

## PENGEMBANGAN *WEB SERVICE* MODUL *ADMIN* PADA SISTEM INFORMASI PENERIMAAN CALON ANGGOTA BARU KSR PMI UNIVERSITAS NEGERI JAKARTA

Mohammad Saeful Amin<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>msaefulamin13@gmail.com, <sup>2</sup>hamidillah@unj.ac.id, <sup>3</sup>widodo@unj.ac.id

---

### Abstrak

Tantangan terbesar yang dihadapi PMI adalah memastikan bahwa proses rekrutmen relawan dapat terus berlanjut. KSR PMI Unit UNJ merupakan sarana bagi mahasiswa untuk mengamalkan tugas kepalangmerahan dan tugas-tugas organisasi sebagai wujud tanggung jawab kepada masyarakat dan negara. Dalam penyiapan kader atau sumber daya manusia, KSR PMI Unit UNJ akan melaksanakan rangkaian kegiatan penerimaan calon anggota baru (PCAB) sesuai dengan aturan standar pelatihan PMI untuk menghasilkan sumber daya yang berkualitas. Tujuan dari penelitian ini adalah untuk menghasilkan web service yang dapat digunakan dalam Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit UNJ dan memudahkan dalam melaksanakan rangkaian kegiatan penerimaan calon anggota baru KSR PMI Unit UNJ serta memiliki sifat adaptif terhadap perubahan-perubahan di masa mendatang. Pengembangan web service modul admin Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit UNJ dengan metode scrum menggunakan framework Express.js dan menghasilkan 58 endpoint. Pengujian dengan metode unit testing terhadap 58 endpoint menggunakan aplikasi bernama Postman. Hasil pengujian menunjukkan bahwa 58 endpoint yang diuji berfungsi dengan baik, dan layak digunakan sebagai sisi backend yang akan digabungkan dengan sisi frontend untuk menjadi aplikasi Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit UNJ yang utuh.

**Kata kunci :** Sistem Informasi, Web Service, Metode Scrum, Express, Endpoint, Unit Testing.

---

## 1. Pendahuluan

### 1.1. Latar Belakang

Tantangan terbesar bagi PMI adalah untuk memastikan terjadinya proses rekrutmen relawan, sesuai dengan buku Pedoman Manajemen Relawan (Palang Merah Indonesia, 2008). Relawan dalam lingkungan organisasi PMI adalah seseorang yang melaksanakan kegiatan kepalangmerahan baik secara tetap maupun tidak tetap sesuai dengan prinsip-prinsip dasar Gerakan Palang Merah dan Bulan Sabit Merah Internasional serta diorganisasikan oleh Palang Merah Indonesia (Palang Merah Indonesia, 2008). Menurut kantor berita BBC, Indonesia menjadi negara yang paling rawan terhadap bencana di dunia berdasar data yang dikeluarkan oleh Badan Perserikatan Bangsa-Bangsa untuk Strategi Internasional Pengurangan Risiko Bencana (UNISDR) (BBC Indonesia, 2011). Berdasarkan fakta tersebut maka sudah sepantasnya PMI membutuhkan lebih banyak relawan untuk bergabung dan menjadi bagian dari antisipasi dalam pencegahan dan penanggulangan bencana di Indonesia.

Berdasarkan data laporan tahun 2019/2020 pada (Kementerian Riset, Teknologi dan Pendidikan Tinggi, 2020) Universitas Negeri Jakarta memiliki mahasiswa aktif sebanyak 33.155. Dari banyaknya jumlah mahasiswa tersebut, sebagian mahasiswa Universitas Negeri Jakarta berpotensi menjadi relawan PMI. Mengingat mahasiswa masuk ke dalam salah satu syarat untuk menjadi relawan PMI yaitu berada pada rentang usia dari 18 tahun hingga 35 tahun.

Universitas Negeri Jakarta memiliki sejumlah organisasi kemahasiswaan yang disebut Unit Kegiatan Mahasiswa (UKM). Pada saat ini UKM Universitas Negeri Jakarta memiliki 20 Organisasi yang aktif, bagi para mahasiswa yang ingin menjadi relawan PMI dapat menyalurkan minatnya pada UKM Korps Sukarela Palang Merah Indonesia Unit Universitas Negeri Jakarta sebagai organisasi yang mewadahi mahasiswa untuk mengamalkan Tri Dharma Perguruan Tinggi yaitu pengabdian kepada masyarakat.

KSR PMI Unit UNJ berfungsi sebagai salah satu wahana untuk merencanakan, melaksanakan, dan mengembangkan kegiatan kepalangmerahan, keorganisasian, dan pengabdian kepada masyarakat serta sivitas akademika Universitas Negeri Jakarta. KSR PMI Unit UNJ berfungsi sebagai salah satu wahana bagi mahasiswa untuk mendarmabaktikan diri bagi tugas-tugas kepalangmerahan dan keorganisasian sebagai bentuk tanggung

jawab kepada masyarakat dan bangsa. Dalam menyiapkan kader atau sumber daya manusia KSR PMI Unit UNJ melakukan Penerimaan Calon Anggota Baru (PCAB) dalam rangkaian kegiatan untuk menghasilkan sumber daya yang berkualitas sesuai dengan aturan standar pelatihan PMI.

