

Pengembangan REST API Sistem UIIAdmisi dengan Menggunakan Pendekatan Domain Driven Design

Muhammad Fachri Ramadhan¹, Zainudin Zukhri²

^{1,2} Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 04 Agustus 2023

Revisi Akhir: 18 Agustus 2023

Diterbitkan *Online*: 05 September 2023

KATA KUNCI

UIIAdmisi

Domain Driven Design

Microservices

Back-end

Restful API

KORESPONDENSI

E-mail: fachri.ramadhan@students.uui.ac.id

A B S T R A C T

Student admission program is an event carried out by a higher education institution to accept new students every new academic year. UIIAdmisi system is a web-based admission information system that is used to digitize the implementation of the student admissions program at the Islamic University of Indonesia. This system has been used by the organizers of the Indonesian Islamic University student admissions program from 2016. During its use period, new developers experienced difficulties in developing the back-end side of the UIIAdmisi system because the code in the system was difficult to understand. One of the reasons why the UIIAdmission system is difficult for new developers to understand is because UIIAdmisi solves complex business processes, but the system is built on a monolithic architecture which makes it the code to be bloated. For this reason, the UIIAdmisi system architecture migration process was carried out from monolithic to microservices by separating the existing components of the system into small independent components. These small components will communicate via the HTTP protocol with the implementation of the REST interface. The system will also be designed using the Domain Driven Design approach to make it easier for new developers to understand.

1. PENDAHULUAN

Badan Sistem Informasi Universitas Islam Indonesia adalah badan yang bergerak pada bidang sistem dan teknologi informasi di dalam lingkungan Universitas Islam Indonesia (UII). Tugas yang dipegang oleh Badan Sistem Informasi UII adalah mengawal perencanaan, pengembangan, operasi, serta layanan sistem dan teknologi informasi di lingkungan UII. Badan Sistem Informasi melayani kurang lebih 30.000 pengguna yang merupakan mahasiswa, dosen, tenaga kependidikan dan staf UII lainnya [1].

Badan Sistem Informasi UII memiliki beberapa tim developer professional yang telah mengembangkan banyak sistem dan aplikasi untuk membantu proses didalam UII dalam berbagai aspek. Seperti pada aspek akademik terdapat tim yang membuat aplikasi UIIRAS yang memudahkan mahasiswa untuk melakukan proses key-in di setiap awal semester, pada aspek keuangan terdapat tim yang membuat sistem manajemen gaji staf. Salah satu dari sistem yang dikembangkan oleh Badan Sistem Informasi UII adalah UIIAdmisi.

UIIAdmisi adalah sebuah sistem informasi admisi yang digunakan untuk membantu penyelenggaraan program Penerimaan Mahasiswa Baru (PMB) di UII yang dilaksanakan setiap tahun. Sistem UIIAdmisi ini dikembangkan oleh tim Admisi dan sudah beroperasi lebih dari lima tahun. Namun pada masa sistem ini digunakan, developer baru yang menangani sistem ini menghadapi kesulitan disaat terdapat perubahan pada proses bisnis dan kebijakan pada program PMB. Hal ini terjadi karena sistem UIIAdmisi masih mengimplementasikan arsitektur monolitik, kode yang belum dibuat menjadi lebih modular mengakibatkan developer baru sulit untuk mengerti bagaimana *flow* pada kode [5].

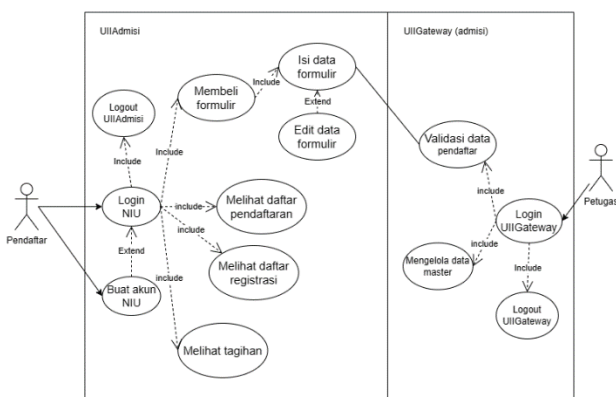
Dari permasalahan tersebut dilakukan proses rekayasa ulang sistem admisi Universitas Islam Indonesia pada sisi *back-end*. Makalah ini akan berfokus pada proses rekayasa ulang *back-end* sistem UIIAdmisi yang dilakukan dengan mengubah arsitektur menjadi *microservices* menggunakan *framework* Laravel Lumen dan mengimplementasikan *design approach domain driven design*. Arsitektur *microservices* digunakan untuk membuat sistem menjadi modular dengan memisahkan sistem menjadi komponen yang kecil sehingga dapat memudahkan proses *maintenance* dan monitoring pada sistem [2]. Tiap *web service*

