

Desain dan Analisis Arsitektur *Microservices* Pada Sistem Informasi Akademik Perguruan Tinggi Dengan Pendekatan *Architecture Tradeoff Analysis Method (ATAM)* (Studi Kasus: iGracias Universitas Telkom)

Design and Analysis of *Microservices* Architecture On Academic Information System With Higher Education Approach *Architecture Tradeoff Analysis Method (ATAM)* (Case Study: iGracias Telkom University)

Muhammad Rezaldy, Ibnu Asror, S.T., M.T², Indra Lukmana Sardi, S.T., M.T³

^{1, 2, 3} Prodi S1 Teknik Informatika, Fakultas Informatika, Universitas Telkom
Bandung, Indonesia

¹ rezaldy22@gmail.com, ² iasror@telkomuniversity.ac.id, ³ indra.luk23@gmail.com

Abstrak

Microservices merupakan sebuah konsep arsitektur perangkat lunak yang dapat menjadi solusi untuk dapat membuat perangkat lunak yang besar lebih terorganisasi dan dapat dengan cepat beradaptasi terhadap perkembangan kebutuhan bisnis yang ada. Bagi lembaga perguruan tinggi teknologi sistem informasi telah menjadi kebutuhan untuk menunjang proses pendidikan. Sebagai media informasi akademik yang memiliki peran penting, iGracias masih mengadopsi arsitektur monolitik. Monolitik yaitu aplikasi yang dikembangkan dan di-deploy sebagai satu entitas [1]. Hal ini mengakibatkan ketika suatu aplikasi monolitik berkembang menjadi sangat besar dan kompleks, akan menjadi sangat sulit untuk proses pengembangan lanjutan, pengujian dan *deploy*. *Microservices* merupakan sebuah kerangka kerja yang dapat mengintegrasikan proses bisnis yang ada, mendukung infrastruktur teknologi informasi sesuai dengan prioritas bisnis. Desain dan Analisis arsitektur *microservices* pada sistem akademik perguruan tinggi dengan pendekatan *architecture tradeoff analysis Method(ATAM)* (Studi Kasus: iGracias Universitas Telkom) sebagai metode evaluasi arsitektur *microservices*. Dalam tugas akhir ini telah dilakukan desain dan analisis terhadap arsitektur *microservices* untuk diterapkan pada iGracias. Dari hasil survey dan pengujian yang menggunakan scenario sebagai bahan penilaian terdapat empat komponen yang dapat menangani permasalahan sistem informasi akademik iGracias yaitu dengan *microservices* yang *modularity, independent, maintainability* dan *scalability*.

Kata kunci: ***Microservices, monolitik, iGracias, Arsitektur.***

Abstract

Microservices is a software architecture concept that can be a solution to make large software more organized and can quickly adapt to the development of existing business needs. For institutional colleges of information systems technology has become a necessity to support the educational process. As an academic information media that has an important role, iGracias still adopt monolithic architecture. Monolithic application is developed and deployed as one entity [1]. This results when a monolithic application develops into a very large and complex, it will become very difficult for advanced development process, testing and deployment. *Microservices* is a framework that can integrate existing business processes, supporting information technology infrastructure according to business priorities. Design and Analysis of *microservices* architecture in academic system of college with approach of *architecture tradeoff analysis Method (ATAM)* as method of evaluation of *microservices* architecture. In this final project, design and analysis of *microservices* architecture have been applied to iGracias. From the results of surveys and tests using scenarios as assessment materials there are four components that can handle the problem of iGracias academic information system that is with *microservices* that *modularity, independent, maintainability* and *scalability*.

Keywords: ***Microservices, monolithic, iGracias, Architecture.***

1. Pendahuluan

Microservices merupakan sebuah konsep arsitektur perangkat lunak yang dapat menjadi solusi untuk dapat membuat

perangkat lunak yang besar lebih terorganisasi dan dapat dengan cepat beradaptasi terhadap perkembangan kebutuhan bisnis yang ada [1]. Pada prinsipnya *microservices* adalah arsitektur perangkat lunak untuk mengembangkan suatu aplikasi yang dibentuk dari susunan beberapa *service* yang kecil, berjalan dengan prosesnya sendiri dan berkomunikasi menggunakan mekanisme yang sederhana, sering kali menggunakan HTTP sebagai media komunikasi API [2].