Rangkaian kegiatan untuk Penerimaan Calon Anggota Baru (PCAB) KSR PMI Unit UNJ terdiri dari (1) Calon anggota melakukan pendaftaran melalui kertas yang berisi biodata, surat persetujuan orang tua, dan berkomitmen dalam mengikuti rangkaian kegiatan PCAB, (2) Calon anggota mengikuti delapan pertemuan pendidikan dan pelatihan (diklat) yang di dalamnya terdapat materi dan latihan soal, latihan soal sendiri terdiri dari tiga bagian yaitu *pretest*, *weekly test* dan *posttest*, (3) Calon anggota melaksanakan pengaplikasian materi yang sudah dipelajari dalam diklat pada area kampus, (4) Calon anggota melaksanakan kegiatan pengukuhan di luar kampus untuk mengaplikasikan materi yang sudah dipelajari dalam pelatihan yang sudah dilewati.

Dalam melaksanakan rangkaian kegiatan PCAB KSR PMI Unit UNJ seringkali mengalami sejumlah kendala. Berdasarkan wawancara yang dilakukan dengan pengurus KSR PMI Unit UNJ ada beberapa point permasalahan yang dihadapi yaitu (1) Pendaftaran calon anggota baru masih menggunakan kertas sehingga seringkali berkas pendaftaran calon anggota baru sulit ditemukan, (2) Data calon anggota baru tidak tersimpan dengan baik karena tersebar di beberapa pengurus KSR PMI Unit UNJ, (3) Dalam melaksanakan diklat para calon anggota baru mengerjakan soal latihan masih menggunakan kertas, sehingga sering kali hasil soal latihan hilang dan terkadang jawaban tidak dapat terbaca dengan jelas, (4) Pengolahan dan penyimpanan nilai di setiap rangkaian pelatihan belum tersimpan dengan baik, (5) Catatan pembayaran dan keuangan dilakukan secara manual sehingga sering terjadi perbedaan data pembayaran diantara panitia.

Sistem Informasi PCAB akan dibangun dengan menggunakan teknologi *web*. Pemanfaatan *web* dibutuhkan dalam mengelola dokumen dan nilai calon anggota baru, mengembangkan fitur-fitur *web* sistem informasi meliputi pendaftaran calon anggota baru, validasi calon anggota baru, *monitoring* seluruh pembayaran, mengelola hasil nilai *test* dan dokumen penting lainnya. Untuk menghasilkan sebuah sistem yang bagus dan sesuai kebutuhan *user*, maka diperlukan analisis *requirement* yang nantinya akan menghasilkan fitur-fitur yang ada pada *web* tersebut.

*Web* sebagai sebuah perangkat lunak memiliki siklus hidup yang dinamis. Selain penambahan fitur baru, perubahan *web* termasuk di antaranya adalah pemenuhan kebutuhan cara pengguna berinteraksi dalam bentuk perbaikan tampilan *web*. Pengembangan suatu *web* dapat dipisahkan menjadi dua buah komponen, yaitu komponen *frontend* dan *backend*. Komponen *frontend* adalah bagian *web* yang langsung berinteraksi dengan pengguna (*user interface*). Sedangkan komponen *backend* adalah bagian *web* yang melakukan pengolahan data dan logika bisnis. Pada *web* yang tidak memisahkan *frontend* dan *backend*, maka dalam pengembangannya akan membutuhkan usaha yang lebih besar pada proses selanjutnya, dikarenakan harus melakukan pengembangan secara keseluruhan. Hal ini berbeda jika pengembangan *web* memisahkan kedua komponen tersebut. Pengembangan pada sisi *frontend* dapat dilakukan walaupun tidak ada pengembangan pada sisi *backend*, begitu pula sebaliknya.

*Frontend* merupakan unsur pada sisi *client* yang akan menampilkan sisi atau *view* yang akan berinteraksi langsung dengan pengguna, sedangkan *backend* atau sering disebut *service*. *Web service* merupakan server *side* yang mengelola data yang berasal dari *database* dan menjalankan *business logic* dari sebuah *web*. Sementara itu *database* adalah sekumpulan data yang berhubungan secara logika dan deskripsi mengenai data itu sendiri yang dirancang untuk memenuhi kebutuhan informasi organisasi. Pembagian ini dimaksudkan untuk memudahkan aplikasi *web* beradaptasi dengan pembaruan yang akan terjadi sewaktu-waktu, sehingga ketika ada pembaruan pada sisi tampilan cukup melakukan perubahan di sisi *frontend* tanpa harus mengubah sisi *backend*, demikian juga sebaliknya. Salah satu bagian dari komponen *backend* adalah *web service*. *Web service* merupakan salah satu dari implementasi arsitektur berbasis layanan atau yang biasa dikenal sebagai *Service-Oriented Architecture (SOA)*. *Web service* adalah sistem yang dibuat untuk menghubungkan antara bagian *frontend* dengan *database*.

Pengembangan sistem informasi ini dilakukan secara kolektif oleh tim yang terdiri dari pengembang *backend*, *frontend*, dan UI/UX. Penelitian ini berfokus pada pengembangan *backend* bagian admin berbasis REST API menggunakan bahasa pemrograman Javascript dengan *framework* ExpressJs. *Backend* digunakan untuk menyimpan, mengolah, dan mengintegrasikan data pada keseluruhan sistem informasi.