akan berkomunikasi melalui protokol HTTP (*Hyper Text Transfer Protocol*) dengan menerapkan antarmuka *Restful API*. Pendekatan *domain driven design* digunakan agar desain *web service* sistem dapat disesuaikan dengan ranah proses bisnis yang akan diselesaikan sehingga sistem menjadi lebih mudah untuk dimengerti. Batasan yang diterapkan pada makalah ini yaitu makalah ini hanya akan menjelaskan proses rekayasa ulang sistem admisi Universitas Islam Indonesia secara teknis hanya pada *back-end* sisi pendaftar.

2. TINJAUAN PUSTAKA

2.1. UIIAdmisi

UIIAdmisi adalah sebuah sistem informasi admisi berbasis web yang digunakan untuk mendigitalisasi pelaksanaan program PMB Universitas Islam Indonesia yang dilakukan setiap tahun. Sistem ini dikembangkan dan dikelola oleh Badan Sistem Informasi Universitas Islam Indonesia. Pada program PMB Universitas Islam Indonesia terdapat beberapa pola seleksi yaitu Seleksi Berbasis Rapor (SIBER), Computer Based Test (CBT), Paper Based Test (PBT), Penelusuran Siswa Berprestasi (PSB). Masing-masing pola seleksi tersebut memiliki proses bisnis tersendiri. Sistem ini dibuat dengan dua sisi pengguna, terdapat sisi pendaftar untuk menjalani proses pendaftaran dimulai dari mengisi data hingga melakukan perbaikan. Lalu terdapat sisi pengguna petugas untuk proses verifikasi data dan akan membuat keputusan atas diterima atau tidak diterimanya seorang pendaftar, sisi pendaftar diakses melalui sistem induk yang dimiliki oleh digunakan oleh UII. Gambar 1 merupakan UCD (*Use Case Diagram*) dari UIIAdmisi.



Gambar 1. UCD UIIAdmisi

2.2. Domain Driven Design

Domain driven design merupakan sebuah pendekatan desain perangkat lunak yang menghubungkan konsep domain bisnis dan implementasi teknis. Konsep bisnis tersebut akan di abstraksi oleh ahli domain dan tim teknis menjadi sebuah *domain model*. *Domain model* ini akan menjadi sebuah jembatan komunikasi atau *ubiquitous language* yang menghubungkan tim teknis dengan pihak non-teknis [5]. Dengan mengimplementasikan pendekatan *domain driven design* pengembangan sistem yang kompleks dapat menjadi lebih mudah [10].

Terdapat penelitian [5] yang telah mengkaji pendekatan *domain driven design* untuk mengidentifikasi komponen-komponen *microservices* pada studi kasus migrasi sistem informasi dosen dari arsitektur monolitik menjadi arsitektur *microservices*. Penelitian ini dilakukan dengan cara menganalisis model arsitektur monolitik sistem sebelumnya, lalu menggunakan mendapatkan hasil analisis tersebut sebagai fondasi untuk menentukan *subdomain* dalam perancangan model arsitektur *microservices*. Hasilnya adalah proses migrasi dari arsitektur monolitik menuju arsitektur *microservices* berhasil dilakukan dan menghasilkan sebuah sistem yang *reusable* pada *scope* yang lebih luas.

2.3. Microservices

Microservices merupakan sebuah arsitektur perangkat lunak yang memecahkan kompleksitas menjadi lebih kecil dengan cara memisahkan aplikasi menjadi sejumlah *web service* kecil yang independen. Masing-masing *web service* akan menangani tugas tertentu dan dapat berkomunikasi melalui antarmuka *API* seperti *RESTful API*, *SOAP API*, dan *gRPC*.

Penggunaan arsitektur *microservices* membawa beberapa keunggulan yang tidak didapat pada arsitektur monolitik. Seperti proses pengembangan perangkat lunak yang lebih cepat, dapat menggunakan *techstack* yang berbeda-beda [14], skalabilitas sistem yang lebih baik, *error isolation*, dan membuat sistem menjadi lebih komprehensif [6].