Sistem informasi akademik iGracias Universitas Telkom merupakan portal sistem informasi berbasis web yang digunakan oleh mahasiswa Universitas Telkom. Sebagai media informasi akademik yang memiliki peran penting, iGracias masih mengadopsi arsitektur monolitik. Monolitik yaitu aplikasi yang dikembangkan dan di-deploy sebagai satu entitas [1]. Hal ini mengakibatkan ketika suatu aplikasi monolitik berkembang menjadi sangat besar dan kompleks, akan menjadi sangat sulit untuk proses pengembangan lanjutan, pengujian, perbaikan dan deployment. Hal yang terjadi pada iGracias pada saat ini yaitu ketika proses *maintenance* terhadap teknologi baru ini mengakibatkan seluruh sistem menjadi terganggu dan perlu *maintenance* secara keseluruhan dan jika terjadi kesalahan dalam satu sistem aplikasi maka perlu perbaikan secara keseluruhan aplikasi yang terkait dengan permasalahan yang ditimbulkan.

Berdasarkan permasalahan diatas, Perlunya analisis dan desain terhadap iGracias dengan arsitektur *microservices*. Karena *microservices* merupakan sebuah kerangka kerja yang dapat mengintegrasikan proses bisnis yang ada, mendukung infrastruktur teknologi informasi dan juga komponen layanan yang dapat digunakan kembali dan digabungkan sesuai dengan prioritas bisnis. Hal ini dapat memudahkan dalam proses pengembangan lanjutan, pengujian, perbaikan dan *deployment*.

Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan diatas, maka dapat ditarik beberapa rumusan masalah, yaitu:

1. Bagaimana mendesain iGracias Universitas Telkom menggunakan konsep arsitektur *microservices*?
2. Bagaimana perbandingan arsitektur yang diterapkan pada iGracias pada saat sekarang ini dengan arsitektur *microservices*?

Tujuan

Tujuan dari penelitian tugas akhir ini adalah sebagai berikut:

1. Mendesain konsep arsitektur *microservices* pada sistem informasi akademik iGracias Universitas Telkom.
2. Menganalisis dan mengevaluasi penggunaan arsitektur *microservices* pada iGracias.

2. Tinjauan Pustaka

2.1 Microservices

Microservices adalah arsitektur perangkat lunak untuk mengembangkan suatu aplikasi yang dibentuk dari susunan berbagai *service* yang kecil, berjalan dengan prosesnya sendiri, dan berkomunikasi menggunakan mekanisme yang sederhana, seringkali menggunakan HTTP sebagai media komunikasi API [1].

Komponen adalah unit perangkat lunak yang berdiri sendiri (*independent*), *replaceable*, dan *upgradeable* [2]. *Microservices* dapat dipandang sebagai komponen perangkat lunak.

Untuk melakukan perubahan dari arsitektur monolitik ke *microservices* ada tiga tahap untuk melakukanya yaitu *stop digging*, *split fronted and backend* dan *extrack services* [3]:

1. **Stop Digging**, berhenti mengali aplikasi monolitik dan harus berhenti membuat monolitik lebih besar lagi. Ini berarti bahwa ketika Anda menerapkan fungsi baru Anda tidak boleh menambahkan lebih banyak kode ke monolitik. Sebagai gantinya, gagasan besar dengan strategi ini adalah memasukkan kode baru itu ke dalam *microservices*.
2. **Split Fronted and Backed**, Sebuah strategi yang mengecilkan aplikasi monolitik adalah dengan membagi lapisan presentasi dari logika bisnis dan lapisan akses data. Aplikasi perusahaan yang khas terdiri dari setidaknya tiga jenis komponen yang berbeda:
 - a. Lapisan presentasi - Komponen yang menangani permintaan HTTP dan menerapkan API (REST) atau UI web berbasis HTML. Dalam sebuah aplikasi yang memiliki antarmuka pengguna yang canggih, tingkat presentasi seringkali merupakan kumpulan kode yang besar.
 - b. *Business logiclayer* – *Components that are the corehe* aplikasi dan menerapkan peraturan bisnis

