

Pembangunan Sistem Informasi Layanan Kemahasiswaan Berbasis Web Menggunakan *Service-Oriented Architecture*

Studi Kasus : Bagian Administrasi Akademik dan Kemahasiswaan Politeknik Statistika STIS

(*Web-Based Student Service Information System Development Using Service-Oriented Architecture*)

Fikri Septrian Anggara¹, Lutfi Rahmatuti Maghfiroh¹

¹Politeknik Statistika STIS

E-mail: fikriseptriananggara@gmail.com, lutfirm@stis.ac.id

ABSTRAK

Bagian Administrasi Akademik dan Kemahasiswaan (BAAK) merupakan salah satu unsur pelaksana administrasi Politeknik Statistika STIS dan memiliki tanggung jawab salah satunya layanan kemahasiswaan. Proses pelayanan di bagian kemahasiswaan masih dilakukan secara manual contohnya pada layanan Surat Permintaan Data (SPD) dan Surat Keterangan Mahasiswa (SKM). Permintaan layanan dilakukan melalui email atau langsung di ruang BAAK, proses pengulasan permintaan layanan tersebut belum terkomputerisasi sehingga rentan kesalahan dan tidak efisien serta progres persetujuan tidak termonitor oleh pengguna. Penelitian ini bertujuan untuk merancang dan membangun sistem informasi layanan kemahasiswaan berbasis web progresif dengan harapan mampu memenuhi kebutuhan proses bisnis kemahasiswaan BAAK. Pengembangan sistem informasi ini dilakukan menggunakan metode *eXtreme Programming* modifikasi dan RESTful MSOAM. Berdasarkan hasil evaluasi, diketahui bahwa sistem telah mengimplementasikan proses bisnis usulan layanan SPD dan SKM, telah memenuhi 90,77% kriteria penerimaan kebutuhan pengguna, memenuhi syarat PWA, dan memenuhi 4 prinsip inti *service-orientation* dan prinsip *reusability*.

Kata kunci: Sistem Informasi, BAAK, *eXtreme Programming*, *Progressive Web Apps*, RESTful MSOAM, *Service-Oriented Architecture*

ABSTRACT

The Academic and Student Administration Section (BAAK) is one of the implementing elements of the STIS Statistics Polytechnic administration and has responsibilities, one of which is student services. The service process in the student affairs department is still done manually, for example in the Data Request Letter (SPD) and Student Certificate (SKM) services. Service requests are made via email or directly in the BAAK room, the service request review process has not been computerized so that it is prone to errors and is not efficient and the approval progress is not monitored by the user. This study aims to design and build a progressive web-based student service information system with the hope of being able to meet the needs of BAAK's student business processes. The development of this information system is carried out using the modified *eXtreme Programming* method and RESTful MSOAM. Based on the evaluation results, it is known that the system has implemented the proposed business processes for SPD and SKM services, has met 90.77% of the acceptance criteria for user requirements, has met PWA requirements, and has met the 4 core principles of service-orientation and the principle of reusability.

Keywords: Information Systems, BAAK, *eXtreme Programming*, *Progressive Web Apps*, RESTful MSOAM, *Service-Oriented Architecture* 10

PENDAHULUAN

Dalam birokrasi Politeknik Statistika STIS, Bagian Administrasi Akademik, dan Kemahasiswaan (BAAK) berperan dalam melaksanakan pelayanan teknis dan administrasi bidang akademik dan kemahasiswaan di lingkungan Politeknik Statistika STIS (STIS, 2020). Berdasarkan hasil wawancara dengan staff dan kepala Subbagian Kemahasiswaan Bagian Administrasi Akademik dan Kemahasiswaan (BAAK) serta subkoordinator fungsi Administrasi Akademik dan Kerjasama Politeknik Statistika STIS, bidang kemahasiswaan berperan dalam layanan kemahasiswaan seperti Surat Permintaan Data (SPD) dan Surat Keterangan Mahasiswa (SKM). Saat ini proses pelayanan kemahasiswaan tersebut dilakukan secara manual sehingga rentan terhadap kesalahan seperti salah mevalidasi data mahasiswa serta tidak efisien. Proses pelayanan tidak transparan sehingga mahasiswa tidak bisa memantau progres permintaan pelayanan. Proses persetujuan masih dilakukan menggunakan surat fisik atau elektronik dan dikirimkan menggunakan email secara manual.