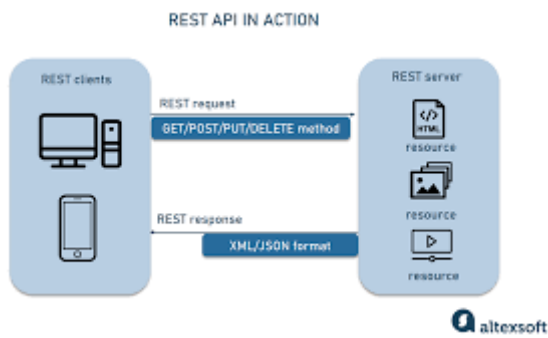
Sebuah penelitian [3] telah membahas tentang penerapan arsitektur *microservices* pada sebuah sistem tata kelola matakuliah proyek. Hasilnya adalah arsitektur *microservice* dapat mempermudah proses pengembangan sistem untuk kedepannya dikarenakan pemisahan komponen pada sistem menjadi terpisah dan mempermudah proses penanganan kesalahan pada sistem karena kesalahan yang terjadi pada satu *web service* tidak akan mengganggu *web service* lain yang sedang berjalan.

Penelitian lain [4] telah mengkaji penerapan arsitektur *microservices* untuk *resiliensi* sistem informasi dengan pendekatan *microservice - Docker Container* pada studi kasus sistem manajemen asosiasi/keanggotaan. Penelitian ini dilakukan dengan melakukan proses *refactoring* pada *software*, lalu mengimplementasikan model arsitektur *microservices*, setelah itu melakukan pengujian pada sistem dengan mengobservasi sistem ketika salah satu *web service* mengalami gangguan. Hasilnya adalah sistem tetap dapat berjalan dan tidak mengganggu sistem lain yang terintegrasi.

2.4. Restful API

Representational State Transfer (REST) merupakan arsitektur perangkat lunak yang biasanya digunakan pada pembuatan *web service*. *RESTful API* adalah sebuah antarmuka yang dibuat menerapkan prinsip REST. *RESTful API* menggunakan protokol *HTTP* untuk melakukan komunikasi antara sistem server dan klien. Alur komunikasi antara klien dan server pada *RESTful API* diawali dengan klien melakukan *HTTP request* ke server *RESTful*, server *RESTful* lalu akan memproses *request* tersebut dan memberi *resource* dalam bentuk *JavaScript Object Notation* (JSON) [7]. *Restful API* dapat diakses dari berbagai perangkat dan mudah untuk dikembangkan [12]. Gambar 2 merupakan

gambaran bagaimana komunikasi yang terjadi antara klien dan server di arsitektur RESTful API.



Gambar 2. Alur kerja REST API [8]

2.5. Back-end

Back-end merupakan sebuah sisi pada sebuah website yang end-user tidak lihat secara langsung berbeda dengan front-end yang merupakan sisi antarmuka yang end-user akan berinteraksi secara langsung. Back-end bertugas untuk membuat aspek fungsional pada website dapat bekerja sehingga dapat membuat website yang dinamis, back-end biasanya juga disebut sebagai sisi server, pada sisi ini bertanggung jawab atas proses interaksi dengan database seperti menyimpan data, mengubah data, menghapus data, dan memberikan data ke sisi front-end untuk dapat ditampilkan ke end-user [7].

3. METODOLOGI

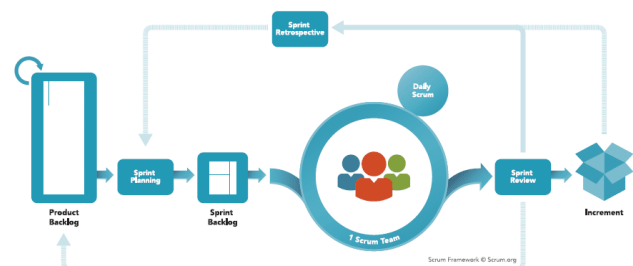
Proses rekayasa ulang sisi back-end pada sistem UIAdmisi dilakukan dengan menggunakan metodologi manajemen pengembangan perangkat lunak agile dengan model scrum. Scrum digunakan untuk dapat mengembangkan sistem dengan cepat, menghasilkan sistem yang berkualitas, dan mudah untuk melakukan adaptasi. Dengan scrum, proses pengembangan sistem dilakukan secara iteratif. Dalam scrum, iterasi ini disebut dengan sprint yang berlangsung selama satu atau empat minggu [9]. Disetiap sprint akan melakukan satu atau lebih tahapan pengembangan sisi back-end sistem sesuai dengan kebutuhan, tahapan-tahapan tersebut yaitu tahap analisis dan desain, tahap pengembangan, dan tahap pengujian. Tahapan tersebut akan dilakukan secara fleksibel untuk mengantisipasi perubahan pada proses bisnis program PMB Universitas Islam Indonesia.