- c. Lapisan akses data - Komponen yang mengakses komponen infrastruktur, seperti database dan pialang pesan
3. **Extract Services**, ketiga adalah mengubah modul yang ada dalam monolit menjadi microservices mandiri Setiap kali Anda mengekstrak modul dan mengubahnya menjadi layanan, monolit menyusut Setelah Anda cukup mengubah modul, monolit akan berhenti menjadi masalah Entah itu akan hilang sama sekali. Atau menjadi cukup kecil sehingga hanya layanan lain. Memprioritaskan Modul mana untuk Dikonversi ke Layanan aplikasi monolitik besar dan kompleks terdiri dari puluhan atau ratusan modul, yang kesemuanya merupakan kandidat untuk ekstraksi.

2.2 ATAM: Method for Architecture Evaluation

ATAM adalah metode analisis yang disusun berdasarkan gagasan, dan arsitektur merupakan penentu utama kualitas atribut arsitektural. Metode ini berfokus pada identifikasi tujuan bisnis yang mengarah pada sasaran kualitas atribut. Berdasarkan tujuan atribut kualitas, tujuan ATAM untuk menganalisis bagaimana gaya arsitektur yang digunakan dapat membantu tercapainya tujuan arsitektur tersebut. Langkah-langkah dari metode ini adalah sebagai berikut [4]:

a. Presentasi

1. **Present the ATAM.** Metode ini dijelaskan kepada pemangku kepentingan yang berkumpul (biasanya perwakilan pelanggan, arsitek atau tim arsitektur, perwakilan pengguna, operator utama, administrator, manajer, penguji, integrator, dll.).
2. **Present business driver.** Manajer proyek menjelaskan tujuan bisnis apa yang sedang memotivasi upaya pengembangan dan karenanya apa yang akan menjadi driver arsitektur utama (mis., Ketersediaan tinggi atau waktu ke pasar atau keamanan tinggi).
3. **Present architecture.** Arsitek akan menjelaskan arsitektur yang diusulkan, memusatkan perhatian pada bagaimana alamat driver bisnis.

b. Investigasi dan Analisis

1. **Identify architectural approaches.** Pendekatan arsitektur diidentifikasi oleh arsitek, namun tidak dianalisis.
2. **Generate quality attribute utility tree.** Faktor kualitas yang terdiri dari sistem "utilitas" (kinerja, ketersediaan, keamanan, modifikasi, dll.) Diperoleh, ditentukan sampai pada tingkat skenario, yang diberi catatan dengan rangsangan dan tanggapan, dan diprioritaskan.
3. **Analyze architectural approaches.** Berdasarkan faktor prioritas tinggi yang diidentifikasi pada Langkah 5, pendekatan arsitektural yang membahas faktor-faktor tersebut diperoleh dan dianalisis (misalnya, pendekatan arsitektural yang ditujukan untuk memenuhi sasaran kinerja akan ditunjukkan pada analisis kinerja). Selama ini risiko arsitektural, titik sensitivitas, dan titik tradeoff diidentifikasi.

c. Pengujian

1. **Brainstorm and prioritize scenarios.** Berdasarkan skenario eksemplar yang dihasilkan di pohon utilitas, serangkaian skenario yang lebih besar diperoleh dari keseluruhan kelompok pemangku kepentingan. Skenario ini diprioritaskan melalui proses pemungutan suara yang melibatkan seluruh kelompok pemangku kepentingan.
2. **Analyze architectural approaches.** Langkah ini mengulangi langkah 6, tapi di sini skenario peringkat yang sangat tinggi dari Langkah 7 dianggap sebagai kasus uji untuk analisis pendekatan arsitektural yang ditentukan sejauh ini. Skenario uji coba ini dapat mengungkap pendekatan arsitektural, risiko, titik sensitivitas, dan titik pertukaran yang kemudian didokumentasikan.

d. Pelaporan

Present results. Berdasarkan informasi yang dikumpulkan di ATAM (gaya, skenario, pertanyaan khusus atribut, pohon utilitas, risiko, titik sensitivitas, pengorbanan), tim ATAM menyajikan temuan tersebut kepada pemangku kepentingan yang berkumpul dan berpotensi menulis laporan yang merinci informasi ini bersamaan dengan Strategi mitigasi yang diusulkan.