Aplikasi sistem informasi dapat mengatasi masalah ini dengan mengotomatisasi tugas, mengurangi redundansi dan menyediakan proses yang transparan (Whitten & Bentley, 2007). Sebuah aplikasi sistem informasi dapat diimplementasikan ke berbagai platform. Platform aplikasi dapat berbasis web, desktop, atau seluler/asli. Setiap platform memiliki kelebihan dan kekurangannya masing-masing. Terdapat inovasi pada aplikasi berbasis web yaitu *Progressive Web Apps* yang mampu memberikan fungsionalitas aplikasi asli yang sebelumnya tidak dapat diimplementasikan dalam aplikasi web seperti akses saat tidak ada internet, *push notification*, aplikasi yang dapat diinstal, dll (Richard & LePage, What are Progressive Web Apps, 2020). PWA menggabungkan fitur-fitur aplikasi berbasis web dan aplikasi *native* android tanpa kehilangan fungsionalitas aplikasi berbasis web, sehingga aplikasi ini menjadi lebih progresif dan memungkinkan pengembang web untuk membangun aplikasi yang ramah pengguna *mobilephone*.

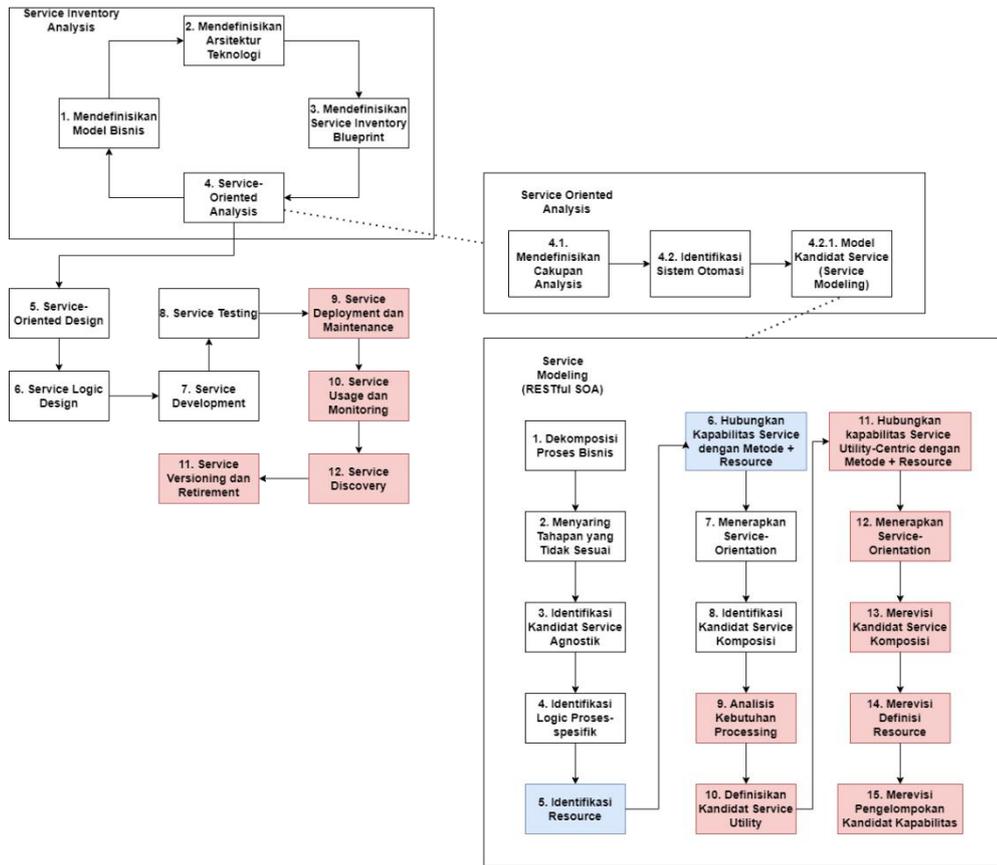
Sistem yang dibangun diharapkan dapat digunakan kembali dan dapat dioperasikan dengan sistem lain. Hal ini dapat diwujudkan dengan menggunakan arsitektur sistem yang mendukung *reusability* dan *interoperability*. *Service-Oriented Architecture* adalah gaya arsitektur sistem terdistribusi yang memiliki keunggulan dalam *reusability* dan interoperabilitas sistem. *Reusability* dan *interoperability* menghilangkan aktivitas yang harus dilakukan oleh pengembang lain yang membutuhkan fungsionalitas yang sama dengan sistem yang ada, sehingga jika ada sistem lain yang membutuhkan fungsionalitas yang sama dalam layanannya, pengembangan sistem dapat dilakukan lebih cepat (IBM, 2021) (OMG, 2012).