Terdapat beberapa peran didalam scrum [11], yaitu: product owner, project manager, scrum master, dan tim scrum. Product owner berperan untuk mengatur prioritas ketika sprint telah berjalan dan menentukan product backlog ketika sprint pertama belum dimulai dan ketika sprint telah berjalan, product backlog adalah sebuah daftar kebutuhan produk. Project manager berperan memimpin tim pengembangan dalam proses pengembangan dan memastikan proyek dikerjakan semaksimal mungkin. Scrum master berperan untuk membantu tim scrum untuk menegakkan nilai yang terdapat pada scrum. Tim scrum bertugas untuk mengembangkan produk, anggota tim scrum dapat berupa seorang pengembang, analis, dan desainer. Dalam melaksanakan scrum, seluruh peran yang bersangkutan pada

scrum akan menggunakan sebuah alat yaitu Jira untuk memastikan scrum berjalan dengan maksimal.

Didalam scrum terdapat aktivitas yang dinamakan scrum event, aktivitas tersebut yaitu: sprint planning, daily scrum, sprint review, dan sprint retrospective [13]. Sebelum sprint dimulai product owner akan membuat product backlog dan akan menentukan prioritas dari product backlog tersebut, lalu diawal sprint tim scrum akan melakukan sprint planning untuk menentukan product backlog mana saja yang akan dikerjakan pada sprint. Setelah itu proses pengembangan dilakukan. Disetiap hari pengembangan, sebelum memulai bekerja, tim akan melakukan daily scrum dimana anggota tim scrum akan melaporkan ke masing-masing anggota. Lalu pada hari terakhir sprint akan dilakukan sprint review dan sprint retrospective, pada sprint review tim scrum akan melakukan reuiu dengan product owner terhadap proses pengerjaan sprint backlog. Pada sprint retrospective tim scrum akan melakukan refleksi terhadap proses berjalannya sprint dan menentukan sisi apa yang dapat ditingkatkan. Hasil dari sebuah sprint adalah sebuah increment terhadap produk. Gambar 3 merupakan gambaran bagaimana alur kerja metologi scrum.

SCRUM FRAMEWORK



Gambar 3. Alur sprint events pada scrum [5]

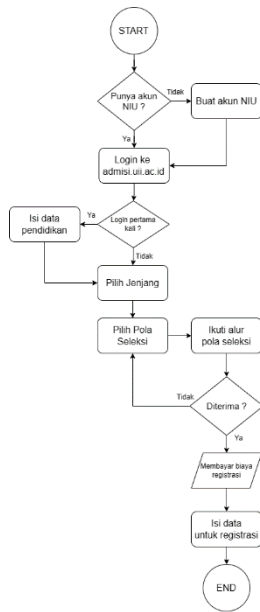
4. HASIL DAN PEMBAHASAN

4.1. Tahap Analisis

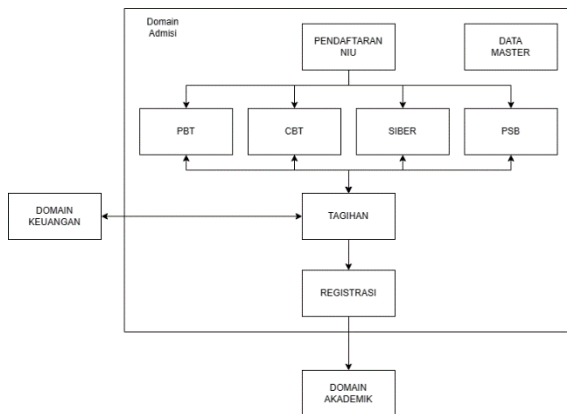
Tahap analisis dan desain ini dilakukan pertama kali sebelum sprint pertama dimulai. Tahap ini bertujuan untuk menentukan rancangan arsitektur sistem yang akan menjadi fondasi pada tahap pengembangan. Namun tidak menutup kemungkinan pada saat sprint pertama telah berjalan untuk kembali ke tahap analisis dan desain lagi ketika mendapati perubahan pada kebutuhan sistem.

Pada tahap analisis, dilakukan analisis pada proses bisnis yang terdapat pada sistem UIAdmisi bagian pendaftar. Gambar 4 merupakan flowchart dari proses bisnis UIAdmisi. Analisis dilakukan pada proses bisnis untuk dapat mengidentifikasi subdomain yang ada pada domain bisnis admisi, proses ini akan menghasilkan daftar subdomain yang ada pada UIAdmisi bagian pendaftar. Subdomain tersebut akan menjadi fondasi untuk pembagian komponen microservices. Subdomain yang teridentifikasi dari hasil analisis, yaitu: pengelolaan pendaftaran, pengelolaan data master, pengelolaan akun pendaftar/NIU, pengelolaan tagihan dan pengelolaan proses registrasi. Gambar 5 merupakan gambaran dari komponen microservices dari sistem UIAdmisi. Pada subdomain pengelolaan pendaftaran akan dibuat beberapa microservice pada masing masing pola seleksi