3. Perancangan sistem

Alur pemodelan dari penelitian ini dapat dilihat pada Figure 3.1 yang mengacu pada tahapan *microservices*

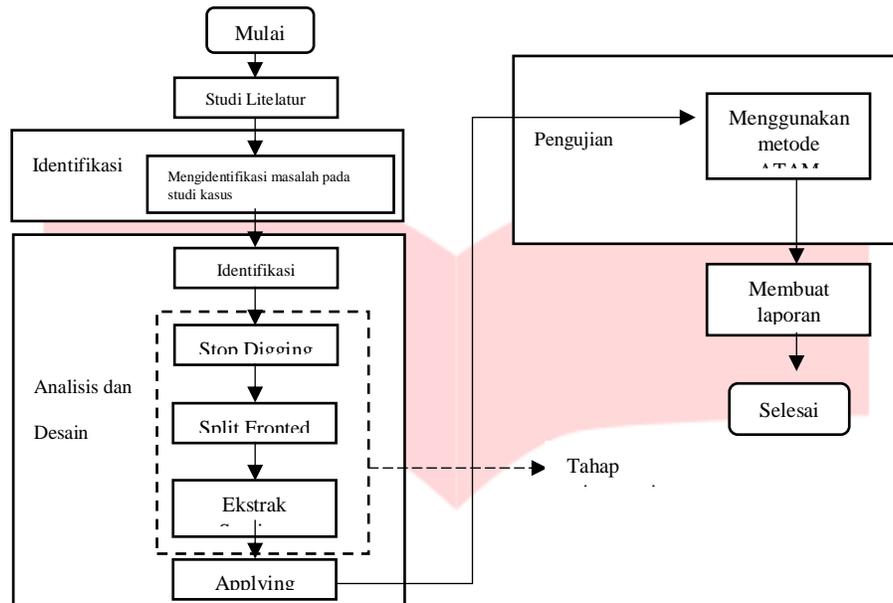


Figure 3. 1 Alur Pemodelan

3.1 Studi Literatur

Dalam tahap ini dilakukan teknik penyusunan yang sistematis untuk memudahkan langkah-langkah yang akan diambil. Begitu pula yang dilakukan penulis dalam penelitian ini, langkah pertama yaitu dengan melakukan studi literatur pada buku-buku yang membahas tentang *theory of constraints*, jurnal, dan penelitian yang telah dilakukan yang berkaitan dengan *microservices*. Data yang didapat dari studi literatur ini akan digunakan sebagai acuan untuk membuat analisis dan desain tentang penerapan arsitektur *microservices*.

3.2 Identifikasi Masalah

Pada tahap ini dilakukan identifikasi masalah pada studi kasus yang akan digunakan pada penelitian ini. Dalam menemukan masalah penelitian terlebih dahulu dilakukan kegiatan berpikir dan merenung guna memperoleh gagasan, ide dan motivasi untuk melakukan suatu penelitian. Dalam proses ini dilakukan pengamatan atau survei mengenai subyek dan obyek dari masalah yang terjadi di suatu tempat. Setelah itu, dilakukan identifikasi terhadap suatu masalah. Dengan beberapa cara:

1. Memahami tema dan metode yang akan digunakan, bertujuan untuk memperoleh gambaran tentang penerapan metode yang akan digunakan dengan cara bertanya dan mencari informasi mengenai metode yang digunakan. Dengan demikian dapat membuka wawasan terhadap metode dan permasalahan yang bisa diselesaikan.
2. Menampilkan masalah yang mungkin terjadi dan mencari tempat atau studi kasus yang dapat diselesaikan dengan metode yang digunakan.
3. Merumuskan masalah dalam suatu kalimat yang dapat mendeskripsikan penelitian ini.

