

# PENGEMBANGAN WEB SERVICE MENGGUNAKAN FRAMEWORK FASTAPI UNTUK MENINGKATKAN KEMUDAHAN INTEGRASI SISTEM INFORMASI AKADEMIK MULTIPLATFORM

Aprilyani Nur Safitri<sup>1)\*</sup>, Imanuel Harkespan<sup>2)</sup>

<sup>1,2</sup>Program Studi Teknik Informatika

<sup>1,2</sup>Fakultas Ilmu Komputer

<sup>1,2</sup>Universitas Dian Nuswantoro

E-mail: aprilyani.safitri@dsn.dinus.ac.id<sup>1)\*</sup>, harkespan@dsn.dinus.ac.id<sup>2)</sup>

## Abstract

*Academic activities at Dian Nuswantoro University are managed using a web-based and mobile-based (Android and IOS) academic information system for students, lecturers, educators, and parents (guardians). The data retrieval process used by the academic information system is currently in each system itself so that it is prone to errors. Therefore, a back-end service is needed in the form of a web service that acts as a portal for the data retrieval process that can be used by the multiplatform academic information system. In addition to helping to avoid data retrieval errors, the web service that is built also provides complete and easy-to-understand documentation of web service usage. The average time required for the web service to provide a response when accessed by 1000 users is 6198ms (minimum 17ms and maximum 10017ms), meaning that the web service has good performance under high loads. The Extreme Programming method was chosen for the development of the web service in this study. This method consists of four stages, namely planning (analysis of what the system needs), design (visualization with Use Case diagrams), coding (using FastAPI Framework), and the last is testing (using BlackBox and JMeter for testing functions and security). The simplicity of this method can support the achievement of the desired results, namely a back-end service in the form of a web service, which can be used by a multi-platform academic information system to exchange data easily and accurately so that errors can be avoided, especially inconsistencies in presenting academic data.*

**Keywords-** Academic System, Multiplatform, Web Service, FastAPI

## Intisari

*Kegiatan akademik yang ada pada Universitas Dian Nuswantoro dikelola menggunakan sistem informasi akademik mahasiswa, dosen, tenaga pendidik, dan orang tua (wali) mahasiswa berbasis web dan berbasis mobile (Android dan IOS). Proses pengambilan data yang digunakan oleh sistem informasi akademik saat ini berada pada masing-masing sistem itu sendiri sehingga rawan terjadi kesalahan. Maka, diperlukan sebuah layanan back-end berupa web service yang bertindak sebagai portal proses pengambilan data yang bisa digunakan oleh sistem informasi akademik multiplatform tersebut. Selain membantu untuk terhindar dari kesalahan pengambilan data, web service yang dibangun juga menyediakan dokumentasi penggunaan web service yang lengkap dan mudah dipahami. Waktu rata-rata yang dibutuhkan web service untuk memberikan response ketika diakses 1000 user adalah 6198ms (minimal 17ms dan maksimal 10017ms), artinya web service memiliki peforma yang baik pada beban yang tinggi. Metode Extreme Programming dipilih untuk pengembangan web service pada penelitian ini. Metode ini terdiri dari empat tahapan, yaitu planning (analisa apa saja yang dibutuhkan sistem), design (visualisasi dengan Use Case diagram), pengkodean (menggunakan FastAPI Framework), dan yang terakhir adalah pengujian (menggunakan BlackBox dan JMeter untuk pengujian fungsi dan keamanan). Kesederhanaan dari metode ini dapat mendukung tercapainya hasil yang diinginkan yaitu sebuah layanan back-end berupa web service, yang bisa digunakan oleh sistem informasi akademik multipaltform untuk bertukar data dengan mudah dan tepat sehingga dapat terhindar dari kesalahan terutama ketidakkonsistenan dalam menyajikan data akademik.*

**Kata Kunci -** Sistem Akademik, Multiplatform, Web Service, FastAPI.

## 1. PENDAHULUAN

Menurut Unit Data dan Informasi yang ada pada Universitas Dian Nuswantoro (UDINUS), pada semester genap tahun ajaran 2023/2024 terdapat 12016 mahasiswa aktif dan 957 staf yang

meliputi dosen dan tenaga pendidik. Dalam pelaksanaan rutinitas kegiatan akademik yang ada, UDINUS sudah menggunakan Sistem Informasi Akademik dengan berbasis web dan berbasis *mobile* (Android dan IOS) yang bisa digunakan oleh mahasiswa, dosen, tenaga

pendidik, dan orang tua (wali) mahasiswa. Pengertian dari Sistem Informasi itu adalah sistem yang dimiliki oleh organisasi yang didalamnya terdapat gabungan dari kebutuhan pengolahan transaksi yang dilakukan sehari-hari dan fungsi operasi manajerial dari organisasi disertai kegiatan strategi yang dimiliki organisasi sehingga laporan-laporan yang dibutuhkan untuk pihak luar dapat tersedia [1]. Kegiatan Akademik merupakan kegiatan utama yang ada pada sebuah universitas, sehingga untuk kelancaran dalam pelaksanaan kegiatan tersebut, UDINUS menyediakan berbagai macam sistem informasi akademik yang disesuaikan dengan penggunaannya termasuk juga menyediakan sistem yang bisa digunakan *multiplatform* yaitu berbasis web dan berbasis *mobile* (Android dan IOS).