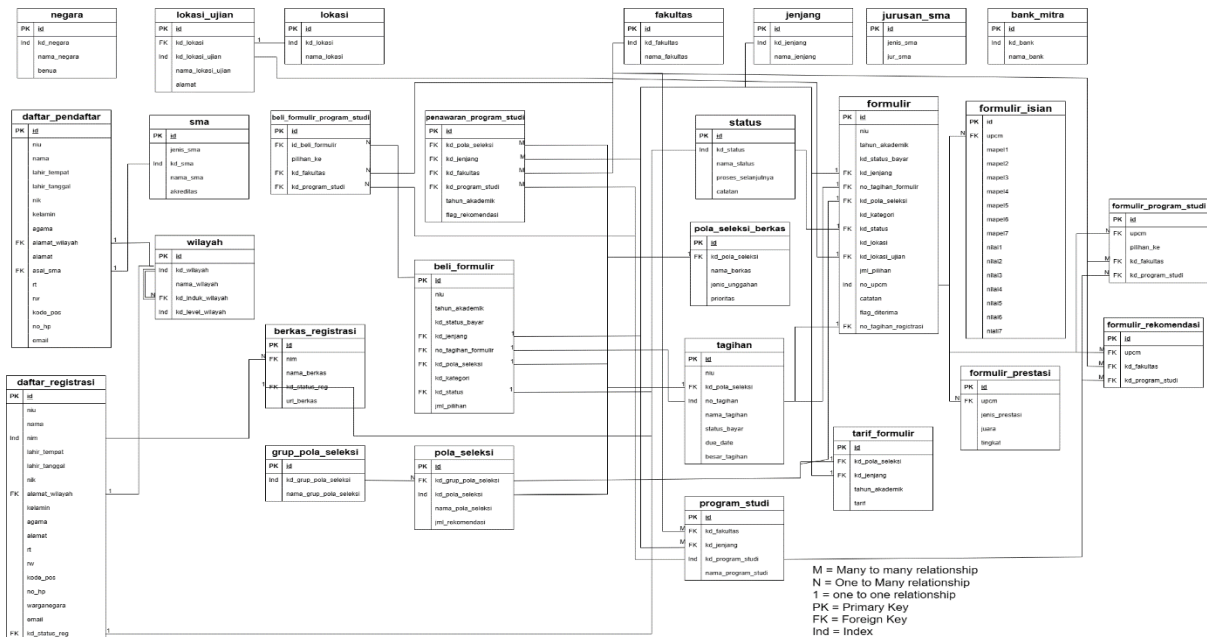
untuk mempermudah proses pengawasan pelaksanaan program PMB UII.



Gambar 4. Flowchart UIIAdmisi



Gambar 5. Komponen *microservices* UIIAdmisi



Gambar 6. ERD basis data UIIAdmisi

Terdapat delapan *web service* yang telah diidentifikasi, yaitu: *web service* pendaftaran NIU, *web service* data master, *web service* PBT, *web service* CBT, *web service* SIBER, *web service* PSB, *web service* tagihan, dan *web service* registrasi. *Web service* pendaftaran NIU akan digunakan untuk melayani proses pendaftaran akun NIU baru dan menyediakan master data yang diperlukan untuk proses pendaftaran, seperti data wilayah provinsi dan kabupaten. *Web service* data master digunakan untuk melakukan proses *Create, Read, Update* dan *Delete* (CRUD) pada data master. *Web service* PBT, SIBER, CBT, dan PSB akan digunakan untuk melayani proses pendaftaran pada masing-masing pola seleksi, mulai dari proses membeli formulir hingga sebelum ke tahap selanjutnya yaitu registrasi, masing-masing pola seleksi tersebut dipisah karena terdapat aturan bisnis yang berbeda diantara pola seleksi tersebut. *Web service* Tagihan akan digunakan untuk mengintegrasikan sisi finansial yang terjadi pada sistem UIIAdmisi seperti proses membeli formulir dan pembayaran registrasi dengan sistem keuangan induk Badan Sistem Informasi Universitas Islam Indonesia. *Web service* registrasi akan digunakan untuk menangani proses integrasi data pendaftar dan melayani pengisian data lengkap pendaftar yang status pendaftarannya telah diterima dan sudah membayar biaya registrasi, menuju sistem akademik Badan Sistem Informasi Universitas Islam Indonesia. Masing masing *web service* akan memiliki basis data tersendiri untuk mengurangi dependensi antara *web service*.