3.3 Analisis dan Desain

Pada tahap ini dilakukan indentifikasi dari studi kasus yang akan menerapkan arsitektur *microservices* dan *refactoring* dari arsitektur yang sudah ada pada studi kasus dengan arsitektur *microservices*. Dan dilakukan tahap-tahap pembangunan dan desain berdasarkan arsitektur *microservice*.

3.3.1 Identifikasi

Pada tahap ini menjelas kan proses bisnis yang ada pada iGracias dan apa saja yang akan menjadi penelitian tugas akhir ini.

3.3.2 Tahap *microservices*

Pada tahap ini dilakukan proses desain arsitektur *microservices* pada iGracias Universitas Telkom. Yang terdiri dari tahapan *stop digging*, *split fronted and backend* dan *extract service*.

3.3.3 *Applying*

Menerapkan Desain yang telah dibagun pada iGracias.

3.4 Pengujian

Tahap ini dilakukan pengujian menggunakan pendekatan metode ATAM terhadap implentasi dan analisis menggunakan arsitektur *microservices* pada studi kasus iGracias Universitas Telkom. Menggunakan beberapa tahapan dari pendekatan ATAM yaitu tahap prentasi metode yang akan digunakan beberapa 3 tahapan yaitu *presentation*, *investigation and analysis*, *testing* dan *Reporting*.

4 Pengujian dan Analisis

4.1 Pengujian

Pengujian dilakukan dengan metode ATAM, Untuk mencapai tujuan penelitian di atas, maka diperlukan skenario-skenario yang sekiranya dapat menunjang tujuan tersebut. Skenario pada penelitian ini dibagi menjadi empat sesuai dengan *quality attribute microservices* yaitu, skenario secara *Modularity*, *Scalability*, *Maintainability* dan *Independent*. Yang hasilnya akan diberikan kepada programmer sisfo Univesitas Telkom.

4.2 Analisis Hasil Pengujian

4.2.1 Hasil desain *microservices*

hasil desain yang didapatkan dari tahap *microservices* sebagai berikut:

a. **Stop digging**, pada Figure 4.1 ini mejelaskan hasil pada tahap *stop digging*

NO	iGracias	Fungsionalitas <i>Microservices</i>
1	Sistem PMB	Registrasi Mahasiswa Baru
		Undur Diri
		Penetapan Mahasiswa baru
		Asrama
2	Sistem Manajemen Kurikulum	Kurikulum
3	Sistem Manajemen Pembelajaran	Penetapan Kalender akademik
4	Sistem Registrasi	Penawaran Matakuliah
		Pemrosesan Rencana studi
		Pembuatan KTM
		Perubahan Rencana Studi
5	Sistem Perkuliahan dan praktikum	Penjadwalan praktikum
		Pelaksanaan praktikum

		Pemantauan praktikum
		Pelaporan praktikum
		Pengurusan Lokasi
		Bimbingan dan pengawasan
6	Sistem Penjadwalan kuliah, praktikum dan ujian	Penjadwalan Kuliah Reguler
		Pelaksanaan Kuliah Reguler
		Pemantauan Perkuliahan Reguler
		Ujian Perkuliahan Reguler
		Pelaporan Perkuliah Reguler
7	Sistem Perwalian	Perwalian
	Sistem Tesis/TA/PA	Tesis/TA/PA
		Cuti Akademik
8	Sistem Wisuda, Pembuatan Transkrip dan ijazah	Sidang kelulusan Tingkat
		Sidang Kelulusan Studi
		Drop out
		Penguduran Diri
		Pembuatan Ijazah
		Pembuatan Transkrip
		Wisuda
8	Sistem Nilai	Pemasukan Nilai
		Penghapusan Nilai
		Ekivalensi Nilai
		Perubahan Nilai
9	Sistem Pelaporan Akademik	Pelaporan Akademik
10	Sistem Manajemen Pembelajaran	IDEA
		Keuangan