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem informasi layanan mahasiswa yang diimplementasikan sebagai *Progressive Web-Apps* (PWAs) dengan *Service-Oriented Architecture* (SOA) untuk memenuhi kebutuhan kemahasiswaan BAAK.

METODE

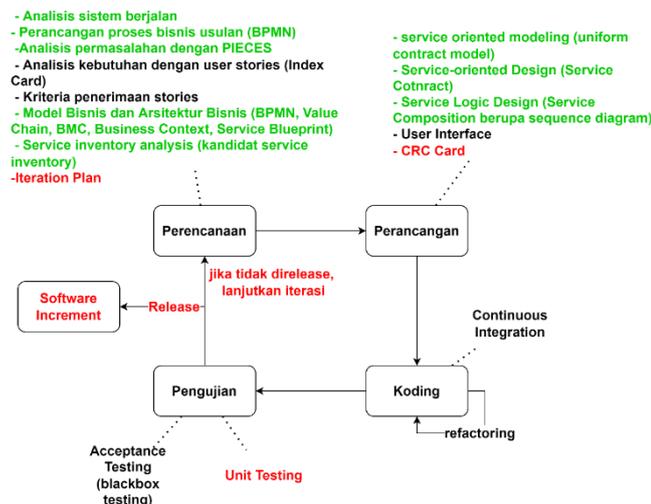
Penelitian dilakukan dengan menggunakan metode *Design Science Research* (DSR). DSR merupakan metode penelitian yang menjembatani kontribusi penelitian dari segi praktis dan pengetahuan (Brocke & Hevner, 2020). Metode penelitian DSR menghasilkan 2 hal yaitu artefak dan pengetahuan desain. Dalam penelitian ini, DSR dimulai dari mengidentifikasi masalah yang berasal dari hasil wawancara materi pelajaran yang terkait dengan layanan kemahasiswaan SPD dan SKM, studi literatur dan observasi oleh peneliti dan kami mengelompokkan masalah menggunakan kerangka PIECES (Whitten & Bentley, 2007). Dari permasalahan tersebut, tujuan penelitian adalah membangun sebuah aplikasi sistem informasi layanan kemahasiswaan SPD dan SKM di BAAK. Untuk mengembangkan sistem, kami menggunakan metode *eXtreme Programming* (XP) (Pressman, 2010) dan *RESTful MSOAM* (Whitten & Bentley, 2007) (Erl, *Service-Oriented Architecture: Concepts, Technology and Design*, 2009) (Erl, Carlyle, Pautasso, & Balasubramanian, 2012).