Secara keseluruhan, terdapat lima sistem informasi akademik yang sudah dimiliki oleh UDINUS. Empat sistem yang berbasis web yaitu SiAkad yang digunakan oleh tenaga pendidik (administrasi), SiAdin Dosen yang digunakan oleh dosen, SiAdin Mahasiswa yang digunakan oleh mahasiswa, dan SiAdin Parent yang dapat diakses oleh orang tua (wali) dari mahasiswa. Lalu ada satu sistem berbasis *mobile* yaitu untuk SiAdin Mahasiswa, sistem ini tersedia untuk perangkat yang mendukung sistem operasi *mobile* baik Android maupun IOS. Pada masing-masing sistem tersebut menggunakan data yang bersumber dari *database* pusat yang sama, namun untuk proses pengambilan datanya saat ini masih berada pada masing-masing sistem itu sendiri sehingga rawan terjadi kesalahan karena menggunakan *query* yang terletak didalam masing-masing sistem tersebut.

Permasalahan tersebut dapat diselesaikan dengan membangun sebuah layanan *back-end* berupa *web service* (API) yang bertindak sebagai portal yang melakukan proses *query* pengambilan data yang bisa digunakan oleh sistem informasi akademik *multiplatform*. *Web service* dapat diartikan sebagai layanan dari sebuah sistem atau aplikasi yang berjalan melalui jaringan protokol standar seperti HTTP yang dapat membuat sebuah sistem bisa diakses oleh sistem lain. Dengan menggunakan *web service*, interoperabilitas antar aplikasi yang berbeda bisa menjadi lebih mudah dan mendukung integrasi antar sistem yang

terdistribusi dengan baik [2]. *Back-end* adalah istilah untuk sistem yang berjalan di balik layar website atau aplikasi [3], sehingga walaupun terdapat sistem yang terpisah, dengan menggunakan *web service* tetap dapat terjadi komunikasi dan pertukaran data yang baik antar satu sistem dengan sistem lainnya [4]. Penggunaan *web service* dalam bidang pendidikan terutama pada data universitas dapat sangat membantu, misalnya ketika melakukan integrasi antara data pada sistem akademik universitas dengan aplikasi *Feeder PDDIKTI*, data dapat dengan mudah dikirim ke aplikasi *Feeder PDDIKTI* dalam jumlah besar sekaligus, sehingga data tidak perlu dimasukkan satu persatu [5].

Salah satu arsitektur yang sering digunakan dalam pembuatan *web service* adalah menggunakan teknologi *Representational State Transfer* yang disingkat REST. Pada REST, menggunakan standar arsitektur basis web dengan protokol HTTP agar bisa berkomunikasi data. Sehingga, semua sistem informasi akademik yang dimiliki UDINUS akan melakukan permintaan data melalui pengaksesan *API endpoint* (URL) yang disediakan oleh *web service*, kemudian *web service* akan menjalankan *query* untuk mengambil data dari *database*, lalu hasilnya dari *web service* akan memberikan respon data kepada sistem informasi akademik berupa JSON (*JavaScript Object Notation*) sehingga data bisa digunakan oleh sistem informasi akademik berbasis web maupun berbasis *mobile*.

Selain membantu sistem informasi akademik yang ada pada UDINUS untuk terhindar dari kesalahan penyajian data, *web service* yang dibangun juga harus menyediakan dokumentasi penggunaan layanan yang lengkap dan mudah dipahami sehingga pengembang sistem yang akan menggunakan layanan *web service* tidak mengalami kesulitan. Untuk hal itu, pada bahasa pemrograman web Python terdapat sebuah *framework* bernama FastAPI yang dapat digunakan untuk membangun layanan *back-end web service* sekaligus membuat antarmuka yang baik untuk dokumentasi penggunaan *web service* itu sendiri [6]. *Framework* ini dipilih karena konsep yang digunakan pada FastAPI adalah *Asynchronous Server Gateway Interface* (ASGI) yang memungkinkan *endpoint API* untuk dipanggil dan mengeksekusi perintah