Figure 4.1 Hasil Stop digging

b. Split fronted and backed

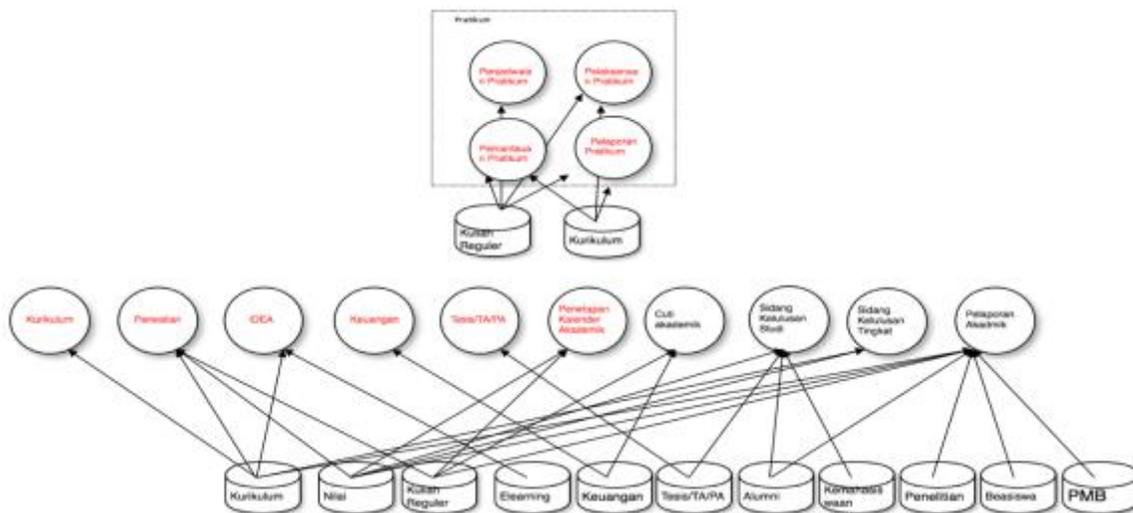


Figure 4.2 Hasil Split fronted and backed

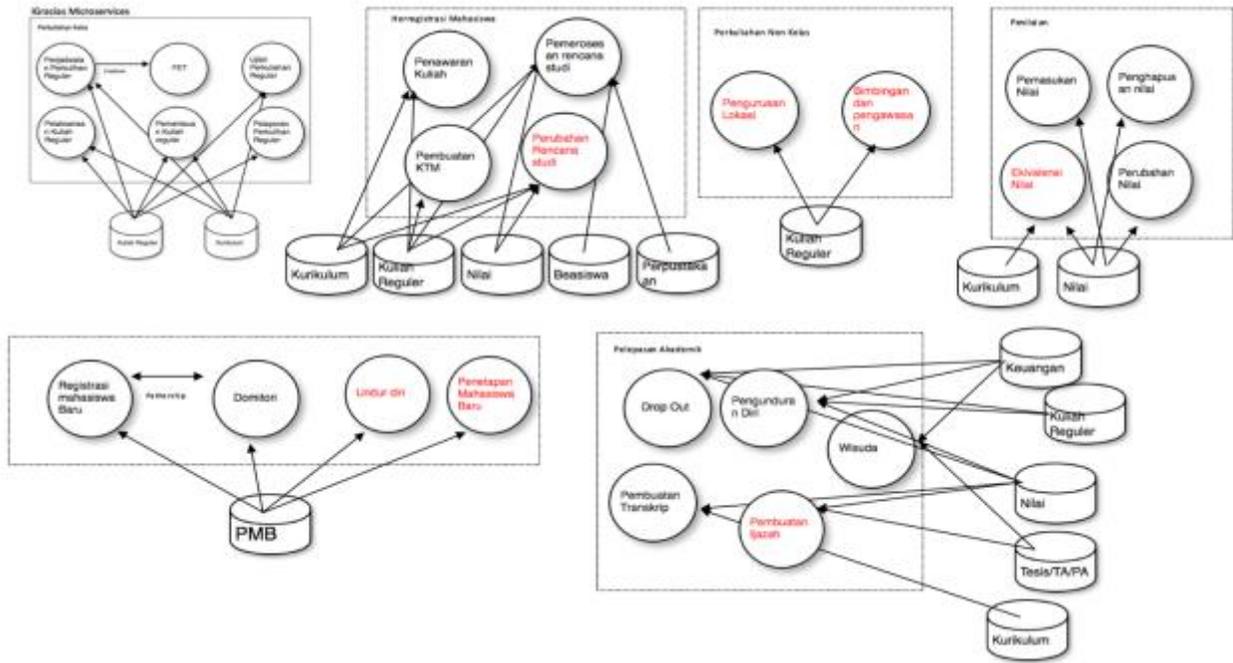


Figure 4.3 Hasil *Split fronted and backed*

c. *Extract services*

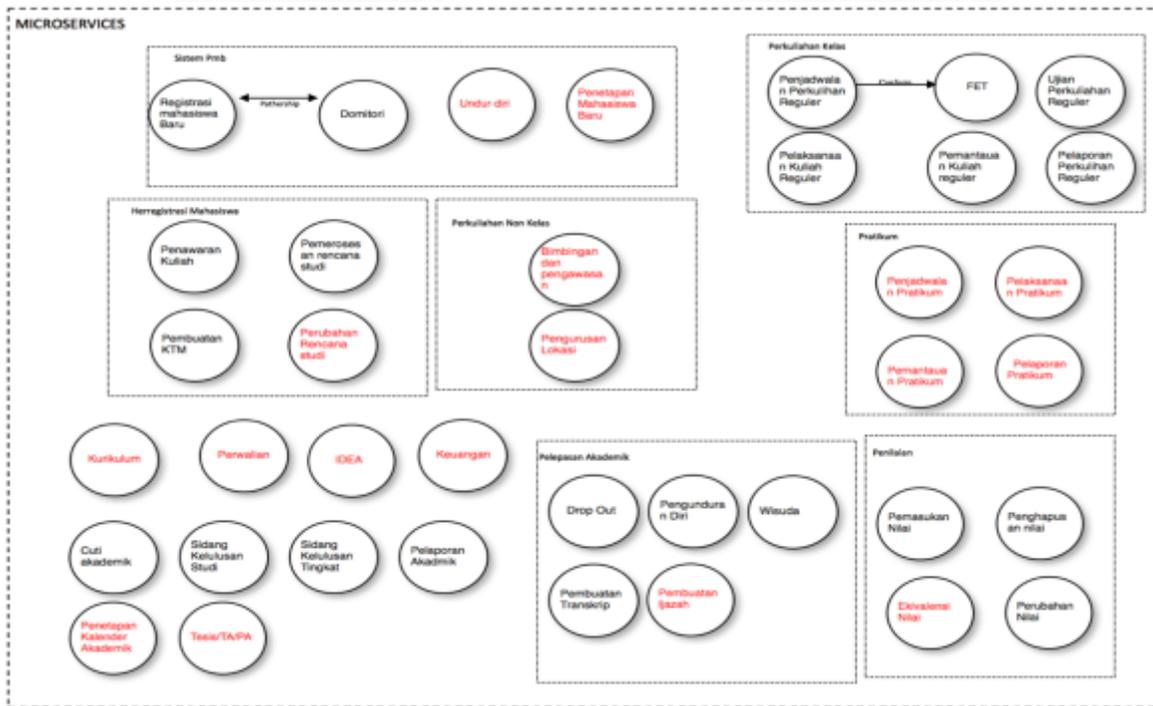


Figure 4.4 Hasil *Extract services*

4.2.2 Hasil evaluasi ATAM

Pada hasil evaluasi ATAM didapat bahwa *microservices* dapat menyelesaikan permasalahan pada iGracias. Dengan 4 quality attribute yang dihasilkan yaitu *modularity*, *independent*, *maintainability* dan *scalability*. Dengan presentase secara *modularity* 100 %, *independent* 100%, *maintainability* 100 % dan *scalability* 75 %. *Scalability* hanya 75 % dikarenakan salah satu koresponden tidak setuju dikarena menurut koresponden bahwa penerapan *microservices* kurang tepat jika dilihat dari sudut pandang *scalability*, hal ini dikarenakan harus mempertimbangkan *traffic* pada gateway dan perlu perhatian khusus jika terjadi penumpukan akses pada *gateway*.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Tujuan dari penelitian ini adalah melakukan desain konsep arsitektur *microservices* pada sistem informasi akademik iGracias Universitas Telkom, menganalisis dan mengevaluasi penggunaan arsitektur *microservices* pada iGracias menggunakan metode pendekatan ATAM. Dari hasil evaluasi