XP digunakan sebagai metode pengembangan sistem karena proyek ini sesuai dengan rekomendasi Kent Beck, yaitu XP cocok digunakan untuk tim kecil, komunikasi antara pengembang dan pengguna, dan kebutuhan sistem yang bisa berubah (Beck & Andres, 2004). Terdapat berbagai metode implementasi SOA seperti SOMA dan MSOAM. SOA terdiri dari komponen *unit logic* yang dikenal sebagai *service* yang bertugas menangani suatu tugas tertentu (Erl, Carlyle, Pautasso, & Balasubramanian, 2012). *Service* dibangun menjadi *web service*. *Web service* bisa dibangun menggunakan *RESTful web service* atau *SOAP web service*. Pada penelitian ini, *Service* pada solusi SOA diimplementasikan menjadi *RESTful web service* karena pembangunannya relatif lebih mudah dan cepat dari pada *SOAP web service*. *RESTful MSOAM* merupakan metode MSOAM yang dikhususkan untuk pengembangan solusi SOA menggunakan *RESTful web service*. Karena metode implementasi yang cocok dengan kebutuhan implementasi *service*, kami memilih metode tersebut.



Gambar 1 Tahapan RESTful MSOAM

Terdapat 15 tahapan *service modeling* pada RESTful MSOAM seperti terlihat pada gambar 1 (bagian berwarna merah tidak dilakukan pada penelitian, bagian berwarna biru merupakan penambahan khusus MSOAM untuk mengadopsi SOA menggunakan RESTful *web service*). Pemodelan layanan pada tahap RESTful MSOAM hanya sampai pada tahap penentuan calon *service composition* (tahap 9 pada *service modeling*) karena tahap selanjutnya bersifat opsional (Erl, Carlyle, Pautasso, & Balasubramanian, 2012). Tahap RESTful MSOAM hanya sampai pada pengujian *service* karena cakupan penelitian hanya sampai pembangunan sistem tidak sampai deployment dan seterusnya.