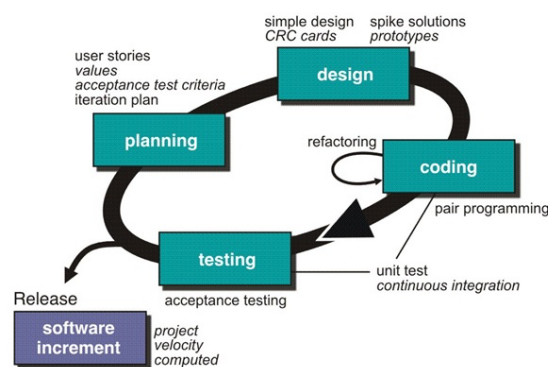
secara asinkron sehingga banyak *request* dapat ditangani secara bersamaan tanpa harus menunggu selesainya operasi yang sedang berjalan [7]. Hal ini membuat *web service* dapat memberikan kinerja yang cepat dan responsif kepada Sistem Informasi Akademik *multiplatform* dengan waktu respon yang efisien ketika diakses oleh banyak pengguna.

Terdapat penggunaan metode pengembangan sistem yang sama pada beberapa penelitian sebelumnya dari berbagai bidang. Pada penelitian dibidang pemerintahan, menghasilkan sistem yang mengelola surat keterangan kependudukan berbasis desktop yang bertujuan untuk peningkatan kualitas pada layanan administrasi [8]. Pada penelitian yang dilakukan dibidang kesehatan, terdapat hasil sistem yang digunakan sebagai sarana konsultasi kepada apoteker mengenai informasi dan pengetahuan tentang obat-obatan dengan berbasis *mobile* (Android) dan berbasis web [9]. Dari penelitian-penelitian tersebut, menjelaskan bahwa metode *Extreme Programming* dapat digunakan dalam berbagai macam pengembangan sistem. Metode ini menggunakan pendekatan berorientasi objek dengan sasaran untuk pengembangan sistem berskala kecil hingga menengah dan dapat mengatasi perubahan permintaan terhadap sistem yang sangat cepat [10]. Hal itu sesuai dengan yang akan dilakukan pada penelitian ini yaitu memfokuskan pada pengembangan layanan *back-end web service* yang bisa digunakan oleh sistem informasi akademik *multiplatform* menggunakan metode *Extreme Programming* (XP). Dengan adanya *web service* ini diharapkan semua sistem informasi akademik yang ada pada UDINUS dapat terhindar dari kesalahan dalam menyajikan data akademik.

## 2. METODOLOGI

Dalam proses pengembangan *web service* untuk Sistem Informasi Akademik *multiplatform* pada UDINUS, peneliti menerapkan metode *Extreme Programming* karena bersifat adaptif serta responsif kepada perubahan kebutuhan yang ada pada system [11], hal ini menjadikan proses pengembangan *web service* akan lebih sederhana dan fleksibel karena permintaan data dari sistem informasi akademik terus berkembang. Pada gambar di bawah ini

menggambarkan tahapan metode dari penelitian yang dilakukan



Gambar 1. Metode *Extreme Programming* [12]

### 2.1. Planning

Pada langkah awal peneliti melakukan wawancara kepada tim pengembang sistem informasi akademik sebelumnya, yaitu para programmer yang ada pada Unit Data dan Informasi sebagai tim pengembang dari seluruh sistem informasi akademik berbasis web dan berbasis *mobile* yang ada pada UDINUS. Kegiatan ini bertujuan untuk melakukan analisa masalah yang muncul pada sistem yang sedang digunakan serta analisa solusi yang ditawarkan [13].

### 2.2. Design

Langkah selanjutnya adalah membuat rancangan *web service*. *Unified Modelling Language (UML)* adalah diagram yang digunakan pada perancangan dalam penelitian ini. *UML* itu sendiri adalah bahasa visual dalam bentuk diagram yang digunakan ketika memodelkan dan menggambarkan komunikasi pada sebuah sistem [14]. Salah satu bentuk dari *UML* adalah *Use Case* diagram, diagram inilah yang akan digunakan pada tahap design penelitian ini karena *Use Case* adalah salah satu diagram yang mampu membuat visual dari interaksi antara aktor dengan *web service* yang akan dikembangkan [15].