## 1.2. Identifikasi Masalah

Berdasarkan latar belakang yang telah dijelaskan, dapat diidentifikasi beberapa masalah yang muncul, antara lain :

1. Data calon anggota baru tidak tersimpan dengan baik dan tidak terpusat di satu tempat karena tersebar di beberapa anggota KSR PMI Unit UNJ.
2. Berkas pendaftaran calon anggota baru sering kali sulit ditemukan.
3. Pengerjaan soal latihan masih konvensional menggunakan kertas, sehingga panitia cukup kerepotan dalam mengoreksi hasil latihan.
4. Pengolahan dan penyimpanan nilai di setiap rangkaian masih belum baik.
5. Pengolahan sistem pembayaran dan keuangan di setiap rangkaian masih belum baik.

6. Diperlukan sebuah sistem informasi sebagai wadah untuk mengatur dan mengolah data dalam rangkaian penerimaan calon anggota baru KSR PMI Unit UNJ.

### 1.3. Pembatasan Masalah

Melihat luasnya lingkup permasalahan yang teridentifikasi pada pembahasan sebelumnya, pembatasan masalah sangat penting untuk dilakukan.

Penelitian dibatasi pada :

1. Komponen-komponen yang disediakan khusus pada sisi admin sistem informasi *web* PCAB KSR PMI Unit UNJ.
2. Fokus penelitian ini menghasilkan *web service* berupa *endpoint-endpoint* dan tidak menghasilkan sistem informasi secara utuh.
3. Fokus penelitian ini hanya pada pengembangan *web service*, dalam rangka mengembangkan sistem informasi yang siap digunakan.

### 1.4. Rumusan Masalah

Berdasarkan latar belakang, identifikasi, dan pembatasan masalah, maka perumusan masalah yang akan dibahas pada tugas akhir ini adalah: Bagaimana mengembangkan *web service* modul admin untuk Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit UNJ.

### 1.5. Tujuan Penelitian

Berlandaskan dengan rumusan masalah, hasil penelitian ini bertujuan untuk menghasilkan *web service* yang dapat digunakan dalam membuat sistem informasi berbasis *web* yang memenuhi kebutuhan dan kenyamanan bagi pengguna, dan juga diharapkan dapat memudahkan dalam melaksanakan rangkaian kegiatan penerimaan calon anggota baru KSR PMI Unit UNJ.

### 1.6. Manfaat Penelitian Tujuan Penelitian

Berdasarkan penjabaran di atas tentang penelitian yang akan dilakukan, manfaat yang akan didapatkan di antaranya adalah:

1. Hasil dari penelitian ini diharapkan dapat dimanfaatkan sebagai referensi dan masukan bagi perkembangan ilmu pengembangan aplikasi *web* khususnya *web programming*.
2. Hasil dari penelitian ini diharapkan dapat digunakan sebagai sebuah *backend* yang akan diimplementasikan dengan *frontend* sehingga menghasilkan sistem informasi PCAB KSR PMI Unit UNJ berbasis *web* secara utuh.

## 2. Dasar Teori

### 2.1. Palang Merah Indonesia

Palang Merah Indonesia (PMI) merupakan perhimpunan nasional yang berdiri atas asas perikemanusiaan dan atas dasar sukarela dengan tidak membeda-bedakan bangsa, golongan, dan paham politik (Dewan Perwakilan Rakyat, 2018). Berdirinya Palang Merah di Indonesia sebetulnya sudah dimulai sebelum Perang Dunia ke II, tepatnya 12 Oktober 1873. Pemerintah Kolonial Belanda mendirikan Palang Merah di Indonesia dengan nama *Nederlandsche Roode Kruis Afdeeling Indië* (NERKAI) yang kemudian dibubarkan pada saat pendudukan Jepang.

PMI adalah organisasi kemanusiaan yang berstatus badan hukum, diundangkan dengan Undang-undang Nomor 1 Tahun 2018 tentang Kepalangmerahan guna menjalankan kegiatan Kepalangmerahan sesuai dengan Konvensi Jenewa Tahun 1949, dengan tujuan untuk mencegah dan meringankan penderitaan dan melindungi korban tawanan perang dan bencana, tanpa membedakan agama, bangsa, suku bangsa, warna kulit, jenis kelamin, golongan, dan Pandangan Politik (Dewan Perwakilan Rakyat, 2018).

Adapun tugas yang dilakukan PMI adalah : Memberikan bantuan kepada korban konflik bersenjata, kerusuhan dan lainnya, memberikan pelayanan darah sesuai dengan ketentuan peraturan perundang-undangan, melakukan pembinaan relawan, melaksanakan pendidikan dan pelatihan yang berkaitan dengan Kepalangmerahan, menyebarkan informasi yang berkaitan dengan kegiatan Kepalangmerahan, membantu dalam penanganan musibah dan/atau bencana di dalam dan di luar negeri, membantu pemberian pelayanan kesehatan dan sosial, dan melaksanakan tugas kemanusiaan lainnya yang diberikan oleh pemerintah.

