harijono (1984). Pengantar Sistem Komputer. Jakarta: Erlangga.
- Erl, Thomas., dkk. (2008). Web Service Contract Design and Versioning for SOA. New Jersey: Prentice Hall.
- Hidayat, Rahmat. (2010). Cara Praktis Membangun Website Gratis : Pengertian Website. Jakarta : PT Elex Media Komputindo Kompas, Granedia.
- Imelda, Erik, M (2014). Perancangan Sistem Informasi Akademik Pada Sekolah Dasar Negeri Sukajadi 9 Bandung. *Jurnal Sistem Informasi Akademik*, 3:47-48.
- Jogianto, H.M (2005). Sistem Teknologi Informasi. Yogyakarta: Andi.
- Kapojos, F., Wowor, H.F., Rumagit, A.M., Wowor, A.P.R. (2012). Implementasi Service-Oriented Architecture dengan Web Service untuk Aplikasi Informasi Akademik. *Jurnal Teknik Elektro dan Komputer*, 1.
- Kertahadi (2007). Sistem Informasi Manajemen. Jakarta: Pustaka Binanam Pressindo.
- O'Brein, James A. (2005). Pengantar Sistem Informasi. Jakarta: Salemba 4.
- Oliveira, Anderson S; Serme, Gabriel; Massiera, & Roudier, Yves. (2012). Enabling Message Security for RESTful Services. *Web Services (ICWS), 2012 IEEE 19th International Conference,*

Available at:

<http://journal.unj.ac.id/unj/index.php/pinter/article/view/23587>

- Rusdiana, H.A; & Irfan, Moch. (2014). Sistem Informasi Manajemen. Bandung : Pustaka Setia.
- Satoto, Kodrat Iman. 2009. Analisis Keamanan Sistem Informasi Akademik Berbasis Web di Fakultas Teknik Universitas Diponegoro. Di dalam: Seminar Nasional Aplikasi Sains dan Teknologi. Yogyakarta, 13 Des 2009.
- Snell, James; Tidwel, Doug; & Kuchenko, Pavel. (2001). Programming web service with SOAP. California: O'Reilly Media.
- Tutorialpoints. What are web wervices?. Diambil dari [https://www.tutorialspoint.com/webservicees/what\\_are\\_web\\_services.htm](https://www.tutorialspoint.com/webservicees/what_are_web_services.htm). Diakses pada tanggal 25 April 2019 Pukul 23.15 WIB.
- Yuhefizar; Mooduto; & Hidayat, R. (2009). Cara Mudah Membangun Website Interaktif Menggunakan Content Management System Joomla Edisi Revisi. Jakarta: PT Elex Media Komputindo.
- Wardana. (2016). Aplikasi Website Profesional dengan PHP dan jQuery. Jakarta: Elex Media Komputindo.

Available at:

<http://journal.unj.ac.id/unj/index.php/pinter/article/view/23587>