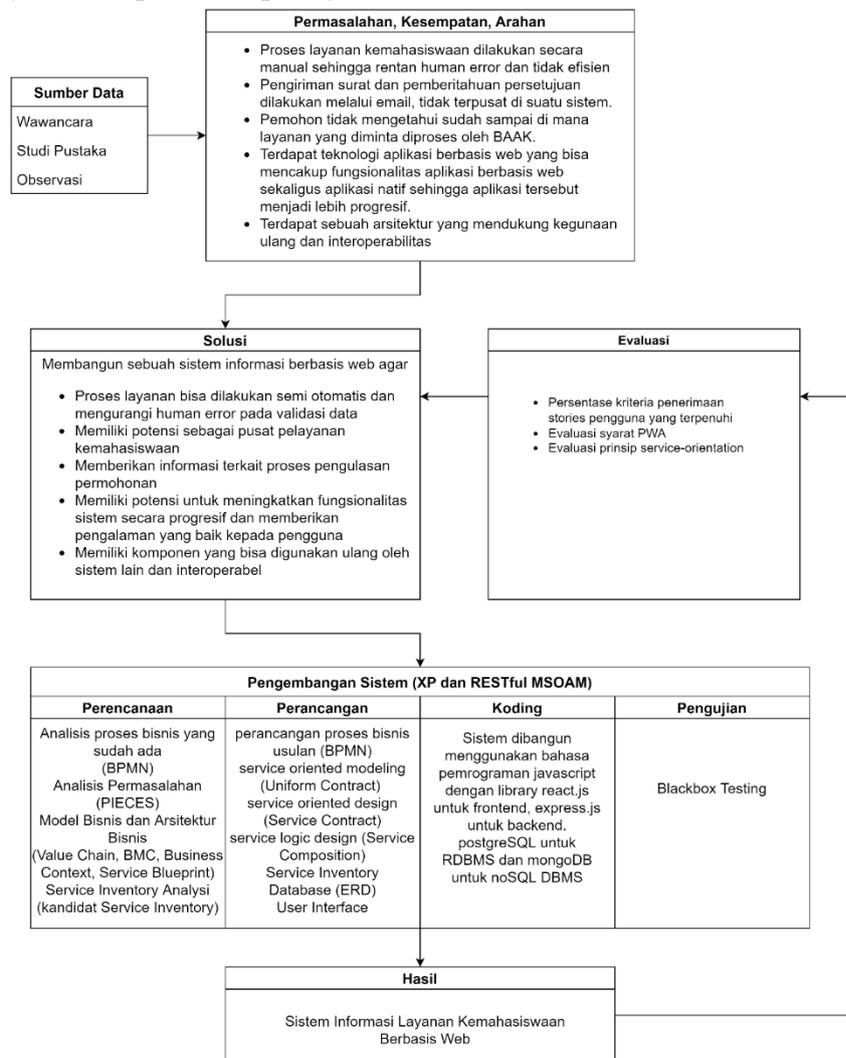
Gambar 2 Tahapan XP hasil modifikasi

Untuk menggabungkan XP dan *RESTful* MSOAM, kami melakukan beberapa modifikasi tahapannya seperti terlihat pada gambar 2 (bagian berwarna hijau merupakan penambahan proses, bagian berwarna merah merupakan proses yang dihilangkan). Kami tidak menggunakan kartu CRC dan tidak menerapkan *unit testing* karena ruang lingkup kami hanya sampai pengembangan layanan. Kami menambahkan beberapa subfase dalam tahap perencanaan XP yaitu analisis sistem yang ada, mendefinisikan ontologi model bisnis, *user stories* dan kriterianya, dan *service inventory analysis*. Pada fase XP berikutnya yaitu fase Desain, kami menambahkan pemodelan

berorientasi layanan subfase, *service oriented design*, *service logic design*, dan perancangan *user interface*. Kami menggunakan *blackbox testing* untuk menguji fungsionalitas aplikasi kami. Skenario *blackbox testing* didefinisikan dari proses bisnis usulan dan *user stories*.

Salah satu tahapan perancangan arsitektur menggunakan *RESTful* MSOAM adalah mendefinisikan ontologi model bisnis. Untuk mendefinisikan ontologi bisnis, kami menggunakan *Business Model Canvas* (BMC), *Value Chain Diagram*, *Service Blueprint*, *Entity Model*, dan *Business Process Modeling Notation* (BPMN) (Rosen, Lublinsky, T.Smith, & Balcer, 2008) (Wijayanto & Suhardi, 2014).

Evaluasi penelitian dilakukan dengan menghitung persentase kriteria penerimaan pengguna yang berhasil dipenuhi sistem, menguji pemenuhan checklist PWA (Richard, web.dev, 2020), dan mengevaluasi empat prinsip inti orientasi layanan ditambah prinsip *reusability* (Erl, Carlyle, Pautasso, & Balasubramanian, 2012). Alur penelitian secara menyeluruh dapat dilihat pada gambar 3.



Gambar 3 Alur penelitian

HASIL DAN PEMBAHASAN

Perencanaan

Analisis Sistem Berjalan

Pada tahap ini diketahui proses bisnis layanan SPD dan SKM pada system berjalan, aktor yang terlibat dan data yang digunakan. Aktor yang terlibat selama proses layanan SPD dan SKM yaitu mahasiswa, staff BAAK, dosen pembimbing, penyetuju (Kepala BAAK, Wakil Direktur 1). Data yang digunakan selama proses ini yaitu data mahasiswa, form pengajuan serta SPD dan SKM.

Rancangan Proses Bisnis Usulan

Rancangan proses bisnis dilakukan BAAK. BAAK menentukan proses bisnis layanan SPD dan SKM yang baru kemudian peneliti menggabungkan proses SPD dan SKM ke dalam satu proses.

Analisis Permasalahan

Analisis permasalahan dilakukan menggunakan PIECES . Diketahui permasalahan dan solusi permasalahan seperti pada table di bawah.