Dalam melaksanakan tugasnya PMI memiliki sumber daya manusia berupa relawan yang telah memenuhi persyaratan dan sudah menjalankan standar pelatihan PMI. Komponen relawan PMI yaitu (1) Palang Merah Remaja (PMR), (2) Korps Sukarela (KSR), (3) Tenaga Sukarela (TSR) dan (4) Donor Darah Sukarela (DDS) (Palang Merah Indonesia, 2014). Para relawan ini diwadahi pada suatu forum yang disebut Forum Relawan untuk KSR yang terdiri dari KSR PMI Unit Perguruan Tinggi, Markas dan TSR, sedangkan untuk tingkatan PMR disebut

Forum Remaja Palang Merah Indonesia (FORPIS) di setiap masing-masing wilayah baik di tingkat provinsi maupun ditingkat kabupaten/kota.

## 2.2. Korps Sukarela Universitas Negeri Jakarta

Korps Sukarela Palang Merah Indonesia Unit Universitas Negeri Jakarta (KSR PMI Unit UNJ) yang semula bernama KSR PMI Unit IKIP Jakarta didirikan pada tanggal 10 Mei 1994. Sehubungan dengan perubahan institusi yang bersangkutan dan hasil Musyawarah Istimewa KSR PMI Unit IKIP Jakarta, sejak tanggal 20 November 1999 berubah menjadi KSR PMI Unit Universitas Negeri Jakarta. Kemudian berdasarkan buku pedoman KSR PMI Perguruan Tinggi yang diterbitkan oleh kantor pusat PMI Tahun 2003, KSR PMI Unit UNJ berubah menjadi KSR PMI UNJ, berdasarkan buku Pedoman Manajemen Relawan Tahun 2008 dan Surat Kesepakatan Bersama PMI Pusat dengan Mendikbud No. 1/II/KB/2012, maka KSR PMI UNJ berubah menjadi KSR PMI Unit UNJ yang berbentuk Unit Kegiatan Mahasiswa pada tahun 2012.

## 2.3. Sistem Informasi

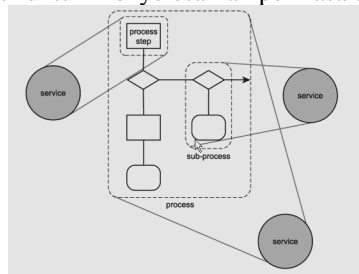
Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Sutabri, 2012).

## 2.4. 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*.

## 2.5. 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 1. Enkapsulasi business process dengan service (Kapojos, 2012)

## 2.6. Web Service Restful

*Web service* adalah perangkat lunak yang membuatnya tersedia melalui internet dan menggunakan sistem pesan XML standar atau JSON. XML atau JSON digunakan untuk menyandikan semua komunikasi ke *web service*. Misalnya, *client* meminta *web service* dengan mengirim pesan XML atau JSON, lalu menunggu respons XML atau JSON yang sesuai. Karena semua komunikasi menggunakan XML atau JSON, maka *web service* tidak terikat pada satu sistem operasi atau bahasa pemrograman apa pun (Tutorialspoint).

*Web service* memiliki dua bentuk, yaitu *Simple Object Access Protocol* (SOAP) dan *Representational State Transfer* (REST). SOAP dan REST memiliki kesamaan diantaranya, 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.

## 2.7. 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.

## 2.8. Express.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 dikalangan 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 memanajemen *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')
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. Potongan program express.js

## 2.9. 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).

## 2.10. Pengembangan Web Service dengan Metode Scrum

Dalam mengembangkan perangkat lunak terdapat beberapa metodologi yang dapat digunakan, salah satunya yaitu metode *scrum* yang akan digunakan dalam mengembangkan *web service* pada penelitian ini. Metode *scrum* merupakan salah satu dari metode *agile* dimana metode ini melihat teknik *requirement* sebagai proses yang berulang. *Scrum* merupakan sebuah pendekatan dari *agile* yang dilakukan untuk mengembangkan produk serta layanan agar lebih inovatif (Rubin, 2013). *Scrum* adalah metode pengembangan perangkat lunak modern yang lahir akibat sistem dan pola pengembangan perangkat lunak yang semakin maju dan berkembang.

## 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 1 dan Tabel 2. Penelitian ini dilakukan dengan menggunakan hasil wawancara anggota dan pengurus KSR PMI Unit UNJ. Sehingga didapat dokumentasi berupa daftar *requirement*.

Tabel 1. Daftar Perangkat Keras

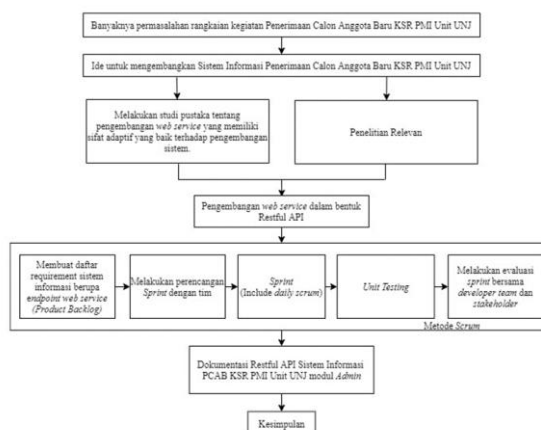
| Perangkat Keras | Spesifikasi                                   |
|-----------------|---|
| Prosesor        | Intel Core i5-8250U<br>berkecepatan 1.60 GHz. |
| RAM             | DDR4 8GB                                      |
| Harddisk        | 1,24 TB                                       |
| Display         | AMD Radeon™ 530                               |