menggunakan pendekatan ATAM dengan koresponden adalah programmer unit sistem informasi Universitas Telkom, dihasilkan desain arsitektur *microservices* yang dapat diterapkan pada sistem informasi akademik iGracias. *Microservices* dapat menjadikan *alternative* solusi untuk sistem informasi akademik iGracias dengan sifat *modularity*, *independent*, *scalability* dan *maintainability*. Dan dari hasil evaluasi menggunakan metode evaluasi ATAM dapat dihasilkan bahwa dari 4 koresponden menyetujui bahwa *microservices* dapat menyelesaikan permasalahan pada iGracias. Dengan presentase secara *modularity* 100 %, *independent* 100%, *maintainability* 100 % dan *scalability* 75 %. *Scalability* hanya 75 % dikarenakan salah satu koresponden tidak setuju dikarena menurut koresponden bahwa penerapan *microservices* kurang tepat jika dilihat dari sudut pandang *scalability*, hal ini dikarenakan harus mempertimbangkan *traffic* pada gateway dan perlu perhatian khusus jika terjadi penumpukan akses pada *gateway*.

5.2 Saran

Sistem yang telah dibangun memungkinkan untuk dikembangkan lebih mendalam. Dalam hal ini peneliti menyarankan yaitu perlunya penelitian pembangunan *framework gateway* untuk mengatur akses *traffic microservices* dan juga proses migrasi dari iGracias ke *microservices*.

6. Daftar Pustaka