4.2. Tahap Desain

Setelah melakukan analisis dan telah menentukan komponen *microservices* sistem. Selanjutnya adalah melakukan perancangan basis data pada *microservices*. Gambar 6 adalah ERD (*Entity Relationship Diagram*) dari basis data yang akan digunakan oleh UIIAdmisi.

Tabel-tabel pada ERD tersebut akan dikelompokkan lagi ke masing masing *microservice*. Beberapa tabel akan diduplikat pada beberapa *microservice* untuk mengurangi dependensi antar *microservice*. Hasil dari pengelompokkan tabel tersebut yaitu:

1. *Microservice* Pendaftaran NIU
 Pada *microservice* pendaftaran NIU hanya terdapat tabel `daftar_pendaftar`.
2. *Microservice* Data Master
 Pada *microservice* data master terdapat tabel `bank_mitra`, `grup_pola_seleksi`, `fakultas`, `jenjang`, `jurusan_sma`, `lokasi`, `lokasi_ujian`, `status_sma`, `negara`, `wilayah`, `program_studi`,
3. *Microservice* PBT
 Pada *microservice* PBT terdapat tabel `pola_seleksi`, `pola_seleksi_berkas`, `tarif_formulir`, `formulir`, `formulir_program_studi`, `formulir_rekomendasi`, `beli_formulir`, `beli_formulir_program_studi`.
4. *Microservice* CBT
 Pada *microservice* CBT terdapat tabel `pola_seleksi`, `pola_seleksi_berkas`, `tarif_formulir`, `formulir`, `formulir_program_studi`, `formulir_rekomendasi`, `beli_formulir`, `beli_formulir_program_studi`.
5. *Microservice* SIBER
 Pada *microservice* SIBER terdapat tabel `beli_formulir`, `beli_formulir_program_studi`, `formulir`, `formulir_isian`, `formulir_program_studi`, `formulir_rekomendasi`, `pola_seleksi`, `pola_seleksi_berkas`, `tarif_formulir`, `penawaran_program_studi`.
6. *Microservice* PSB
 Pada *microservice* PSB terdapat tabel `beli_formulir`, `beli_formulir_program_studi`, `formulir`, `formulir_isian`, `formulir_program_studi`, `formulir_rekomendasi`, `formulir_prestasi`, `pola_seleksi`, `pola_seleksi_berkas`, `tarif_formulir`, `penawaran_program_studi`
7. *Microservice* Tagihan
 Pada *microservice* tagihan hanya terdapat tabel `daftar_tagihan`
8. *Microservice* Registrasi
 Pada *microservice* registrasi terdapat tabel `daftar_registrasi` dan `berkas_registrasi`.

4.3. Tahap Implementasi

Setelah membuat rancangan sistem yang telah dibuat pada tahap analisis dan desain. Mulailah dibuat *web service* dengan dalam bentuk *Restful API*. Tabel 1 hingga 8 merupakan hasil dari rancangan *web service* berdasarkan fungsionalitas pada masing masing *web service*.

Tabel 1. Hasil Web Service Pendaftaran NIU

URI	Method
-----	--------

180 Muhammad Fachri Ramadhan

/registration	POST
/validate/email/{email}	GET
/validate/nik/{idNumber}	GET
/validate/paspor/{passport}	GET

Tabel 2. Hasil Web Service Data Master

URI	Method
/master/stage	GET
/master/stage/selection-pattern-group	GET
/master/high-school	GET, POST
/master/high-school-type	GET, POST
/master/high-school/update	PUT
/master/major	GET
/master/nation	GET
/master/region	GET, POST, PUT
/master/region-level	GET
/master/region-induk	GET
/master/selection-pattern-group	GET
/master/partner-bank	GET
/master/region	GET
/master/province/{countryCode}	GET
/master/district/{provinceCode}	GET
/master/subdistrict/{districtCode}	GET
/master/religion	GET
/master/citizen	GET
/master/education-level	GET
/master/sesi	GET
/master/selection-pattern	GET
/master/selection	GET
/master/academic-year	GET
/master/wave	GET
/master/batch	GET
/master/status	GET
/master/elective-study-program	GET
/master/test-location	GET
/master/test-location/{kodeLokasiUjian}	GET
/master/program-study	GET
/master/category	GET
/master/achievement	GET
/master/achievement-field	GET
/master/achievement-type	GET
/master/achievement-level	GET
/master/achievement-rank	GET
/master/file-selection-pattern	GET

Tabel 3. Hasil Web Service Tagihan

URI	Method
/bill/{niu}	GET
/bill/{idTagihan}	GET, POST

Tabel 4. Hasil Web Service Registrasi

URI	Method
/personal-data	GET, POST, PUT
/personal-family	GET, POST, PUT
/personal-language	GET, POST, PUT
/personal-talent	GET, POST, PUT, DELETE