Tabel 2. Daftar Perangkat Lunak

| Perangkat Lunak       | Fungsi         |
|-----------------------|----------------|
| Windows 10 Pro 64-bit | Sistem Operasi |
| Microsoft Office 365  | Pengolah Kata  |

|                                    |  |
|------------------------------------|--|
| Visual Studio Code Version: 1.52.1 | <i>Text Editor</i>   |
| Node.Js Version: 14.15.0           | Node.Js sebagai <i>web server</i>                                |
| Postman Version: 7.26.0            | Aplikasi Postman Native dari Google sebagai <i>media testing</i> |

Penelitian ini dilakukan dengan menggunakan metode *Scrum* merupakan salah satu dari metode *agile* dimana metode ini melihat teknik *requirement* sebagai proses yang berulang. Secara garis besar, metode penelitian yang akan dilaksanakan seperti diagram alir di bawah ini:



**Gambar 3. Alur Penelitian**

Pengembangan *web service* akan menggunakan metode *Scrum* dengan empat tahapan utama. Pertama adalah membuat daftar requirement sistem informasi (*Product Backlog*). Kedua, melakukan perencanaan *Sprint* dengan tim. Ketiga, melakukan *Sprint* dengan *daily scrum*. Keempat, tahapan evaluasi dengan melakukan testing menggunakan metode *Unit Testing*, apakah *web service* yang dihasilkan sudah sesuai dengan kebutuhan, mempunyai logika yang benar, efektif dan efisien sehingga siap digunakan. Kelima, melakukan evaluasi *sprint* bersama *developer team* dan *stakeholder* dengan tujuan untuk mengembangkan Sistem Informasi PCAB KSR PMI Unit UNJ secara utuh, mampu beradaptasi atas pembaruan dan pengembangan sistem di masa yang akan datang.

**Tabel 3. Daftar Endpoint Modul Admin**

| No | Requirement  | Endpoint                       | Type Request |
|----|--|--------------------------------|--------------|
| 1  | Login  | /users/login                   | POST         |
| 2  | Menampilkan pertanyaan keamanan ( <i>captcha</i> )         | /getCaptcha                    | GET          |
| 3  | Menampilkan jumlah pendaftar berdasarkan tahap pendaftaran | admin/tahap/:tahap             | GET          |
| 4  | Mengubah tahap pendaftaran                                 | admin/tahap/edit/:id_user      | POST         |
| 5  | Menampilkan daftar dokumen pendaftaran                     | validasi/dokumen/all           | GET          |
| 6  | Mengubah status validasi dokumen pendaftaran               | validasi/dokumen/edit/:id_user | POST         |
| 7  | Menampilkan daftar transaksi pembayaran                    | validasi/transaksi/all         | GET          |
| 8  | Menampilkan detail daftar transaksi pembayaran             | validasi/transaksi/:id_user    | GET          |
| 9  | Mengubah validasi transaksi pembayaran                     | validasi/transaksi/edit/:id    | POST         |
| 10 | Menampilkan biodata semua peserta                          | biodata/all                    | GET          |

Pengujian (*endpoint*) fungsional dilakukan dengan proses skenario yang telah ditentukan seperti pada Tabel 3 – Tabel 5 menggunakan *tool* Postman.

**Tabel 4. Pengujian Fungsional pada Endpoint Login**

| No | Endpoint     | Skenario | Respon   |
|----|--------------|----------|--|
| 1  | /users/login | Sukses   | Berhasil login dan menampilkan data token, dan mengirimkan HTTP status code 200 (OK) |

| No | Endpoint | Skenario | Respon   |
|----|----------|----------|--|
|    |          | Gagal    | Mengirimkan HTTP status code 404 (Not Found)   |
|    |          |          | Status respon 500 <i>Internal server error</i> |

**Tabel 5. Pengujian Fungsional pada Endpoint Menampilkan Pertanyaan Keamanan (Captcha)**

| No | Endpoint    | Skenario | Respon   |
|----|-------------|----------|--|
| 2  | /getCaptcha | Sukses   | Menampilkan pertanyaan keamanan (captcha), dan mengirimkan HTTP status code 200 (OK) |
|    |             | Gagal    | Mengirimkan <i>HTTP status code 500 (Internal Server Error)</i>                      |

#### 4. Hasil dan Analisis

##### 4.1 Deskripsi Hasil Penelitian

Produk yang dikembangkan menghasilkan 58 buah *endpoint*. Setiap *endpoint* dikembangkan dalam beberapa proses, yaitu: 1) penentuan data yang akan digunakan; 2) penentuan algoritma; 3) penulisan kode program; dan 4) pengujian terhadap *endpoint*. Pengembangan *endpoint* dilakukan menggunakan metode *scrum* hingga berhasil dikembangkan dengan baik tanpa kesalahan sesuai dengan target pembuatan produk. Dan tidak lupa memperhatikan pesan yang akan dikirimkan agar memudahkan pihak pengembang *frontend* dalam menerima *endpoint* yang dikirimkan.

Metode *Scrum* memiliki beberapa tahapan dalam mengembangkan suatu produk perangkat lunak. Tahapan *Scrum* dimulai dari tahap pengumpulan data yang dijadikan sebagai data primer untuk membangun *requirement* dan daftar data primer ini disebut dengan *product backlog*, dilanjutkan dengan tahap *sprint planning* yang merupakan tahapan perencanaan sebelum melakukan *sprint*, kemudian dilanjutkan dengan menyusun *sprint backlog* sebagai daftar kebutuhan yang harus diselesaikan dalam *sprint*, selanjutnya *sprint* dilakukan sebagai tahap kerja implementasi, dan *sprint review* yang merupakan tahap pengujian hasil *sprint* serta *sprint retrospective* yang merupakan perencanaan dalam mengembangkan tingkat produk selanjutnya.