### 2.3. Coding

Tahap selanjutnya adalah melakukan pengkodean (*coding*), yaitu dari rancangan yang dibuat ketika tahap *design* kemudian diimplementasikan ke bentuk kode program sehingga dapat dikenali oleh komputer [16]. Metode *Extreme Programming* di dalamnya terdapat praktik *pair programming* atau

kegiatan pengembangan program oleh *programmer* yang dilakukan berpasangan sehingga kualitas perangkat lunak dapat meningkat dan waktu pengiriman menjadi lebih singkat. Secara teknis seperti dua orang *programmer* bekerja bersama disatu lingkungan untuk menyelesaikan sebuah unit fungsi. Dengan menerapkan hal tersebut, keduanya dapat dengan mudah untuk melakukan diskusi dan saling koreksi jika ada kesalahan dalam pembangunan *web service* [12]. Melalui penerapan *pair programming* juga keterampilan komunikasi dapat ditingkatkan sehingga kerja tim menjadi semakin baik [17]. Pada penelitian ini menggunakan *framework* FastAPI dari bahasa pemrograman Python karena dapat membuat layanan *back-end web service* dan sudah termasuk juga dokumentasi penggunaan dari *web service* tersebut. Jika dibandingkan dengan Django yang juga merupakan *framework* dari bahasa pemrograman python, FastAPI lebih tepat untuk masalah yang ada pada penelitian ini karena hanya berfokus pada pembangunan *back-end web service*. Penggunaan Django lebih tepat jika yang dibangun adalah website *full-stack* (*Front End* dan *Back End*). Penggunaan *framework* FastAPI juga terbukti dapat meningkatkan kinerja sistem sebesar 45% dibandingkan ketika menggunakan Flask, *framework* python yang lain [18].

#### 2.4. Testing

Untuk memastikan fungsi-fungsi yang ada pada *web service* dapat berjalan dengan baik, maka tahap pengujian harus diterapkan supaya kemungkinan kesalahan dapat ditemukan pada tahap ini [19][20]. Selain memastikan *web service* dapat berfungsi dengan baik, pengujian terhadap keamanan juga dilakukan supaya terhindar dari serangan seperti SQL (*Structured Query Language*) *Injection* dan XSS (*Cross Site Scripting*). Hal ini untuk mengidentifikasi potensi celah pada sisi keamanan [21]. Proses pengujian dilakukan dengan menerapkan metode *BlackBox* menggunakan *software* JMeter untuk pengujian fungsi dan pengujian keamanan dari setiap *endpoint* dari *web service* yang dibuat.

### 3. HASIL DAN PEMBAHASAN

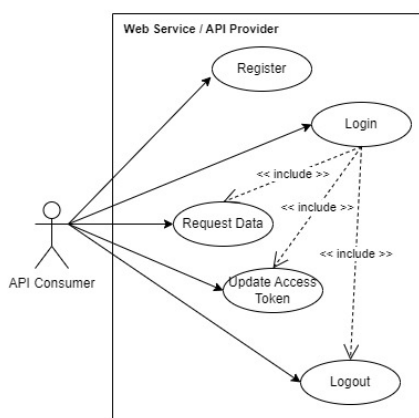
#### 3.1 Planning

Untuk permulaan, langkah paling awal yang dilakukan adalah mengadakan kegiatan wawancara untuk mengetahui keluhan pengguna, hal ini sangat bermanfaat dalam mengumpulkan informasi tentang permasalahan yang pengguna alami. Setelah masalah dapat teridentifikasi, kemudian kebutuhan *web service* yang akan dibangun harus segera disusun. Masalah utama yang ditemukan adalah bagaimana menghindari kesalahan dan ketidakkonsistenan dalam menyajikan data akademik oleh sistem informasi akademik *multiplatform* yang ada di UDINUS. Kebutuhan akan *web service* diketahui dari permasalahan tersebut sehingga perlu adanya pengembangan layanan *back-end* yang mampu menyajikan data akademik yang tepat untuk digunakan oleh sistem informasi akademik *multiplatform* pada UDINUS. Analisa kebutuhan fungsional kemudian disusun berdasarkan hasil identifikasi kebutuhan tersebut karena identifikasi proses, fitur apa saja yang dimiliki oleh *web service*, dan siapa pengguna yang menggunakan layanan tersebut dapat diketahui melalui kebutuhan fungsional [15]. Pengguna *web service* atau biasa disebut dengan *API Consumer* atau *API Client* adalah lima sistem informasi akademik yang ada di UDINUS, yang terdiri dari empat sistem yang berbasis web yaitu Sistem Informasi Akademik Mahasiswa, Dosen, Tenaga Pendidik (administrasi), dan Orang Tua (wali) mahasiswa. Ada juga satu sistem yang berbasis *mobile* (Android dan IOS) untuk Sistem Informasi Akademik Mahasiswa. Berikut ini adalah kebutuhan fungsional dari layanan *back-end web service* pada Universitas Dian Nuswantoro:

1. *Client* dapat melakukan registrasi *username* dan *password*.
2. *Client* dapat melakukan *Login* menggunakan *username* dan *password*.
3. *Client* dapat memiliki *Refresh Token* (masa berlaku tujuh hari) dan *Access Token* (masa berlaku satu hari) setelah berhasil *Login*.
4. *Client* dapat melakukan *request* data akademik menggunakan *Access Token*.
5. *Client* dapat memperbarui *Access Token* yang *expired* menggunakan *Refresh Token*.
6. *Client* dapat melakukan *Logout*.