- [1] M. d. J. L. Folwer, "Microservices," 2 11 2016. [Online]. Available: <https://martinfoowler.com/microservices/>. [Accessed 2 11 2016].
- [2] G. Grunman and A. Marrison, "Microservices :The Resurgence of SOA Principles and an Alternative to the Monolithic," 2014. [Online]. Available: <http://www.pwc.com/us/en/technology-forecast/2014/cloud-computing/features/microservices.html>. [Accessed 2 october 2016].
- [3] C. Richardson and F. Smith, *Microservices From Design to Deployment*, NGINX, 2016.
- [4] R. Kazman, M. Klein and P. Clements, *ATAM: Method for Architecture Evaluation*, Pittsburgh: Carnegie Mellon University, 2000.
- [5] G. D. Garson, *Partial Least Square: Regression & Structural Equation Models*, Blue Book Series, Asheboro: Statistical Publishing Associates, 2016.
- [6] D. S. T. University, *User Manual Distance Education (IDEA) - Student*, Bandung: Universitas Telkom, 2014.
- [7] A. f. P. Management, "Introduction to Gamification," APM, Princes Risborough, 2014.

- [8] R. Shogi, "Evaluasi E-Learning Idea pada S1 Teknik Informatika Telkom University Menggunakan Technology Acceptance Model (TAM)," Universitas Telkom, Bandung, 2015.
- [9] I. Glover, "Play as Your Learn: Gamification as a Technique for Motivating Learners," Sheffield Hallam University, Sheffield, 2013.
- [10] Y. Firdaus, "Rencana Strategis Sistem Infomasi Institut Teknologi Telkom 2011-2014," Bandung , IT TELKOM, 2011.