##### 4.2 Analisis Data Penelitian

###### 4.2.1 Tahap Product Backlog

Pengembangan Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit UNJ dibuat berdasarkan *product backlog web service* bagian *admin* yang telah dibuat dan dibahas oleh *developer team* bersama dengan *product owner*. Tabel 4.1 menunjukkan daftar *product backlog* yang ada pada kebutuhan fungsional bagian *admin*. Penentuan prioritas pengerjaan *product backlog* didapatkan dari hasil komunikasi antara *developer team* dan *product owner*. Pada *product backlog* terdapat beberapa fungsi yang akan dikembangkan dari hasil analisis yang dilakukan.

**Tabel 6. Product Backlog**

| No | Daftar Kebutuhan Perangkat Lunak               | Waktu  | Prioritas | Tingkat Kesulitan | Sprint ke- |
|----|--|--------|-----------|-------------------|------------|
| 1  | Database sistem                                | 3 hari | Tinggi    | Tinggi            | 1          |
| 2  | Pengembangan fungsi <i>login</i>               | 1 hari | Tinggi    | Tinggi            | 1          |
| 3  | Membuat mekanisme pertanyaan <i>captcha</i>    | 1 hari | Sedang    | Sedang            | 1          |
| 4  | Pengembangan fungsi biodata peserta            | 1 hari | Sedang    | Sedang            | 2          |
| 5  | Membuat mekanisme pembayaran                   | 2 hari | Tinggi    | Tinggi            | 2          |
| 6  | Membuat mekanisme validasi akun dan pembayaran | 1 hari | Tinggi    | Tinggi            | 2          |
| 7  | Pengembangan fungsi pengumuman                 | 1 hari | Sedang    | Sedang            | 3          |
| 8  | Pengembangan fungsi peraturan                  | 1 hari | Sedang    | Sedang            | 3          |
| 9  | Pengembangan fungsi materi diklat              | 2 hari | Tinggi    | Tinggi            | 3          |
| 10 | Pengembangan fungsi kehadiran                  | 3 hari | Tinggi    | Tinggi            | 4          |
| 11 | Pengembangan fungsi soal                       | 3 hari | Tinggi    | Tinggi            | 4          |
| 12 | Pengembangan fungsi penilaian                  | 4 hari | Tinggi    | Tinggi            | 4          |



#### 4.2.2 Tahap *Sprint Planning*

Pada tahap ini, *product backlog* yang telah ada, dibagi menjadi beberapa *sprint backlog* dan tugas untuk setiap judul *product backlog* akan lebih dirinci untuk tahapan selanjutnya, dimana setiap *sprint backlog* akan diimplementasikan pada setiap *sprint*. Daftar *sprint backlog* yang dirancang dapat dilihat pada tabel 7.

Tabel 7. *Sprint backlog backend untuk sprint 1*

| No | Deskripsi                            | Prioritas |
|----|--------------------------------------|-----------|
| 1  | Pembuatan database sistem            | Tinggi    |
| 2  | Pengembangan fungsi <i>login</i>     | Tinggi    |
| 3  | Membuat mekanisme pertanyaan captcha | Sedang    |

#### 4.2.3 Tahap *Sprint*

Tabel 8. *Daily Scrum 1*

| Hari/Tanggal           | Evaluasi Pekerjaan  | Perencanaan Kerja  |
|------------------------|---|--|
| Senin, 5 Oktober 2020  | Mengkonfirmasi tabel users beserta <i>primary key</i> yang digunakan.   | Membenahi perancangan <i>database</i> terhadap tabel yang akan mengolah tentang skema <i>login</i> pada sistem                             |
| Selasa, 6 Oktober 2020 | Memperbaiki perkiraan penggunaan NIM diubah menjadi <i>email</i> untuk selanjutnya dijadikan <i>username</i> untuk <i>login</i> kedalam sistem. | Mengubah perkiraan <i>username</i> untuk <i>user</i> calon anggota baru dari NIM menjadi menggunakan <i>email</i> dari calon anggota baru. |
| Rabu, 7 Oktober 2020   | Memperbaiki perkiraan penggunaan <i>primary key</i> terhadap semua tabel yang akan digunakan pada sistem.                                       | Mengubah perancangan <i>primary key</i> untuk semua tabel yang akan digunakan.   |
| Kamis, 8 Oktober 2020  | Memperbaiki validasi yang digunakan yaitu <i>JWT validation</i> untuk mengetahui <i>user</i> terautentikasi pada sistem.                        | Melakukan pengodean yang pertama dilakukan yaitu membuat fungsi <i>login</i> pada sistem.  |
| Jumat, 9 Oktober 2020  | Memperbaiki pengodean mekanisme pengambilan database pada fungsi <i>captcha</i> .   | Melakukan pengodean untuk mekanisme pertanyaan <i>captcha</i> .  |