/personal-educational-stage	GET, POST, PUT, DELETE
/personal-address	GET, POST, PUT, DELETE
/personal-verification-document	GET, POST, PUT
/registration-personal	POST
/registration-achievement	POST
/personal-registration-list	GET, PUT

Tabel 5. Hasil Web Service PBT

URI	Method
/study-program-form/	GET
/purchase-form/	POST
/form/selection-pattern	GET
/form/organization	GET
/list-pbt	GET, POST
/list-pbt/{idFormulir}	GET, PUT

Tabel 6. Hasil Web Service CBT

URI	Method
/study-program-form/	GET
/purchase-form/	POST
/form/selection-pattern	GET
/form/organization	GET
/purchase-study-program-form	GET, POST
/list-cbt	GET, POST
/list-cbt/{idFormulir}	GET, PUT,
/recommendation-form	GET, POST

Tabel 7. Hasil Web Service SIBER

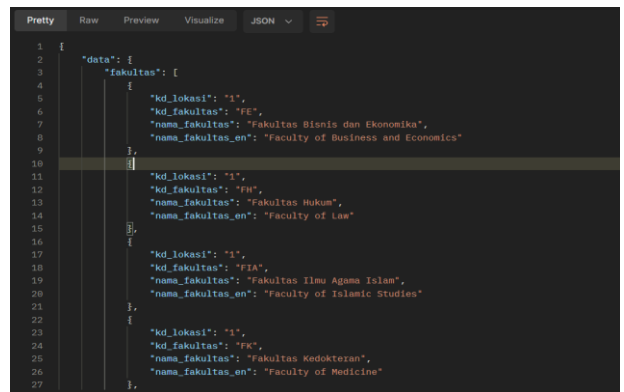
URI	Method
/study-program-form/	GET
/purchase-form/	POST
/form/selection-pattern	GET
/form/organization	GET
/purchase-study-program-form	GET, POST
/fill-out-form	GET, POST, PUT
/achievement-form	GET, POST
/achievement-form/{id}	PUT, DELETE
/list-siber	GET, POST
/list-siber/{idFormulir}	GET, PUT,
/recommendation-form	GET, POST

Tabel 8. Hasil Web Service PSB

URI	Method
/study-program-form/	GET
/purchase-form/	POST
/form/selection-pattern	GET
/form/organization	GET
/purchase-study-program-form	GET, POST
/fill-out-form	GET, POST, PUT
/achievement-form	GET, POST
/achievement-form/{id}	PUT, DELETE
/list-psb	GET, POST
/list-psb/{idFormulir}	GET, PUT,
/recommendation-form	GET, POST

4.4. Tahap Pengujian

Pengujian dilakukan dengan melakukan pengujian API secara manual menggunakan Postman seperti pada Gambar 7, lalu memastikan bahwa klien dapat melakukan *request* ke *URI* pada *web service* dan memastikan *web service* dapat berjalan dengan baik dan sesuai dengan kebutuhan pada proses bisnis. Setelah dilakukan pengujian secara manual, berikutnya adalah meluncurkan *web service* menuju lingkungan staging untuk diuji bersamaan dengan sisi *front-end* oleh penguji. Penguji kemudian akan memberi tanggapan dan kemudian akan dilakukan kembali pengembangan sesuai dengan tanggapan penguji. Setelah *web service* telah diuji, selanjutnya adalah melakukan *deploy* menuju API Gateway milik Badan Sistem Informasi Universitas Indonesia untuk digunakan oleh *end-user*.



Gambar 7. Pengujian *web service* menggunakan Postman

5. KESIMPULAN DAN SARAN

Dari hasil pengembangan dapat disimpulkan bahwa proses rekayasa ulang *back-end* sistem UIIAdmisi berhasil dilakukan. Pemisahan komponen yang diterapkan pada arsitektur *microservice* membuat *web service* skalabilitas sistem menjadi lebih baik dan memungkinkan untuk mengembangkan. Proses pengembangan kedepannya akan menjadi lebih mudah.

Perancangan dengan menggunakan pendekatan *domain driven design* menghasilkan sebanyak delapan *web services*. Delapan *web services* tersebut akan melayani proses bisnis masing-masing sehingga membuat sistem menjadi lebih mudah untuk dimengerti bagi pengembang baru yang menangani sistem ini. Saran untuk pengembangan UIIAdmisi kedepannya adalah melakukan normalisasi basis data untuk mengurangi redundansi data pada tabel.