### 3.2. Design

Tahap selanjutnya yang dilakukan setelah *planning* adalah melakukan perancangan yang didasari oleh analisa kebutuhan fungsional yang sudah diketahui. Berdasarkan hasil analisa kebutuhan fungsional, dapat ditentukan bahwa pengguna pada *web service* ini atau yang disebut sebagai *API Consumer (Client)* adalah sistem informasi akademik *multiplatform* UDINUS. Sehingga hasil visualisasi rancangan tersebut bisa disajikan seperti gambar di bawah ini.



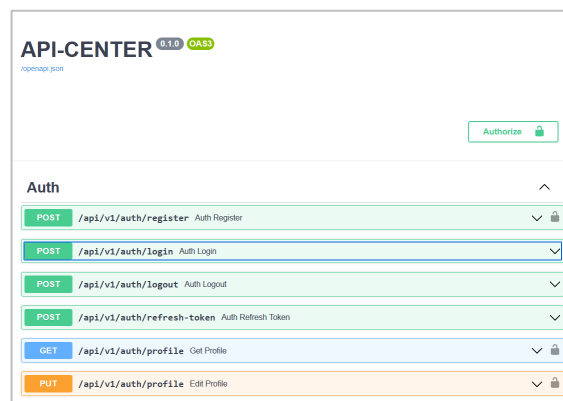
Gambar 2. Diagram Use Case Web Service

Layanan *web service (API Provider)* bisa digunakan oleh sistem informasi akademik *multiplatform* sebagai *API Consumer*. Untuk menggunakan layanan pertama kali harus melakukan registrasi *username* dan *password*. Setelah registrasi berhasil kemudian melakukan Login untuk mendapatkan *Refresh Token* dan *Access Token*. Selanjutnya tinggal melakukan *request data* yang dibutuhkan dengan menyertakan *Access Token* sebagai *authentication*. Masa berlaku *Access Token* hanya satu hari, jika sudah *expired*, bisa mengajukan pembaruan *Access Token* kembali dengan menggunakan *Refresh Token* yang memiliki *expired* lebih lama yaitu tujuh hari. Jika *Refresh Token expired*, *API Consumer* harus melakukan *Login* ulang. Setelah layanan sudah tidak digunakan, bisa melakukan *Logout*.

### 3.3. Coding

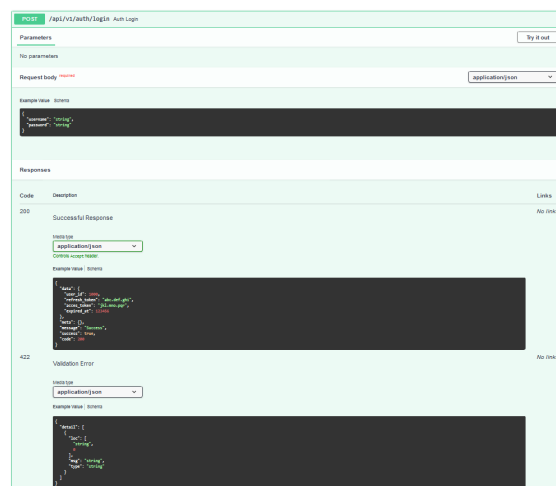
Dalam implementasinya, *web service* dibangun menggunakan *framework* FastAPI dari bahasa pemrograman Python. *Framework* ini dipilih karena memiliki fitur antar muka dokumentasi dari *web service* yang mudah dipahami *API Client* ketika akan menggunakan *endpoint* untuk

kebutuhan data dari sistem informasi akademik. Hasil implementasi *code* ditunjukkan pada Gambar 3.



Gambar 3. Antar muka dokumentasi *web service*

Contoh penggunaan layanan ditunjukkan pada Gambar 4, misalnya pada saat melakukan *Login*, *API Consumer* menggunakan *endpoint* "...api/v1/auth/login" dengan *action POST* mengirimkan *parameter username* dan *password* yang sudah didaftarkan sebelumnya.



Gambar 4. Detail penggunaan *endpoint login*

Setelah berhasil *Login*, lanjutkan dengan melakukan *request data*. Ditunjukkan pada Gambar 5, misalnya untuk mengambil data master mahasiswa menggunakan *endpoint* "...api/v1/mahasiswa/master\_mahasiswa" dengan *action GET* mengirimkan *parameter* kode program studi dan angkatan.

| Name                 | Description                  |
|----------------------|------------------------------|
| proglid<br>(query)   | Default value : E11<br>E11   |
| angkatan<br>(query)  | Default value : 2023<br>2023 |
| nim<br>(query)       | Default value :<br>nim       |
| limitnya<br>(query)  | Default value : 1000<br>1000 |
| offsetnya<br>(query) | Default value : 0<br>0       |

Gambar 5. Contoh request parameter

Hasil response dari web service untuk endpoint master mahasiswa adalah success dengan response code 200, ditunjukkan pada Gambar 6.