Tabel 9. Hasil *Sprint 1*

| <i>Sprint 1</i>             |                           |                               |                              |
|-----------------------------|---------------------------|-------------------------------|------------------------------|
| 1                           | Pembuatan database sistem |                               |                              |
| Perancangan database sistem |                           |                               |                              |
| Nama database: db_pcab      |                           |                               |                              |
| Tabel: tb_users             |                           |                               |                              |
| No                          | Nama Kolom                | Tipe                          | Keterangan                   |
| 1                           | Id                        | int(11)                       | Not null (PK) Auto Increment |
| 2                           | email                     | varchar(40)                   | Not null                     |
| 3                           | password                  | varchar(255)                  | Not null                     |
| 4                           | role                      | enum('admin', 'user')         | Not null                     |
| 5                           | tahap                     | enum('0', '1', '2', '3', '4') | Not null                     |
| 6                           | isVerified                | boolean(1)                    | Not null                     |
| 7                           | created_at                | timestamp                     | Not null                     |
| 8                           | update_at                 | timestamp                     | Not null                     |



#### 4.2.4 Tahap Unit Testing

Tabel 10. Pengujian pada *endpoint* dengan tipe *request* GET untuk menampilkan pertanyaan keamanan (*captcha*)

| No                  | Status Bekerja (Ya/Tidak) | Skenario (1 & 2) | URL Skenario | Output  |
|---------------------|---------------------------|------------------|--------------|---|
| 1                   | Ya                        | Status: 200      | getCaptcha   | <pre>{   "meta": {     "code": 200,     "status": "success",     "message": "berhasil mendapatkan captcha"   },   "data": {     "soal": "1+2 = ",     "jawaban": "3",     "kode": 0   } }</pre> |
| 2                   | Tidak                     | Status: 500      | getCaptcha   | <pre>{   "meta": {     "code": 500,     "status": "error",     "message": "gagal mendapatkan captcha"   },   "data": null }</pre>   |
| Waktu proses: 17 ms |                           |                  |              |   |

### 4.3 Pembahasan

Pengujian dilakukan dengan penguji ahli yang bekerja sebagai salah satu staf *programmer* di UPT TIK Universitas Negeri Jakarta, Febrianto Widyoutomo, S.Pd. Berdasarkan hasil pengujian pada Tabel 4.15 sampai Tabel 4.72 yang sudah dievaluasi oleh penguji, dapat disimpulkan bahwa hasil pengujian menunjukkan tidak adanya *error* atau kesalahan pada *web service* yang dibuat. Dengan spesifikasi *personal computer* yang berbeda dan dengan banyaknya aplikasi lain yang dibuka disaat 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 3 aplikasi. Aplikasi tersebut di antaranya Google Chrome, AnyDesk, dan Discord.

Penguji ahli memberikan masukan untuk memberikan pesan error dalam bentuk json, agar memudahkan *programmer frontend* dalam mengembangkan aplikasi menjadi aplikasi yang utuh. Sehingga dapat diimplementasikan dalam pengembangan aplikasi Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit UNJ yang siap digunakan oleh *end user* nantinya.

### 4.4 Aplikasi Hasil Penelitian

Hasil dari penelitian ini menghasilkan beberapa *endpoint* berdasarkan *requirement* dan *product backlog* yang telah disetujui dan dibahas oleh *developer team* bersama dengan *product owner*. 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 Informasi Penerimaan Calon Anggota Baru KSR PMI Unit Universitas Negeri Jakarta di masa mendatang.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Pengembangan *web service* yang menggunakan bahasa pemrograman JavaScript dengan ruang lingkup menggunakan aplikasi Node.JS dengan *framework* Express.Js dan basis data relasional MySQL. Pengembangan dilakukan dengan menggunakan metode *scrum* sebanyak 4 *sprint*. Jumlah *product backlog* yang dibuat *developer team* bersama *product owner* sebanyak 12 item. Pengerjaan *backlog* dibagi ke dalam 4 *sprint*. Pada *sprint* 1 mengerjakan 3 *backlog*, *sprint* 2 mengerjakan 3 *backlog*, *sprint* 3 mengerjakan 3 *backlog*, *sprint* 4 mengerjakan 3 *backlog*. Pengembangan *web service* modul Admin Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit Universitas Negeri Jakarta menghasilkan 58 buah *endpoint*. Pembuatan *endpoint-endpoint* tersebut mengacu pada kebutuhan yang sudah ditentukan oleh *developer team* bersama *product owner*. REST API yang dikembangkan memiliki keamanan yang baik dalam pengolahan dan pemrosesan data dengan menggunakan JWT *validation*.

*Endpoint-endpoint* tersebut telah diuji coba menggunakan metode *unit testing* untuk mengetahui apakah sudah sesuai dengan kebutuhan atau tidak sehingga dapat digunakan untuk membangun suatu *web service*. Hasil pengujian yang menggunakan metode unit testing menunjukkan bahwa setiap *endpoint* yang telah dibuat berfungsi sesuai kebutuhan seperti yang sudah dijabarkan pada Tabel 4.15 sampai Tabel 4.72, sehingga layak digunakan sebagai *web service* modul Admin Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit Universitas

Negeri Jakarta yang dapat memudahkan dalam melaksanakan rangkaian kegiatan penerimaan calon anggota baru KSR PMI Unit Universitas Negeri Jakarta