DAFTAR PUSTAKA

- [1] Badan Sistem Informasi Universitas Islam Indonesia, "Sekilas Tentang BSI," [Online]. Available: <https://bsi.uui.ac.id/sekilas-bsi/>.
- [2] B. I. Nirwana, "Microservices: Pengertian, Contoh, dan Kelebihannya," 8 July 2022. [Online]. Available: <https://www.niagahoster.co.id/blog/microservices/>.
- [3] I. W. A. J. Pawana, D. M. Wiharta and N. P. Sastra, "Identifikasi Kandidat Microservies Dengan Analisis Domain Driven Design," *Majalah Ilmiah Teknologi Elektro*, vol. 20, no. 2, pp. 273-279, 2021.

- [4] M. H. K. Saputra and L. M. Nabil, "PENERAPAN ARSITEKTUR MICROSERVICE PADA SISTEM TATA KELOLA MATAKULIAH PROYEK POLITEKNIK POS INDONESIA," *Jurnal Teknik Informatika*, vol. 13, no. 3, pp. 22-28, 2021.
- [5] H. Suryotrisongko, "Arsitektur Microservice untuk Resiliensi Sistem Informasi," *Jurnal Sisfo*, vol. 06, no. 02, pp. 235-250, 2017.
- [6] I. Nadareushvuku, R. Mitra, M. Mclarty and M. Amundsen, *Microservice Architecture: Aligning Principles, Practice, and Culture*, Sebastopol: O'Reilly Media, Inc., 2016.
- [7] A. M. Yudha and A. B. Cahyono, "Pengembangan Back End Menggunakan Laravel Lumen (Studi Kasus: Teknologi.id Event)," *Automata*, vol. 3, no. 2, 2022.
- [8] Altexsoft, "REST API: Key Concepts, Best Practices, and Benefits," software r&d engineering, 19 November 2022. [Online]. Available: <https://www.altexsoft.com/blog/rest-api-design/>.
- [9] S. Pratama, S. Ibrahim and M. A. Reybaharsyah, "Penggunaan Metode Scrum Dalam Membentuk Sistem Informasi Penyimpanan Gudang Berbasis Web," *Jurnal Informatika dan Teknologi (INTECH)*, vol. 3, no. 1, pp. 27-35, 2022.
- [10] Nuralamsyah, R. J. Akbar and H. Fabroyir, "Rancang Bangun Modul Job Marketplace di Aplikasi MyITS Connect Berdasarkan Onion Architecture dengan Paradigma Domain Driven Design," *Jurnal Teknik ITS*, vol. 10, no. 2, pp. A99-A105, 2021.
- [11] M. Mahalakshmi and M. Sundararajan, "Traditional SDLC Vs Scrum Methodology - A Comparative Study," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 6, pp. 192-196, 2013.
- [12] I. A. K. P. Paramitha, D. M. Wiharta and I. M. A. Suyadna, "PERANCANGAN DAN IMPLEMENTASI RESTFUL API PADA SISTEM INFORMASI MANAJEMEN DOSEN UNIVERSITAS UDAYANA," *Jurnal SPEKTRUM*, vol. 9, no. 3, pp. 15-23, 2022.
- [13] S. Hadji, M. Taufik and S. Mulyono, "IMPLEMENTASI METODE SCRUM PADA PENGEMBANGAN APLIKASI DELIVERY ORDER BERBASIS WEBSITE (STUDI KASUS PADA RUMAH MAKAN LOMBOK IDJO SEMARANG)," in *PROSIDING KONFERENSI ILMIAH MAHASISWA UNSSULA (KIMU) 2*, Semarang, 2019.
- [14] A. Sinambela, Ernawati and F. F. Coastera, "IMPLEMENTASI ARSITEKTUR MICROSERVICES PADA RANCANG BANGUN APLIKASI MARKETPLACE BERBASIS WEB," *Jurnal Rekursif*, vol. 9, no. 1, pp. 1-13, 2021.
- [15] S. Najmaa, "Apa itu Scrum? Peran, Proses, dan Manfaatnya," STMIK Indo Daya Suvana, [Online], Available: <http://ids.ac.id/apa-itu-scrum/>.

BIODATA PENULIS



Muhammad Fachri Ramadhan

Mahasiswa Program Studi Informatika
Fakultas Teknologi Industri Universitas
Islam Indonesia.

Email: fachri.ramadhan@students.uui.ac.id



Zainudin Zukhri, S.T., MIT.

Dosen Program studi Informatika Fakutlas
Teknologi Industri Universitas Islam
Indonesia yang mendalami bidang
kecerdasan buatan, optimasi, dan pengenalan
pola.

Email: zainudin@uui.ac.id