Code: 200, Description: Successful Response

```

{
  "data": [
    {
      "proglid": "E11",
      "angkatan": 2023,
      "nim": "123456789",
      "nama": "Budi Santoso",
      "email": "budi.santoso@fakultas.com",
      "password": "12345678"
    },
    {
      "proglid": "E12",
      "angkatan": 2023,
      "nim": "987654321",
      "nama": "Siti Rahmawati",
      "email": "siti.rahmawati@fakultas.com",
      "password": "87654321"
    }
  ],
  "total": 2
}

```

Gambar 6. Contoh response dari web service

### 3.4. Testing

Langkah terakhir yaitu pengujian, dengan menerapkan metode *BlackBox* testing menggunakan software JMeter. Software ini termasuk *software performance testing tool* yang *powerful*, ramah pengguna, dan tidak berbayar. Contoh pengaturan pada JMeter ketika testing endpoint master jalur pendaftaran ditunjukkan pada Gambar 7. Dengan menggunakan JMeter, bisa melakukan simulasi untuk menampilkan hasil response sebuah endpoint saat diakses oleh *API Consumer*. Pada testing dilakukan simulasi pengaksesan web service oleh 1000 user dengan durasi eksekusi request 3 detik per user.

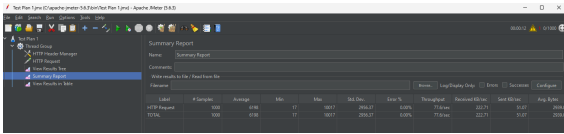
Gambar 7. Setting untuk testing pada JMeter

Hasil response dari JMeter untuk endpoint jalur pendaftaran adalah OK dengan response code 200 ditunjukkan pada Gambar 8.

Gambar 8. Hasil testing endpoint pada JMeter

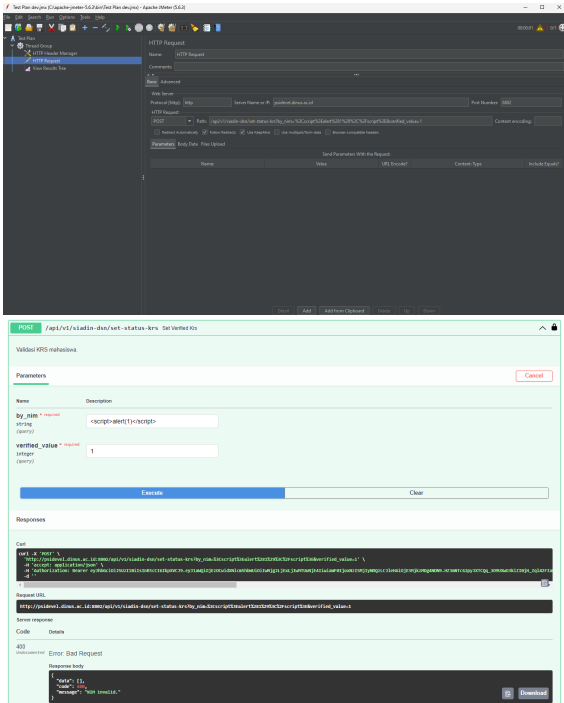


Untuk *summary report* dari 1000 user ditampilkan pada Gambar 9. Waktu rata-rata yang dibutuhkan adalah 6198ms dengan minimal 17ms dan maksimal 10017ms. Tidak terdapat *error* selama simulasi dilakukan (0%).



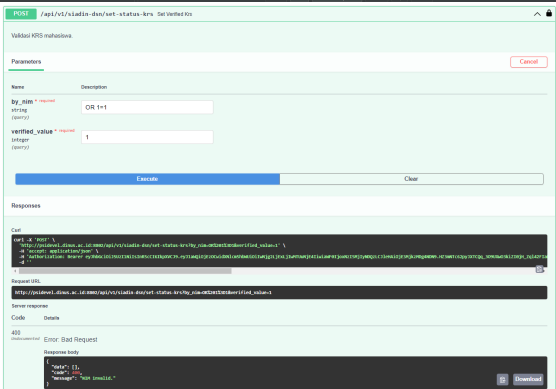
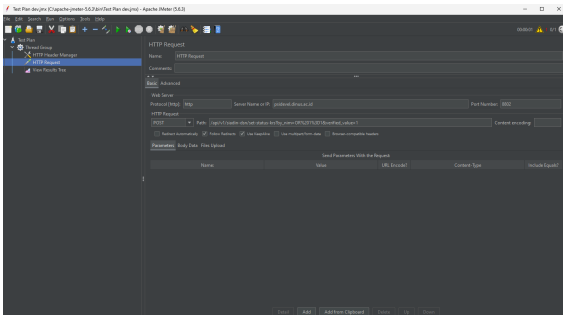
Gambar 9. *Summary report* simulasi 1000 user