## 5.2 Saran

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

1. Proses pembayaran uang belum terintegrasi dengan *payment gateway*. Diharapkan pada pengembangan selanjutnya sistem informasi ini dapat diintegrasikan dengan *payment gateway*;
2. Selalu mengecek perkembangan baik teknologi, bahasa pemrograman, dan *framework* sehingga dapat menyesuaikan dengan perkembangan yang terjadi dan mengurangi *error* yang akan datang dikemudian hari;
3. Melakukan dokumentasi yang lebih terstruktur sehingga memudahkan dalam pemenuhan *requirement* baru;
4. Menggabungkan dengan modul lainnya sehingga dapat menjadi satu kesatuan *web service* Sistem Informasi Penerimaan Calon Anggota Baru KSR PMI Unit Universitas Negeri Jakarta;

## Daftar Pustaka:

- BBC Indonesia. (2011, Agustus 11). Retrieved from [https://www.bbc.com/indonesia/berita\\_indonesia/2011/08/110810\\_indonesia\\_tsunami](https://www.bbc.com/indonesia/berita_indonesia/2011/08/110810_indonesia_tsunami)
- BNPB. (2020, Juni 19). *Potensi Ancaman Bencana*. Retrieved from <https://bnpb.go.id/potensi-ancaman-bencana>
- Dewan Perwakilan Rakyat. (2018). *Undang-undang Nomor 1 Tahun 2018*. Jakarta: Dewan Perwakilan Rakyat.
- Doglio, F. (2015). *Pro REST API Development with Node.js*. La Paz, Canelones: Apress.
- Erl, T. (2009). *SOA Design Pattern*. Boston: Pearson Education, Inc.
- Irviani, E. Y. (2017). *PENGANTAR SISTEM INFORMASI*. Yogyakarta: Andi Offset.
- Jogiyanto, H. M. (2005). *Analisis dan Desain (Sistem Informasi Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis)*. Yogyakarta: Andi.
- json.org. (2020, Juni 11). *json.org*. Retrieved from [json.org](https://www.json.org/json-en.html): <https://www.json.org/json-en.html>
- Kadir, A. (2009). *Membuat Aplikasi Web dengan PHP + Database MySQL*. Yogyakarta: Andi.
- Kapojos, F. (2012). Implementasi Service-Oriented Architecture dengan Web Service untuk Aplikasi Informasi Akademik. *Jurnal Teknik Elektro dan Komputer UNSRAT*.
- Kementerian Riset, Teknologi dan Pendidikan Tinggi. (2020, Juni 14). *Pangkalan Data Pendidikan Tinggi - UNJ*. Retrieved from [Pangkalan Data Pendidikan Tinggi: https://forlap.ristekdikti.go.id/perguruantinggi/detail/NTJERDQ0MTEtREREMC00RkU2LUI1RUMtRjZGMzY3REJDRjk3](https://forlap.ristekdikti.go.id/perguruantinggi/detail/NTJERDQ0MTEtREREMC00RkU2LUI1RUMtRjZGMzY3REJDRjk3)
- Marakas, J. A. (2017). *Pengantar Sistem Informasi Edisi 16*. Jakarta: Salemba Empat.
- Mitchell, R. (2015). *Web Scraping with Python: Collecting Data from the Modern Web*. Sebastopol, CA: O'Reilly Media, Inc.
- Oliveira, A. S. (2012). *Enabling Message Security for RESTful Services. Web Services (ICWS)*. Honolulu: IEEE.
- Palang Merah Indonesia. (2008). *Pedoman Manajemen Relawan*. Jakarta: Palang Merah Indonesia.
- Palang Merah Indonesia. (2020, April 25). *Palang Merah Indonesia*. Retrieved from <http://www.pmi.or.id/index.php/tentang-kami/sejarah-pmi>
- Pressman, R. S. (2015). *SOFTWARE ENGINEERING: A PRACTITIONER'S APPROACH, EIGHTH EDITION*. New York: McGraw-Hill Education.
- Proboyekti, U. (2020, April 21). *Agile Software Development*. Retrieved from Agile Software Development: [http://lecturer.ukdw.ac.id/othie/agile\\_model.pdf](http://lecturer.ukdw.ac.id/othie/agile_model.pdf)
- Rubin, K. S. (2013). *Essential Scrum A Practical Guide to the Most Popular Agile Process*. Michigan: Pearson Education, Inc.
- Rubinger, A. a. (2014). *Continuous Enterprise Development in Java: Testable Solutions with Arquillian*. Sebastopol: O'Reilly Media, Inc.
- Simarmata, J. (2010). *Rekayasa Web*. Yogyakarta: Andi.
- Sutabri, T. (2012). *Konsep Sistem Informasi*. Yogyakarta: Andi Offset.
- Sutherland, K. S. (2017). *The Scrum Guide™*.
- Tutorialspoint. (2020, Juni 11). *Node.js*. Retrieved from Tutorialspoint: [https://www.tutorialspoint.com/nodejs/nodejs\\_introduction.htm](https://www.tutorialspoint.com/nodejs/nodejs_introduction.htm)
- Tutorialspoint. (2020, Juni 11). *Restful Web Service*. Retrieved from Tutorialspoint: [https://www.tutorialspoint.com/restful/restful\\_introduction.htm](https://www.tutorialspoint.com/restful/restful_introduction.htm)
- Wardana. (2016). *Aplikasi Website profesional dengan PHP dan jQuery*. Jakarta: Elex Media Komputindo.