Pengujian keamanan sistem dilakukan pada *web service* yang ada pada *server development* sehingga domain yang digunakan berbeda. Pertama coba simulasi serangan XSS (*Cross Site Scripting*) dengan mengirimkan `<script>alert(1)</script>`, hasilnya *web service* memberikan *response* yang menunjukkan bahwa *request* gagal dieksekusi seperti yang ditunjukkan pada Gambar 10.



Gambar 10. *XSS Testing*

Pengujian keamanan kedua adalah dengan mencoba melakukan *SQL Injection* yaitu memasukkan *code SQL* “OR 1=1”, hasilnya *web service* memberikan *response* yang menunjukkan bahwa *request* gagal dieksekusi seperti yang ditunjukkan pada Gambar 11.



Gambar 11. *SQL Injection Testing*

*Testing* dilakukan pada semua *endpoint* yang ada, lalu hasil dari tahap pengujian ini ditampilkan pada Tabel 1.

Tabel 1. *BlackBox Testing*

| Nama                  | Deskripsi  | Hasil  |
|-----------------------|--|--------|
| Registrasi            | Client mendaftarkan <i>username</i> dan <i>password</i>  | Sesuai |
| Login                 | Client login dengan <i>username</i> dan <i>password</i> untuk mendapatkan <i>Access Token</i> dan <i>Refresh Token</i> | Sesuai |
| Request Data Akademik | Client melakukan permintaan data mahasiswa menggunakan <i>Access Token</i>   | Sesuai |
| Update Access Token   | Client melakukan permintaan <i>Access Token</i> yang sudah <i>expired</i>  | Sesuai |
| Logout                | Client melakukan <i>logout</i>   | Sesuai |

#### 4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian pada pengembangan *web service* menggunakan framework FastAPI untuk Sistem Informasi Akademik *multiplatform* pada Universitas Dian Nuswantoro (UDINUS), dapat disimpulkan bahwa pengembangan *web service* dapat dilakukan dengan cepat dan menghasilkan *web service* yang baik dengan menggunakan metode *Extreme Programming*. Dengan adanya *web service* ini, dapat mempermudah pengembang sistem informasi akademik *multiplatform* dalam memperoleh data yang dibutuhkan oleh sistem karena tersedia juga dokumentasi cara penggunaan *web service* yang lengkap dan mudah dipahami sehingga semua sistem informasi akademik yang ada pada UDINUS dapat dapat terhindar dari kesalahan dalam menyajikan data akademik

#### DAFTAR PUSTAKA

- [1] I. Carolina and A. Rusman, "Penerapan Extreme Programming Pada Sistem Informasi Penjualan Pakaian Berbasis Web (Studi Kasus Toko ST Jaya)," *NOVTEK Polbeng - Seri Informatika*, vol. 4, no. 2, pp. 157–167, 2019.
- [2] D. Akbar, F. Riana, and F. Satriya, "Pembuatan Web Service Pada Aplikasi Sijab Dengan Metode REST," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 4, pp. 5567–5575, 2024.
- [3] L. Pradana, A. Ambarwati, and S. D. Putra, "Rancang Bangun Web Service API Dan Dokumentasi Rest API Web Portal Unit Kegiatan Mahasiswa Di Politeknik Negeri Lampung," *Jurnal Sistem dan Teknologi Informasi*, vol. 1, no. 1, pp. 9–18, 2023.
- [4] F. A. Isfahani, F. Nugraha, R. Mubarak, and A. Rahmatulloh, "Implementasi Web Service untuk Aplikasi Pemantau Coronavirus Disease 2019 (COVID-19)," *Innovation In Research Of Informatics*, vol. 2, no. 1, pp. 33–39, 2019.
- [5] M. Hariyanto and H. Hernawati, "Rancang Bangun Aplikasi Web Service Client Feeder PDDIKTI Versi 3.0 Universitas Tarumanagara," *Reputasi: Jurnal Rekayasa Perangkat Lunak*, vol. 1, no. 2, pp. 103–109, Nov. 2020, doi: 10.31294/reputasi.v1i2.113.
- [6] C. V. Suresh babu, V. Surendar, E. Sriram, and S. Subhash, "Web-Based Deep Learning Model for Zero Day Vulnerability Detection using FastAPI," in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, IEEE, Apr. 2024, pp. 1–6. doi: 10.1109/ADICS58448.2024.10533540.
- [7] Pamungkas M.R.S.P, Huda M.N, Fauzan D.A, Itsna A.H, and Al-Hijri F.M, "Sistem Klasifikasi Otomatis Dengan Konsep Machine Learning As A Service (MLaaS) Pada Kasus Pesan Berindikasi Cyberbullying," *Journal of Computer Science and Applied Informatics*, vol. 4, no. 3, pp. 252–261, Dec. 2022.
- [8] L. Rusdiana, "Extreme Programming untuk rancang bangun aplikasi pengelolaan surat keterangan kependudukan," *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 4, no. 1, pp. 49–55, 2018.
- [9] R. Wahyudi *et al.*, "Pengembangan dan Pengujian Aplikasi 'SIATAP' Sistem Informasi Tanya Apoteker Berbasis Android dan Website," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 4, no. 2, pp. 210–218, 2020.
- [10] M. Lamada, A. Bakry, A. Z. Ifani, and K. Khaerunnisa, "Development of Web-Based Project Tender Documents Application Using Extreme Programming Methods," *Elinvo (Electronics, Informatics, and Vocational Education)*, vol. 7, no. 2, pp. 101–111, Feb. 2023, doi: 10.21831/elinvo.v7i2.49863.
- [11] OpenDSA Project, "OpenDSA Data Structures and Algorithms Modules Collection," <https://opensa-server.cs.vt.edu/ODSA/Books/Everything/html/IntroProcess.html>.
- [12] N. A. Septiani and F. Y. Habibie, "Penggunaan Metode Extreme Programming Pada Perancangan Sistem Informasi Pelayanan Publik," *Jurnal Sistem Komputer dan Informatika*, vol. 3, no. 3, pp. 341–349, 2022.



- [13] P. D. Nusantara, "Pengembangan Aplikasi Berbasis Responsive Web Design Dengan Metode Extreme Programming," *Jurnal Informatika dan Komputasi: Media Bahasan, Analisa dan Aplikasi*, vol. 15, no. 1, pp. 61–73, 2021.
- [14] R. D. Gunawan, T. Oktavia, and R. I. Borman, "Perancangan Sistem Informasi Beasiswa Program Indonesia Pintar (PIP) Berbasis Online (Studi Kasus : SMA N 1 Kota Bumi)," *Jurnal Manajemen Informatika*, vol. 8, no. 1, pp. 43–54, 2018.
- [15] H. Sulistiani, "Rancang Bangun Aplikasi Presensi SMS Gateway Berbasis Web Dengan Framework Codeigniter Pada SMKN 1 Trimurjo," *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, vol. 1, no. 1, pp. 43–50, 2020.
- [16] M. Melinda, R. I. Borman, and E. R. Susanto, "Rancang Bangun Sistem Informasi Publik Berbasis Web (Studi Kasus : Desa Durian Kecamatan Padang Cermin Kabupaten Pesawaran)," *Jurnal Tekno Kompak*, vol. 11, no. 1, pp. 1–4, 2018.
- [17] A. Wiguna, C. C. Astuti, and F. N. Hasanah, "Persepsi Mahasiswa terhadap Implementasi Metode Pair Programming pada Pembelajaran Mata Kuliah Pemrograman Dasar," *Jurnal Ilmiah Edutic: Pendidikan dan Informatika*, vol. 9, no. 2, pp. 101–107, 2023.
- [18] P. Bansal and A. Ouda, "Study on Integration of FastAPI and Machine Learning for Continuous Authentication of Behavioral Biometrics," in *2022 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, Jul. 2022, pp. 1–6. doi: 10.1109/ISNCC55209.2022.9851790.
- [19] A. E. Kumala, R. I. Borman, and P. Prasetyawan, "Sistem Informasi Monitoring Perkembangan Sapi Di Lokasi Uji Performance (Studi Kasus : Dinas Peternakan Dan Kesehatan Hewan Provinsi Lampung)," *Jurnal Tekno Kompak*, vol. 12, no. 1, pp. 5–9, 2018.
- [20] N. Fawwazi, M. S. Rohman, N. A. S. Winarsih, Y. P. Astuti, and D. O. Ratmana, "Rancang Bangun Aplikasi Pencatat Kehadiran Asisten Berbasis Android Dengan Metode Agile Untuk Laboratorium Komputer Universitas Dian Nuswantoro," *Jurnal Manajemen Informatika & Sistem Informas*, vol. 7, no. 2, pp. 106–117, 2024.
- [21] T. Anugrah, "Penetration Testing Keamanan Website STIE Samarinda Menggunakan Teknik SQL Injection dan XSS," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 1, Jan. 2024, doi: 10.23960/jitet.v12i1.3882.