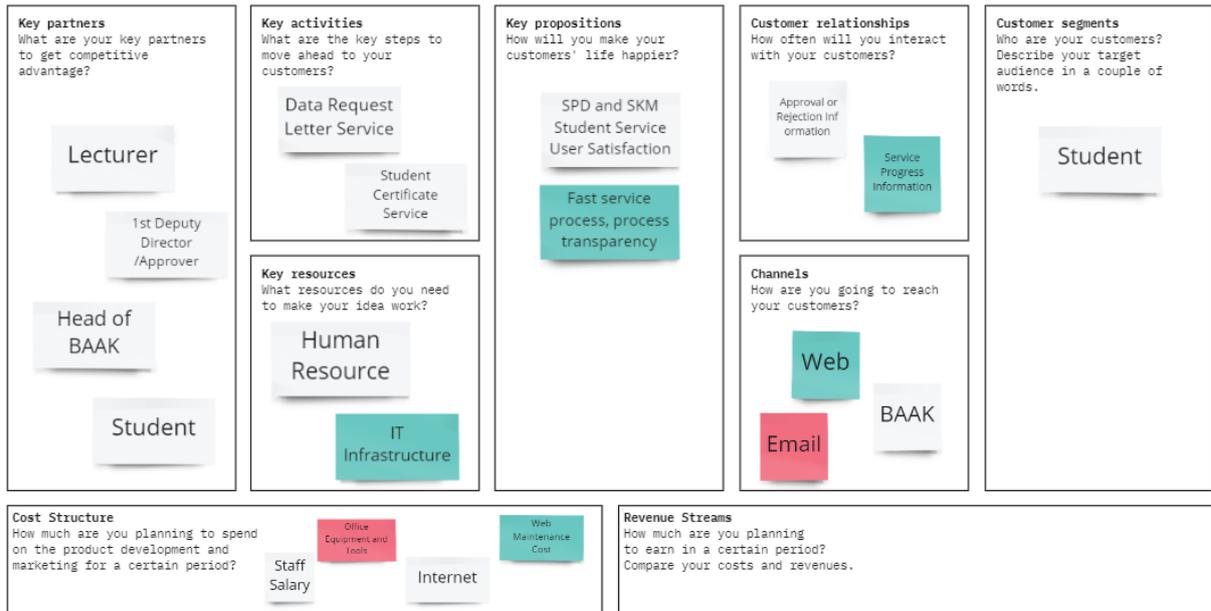
Tabel 1 Permasalahan pada Sistem Berjalan dan Solusi Sistem Usulan.

Kategori PIECES	Sistem Berjalan	Sistem Usulan
Performa	Usaha yang diperlukan untuk mengerjakan tugas bisa diringankan,	Membangun system terkomputerisasi yang bisa mengotomasi proses
Informasi	Penyimpanan data redundan, proses pengajuan tidak diketahui, data hardcopy tidak aman dari kecelakaan, nama atau nim mahasiswa salah, sehingga terjadi kesalahan validasi	Menyimpan data di satu tempat dan dalam bentuk softfile. Otomasi input oleh sistem. Memberikan notifikasi secara berkala
Ekonomi	Minimalisir penggunaan kertas, tinta pena, tinta stempel	Menggunakan <i>qr code</i> sederhana sebagai pengganti tanda tangan dan surat elektronik.
Kontrol	Error saat proses validasi data mahasiswa, Langkah yang rumit, penyimpanan data yang redundan, proses pengajuan layanan tidak diketahui.	Validasi oleh sistem sesuai dengan aturan bisnsi, menggabungkan langkah langkah apabila dimungkinkan
Efisiensi	Usaha validasi data bisa diminalisir, penyimpanan data redundan.	Validasi data mahasiswa oleh sistem, penyimpanan terpusat
Layanan	Mahasiswa bingung bagaimana menggunakan layanan	Sistem memberikan informasi langkah langkah penggunaan layanan

Mendefinisikan Ontologi Bisnis

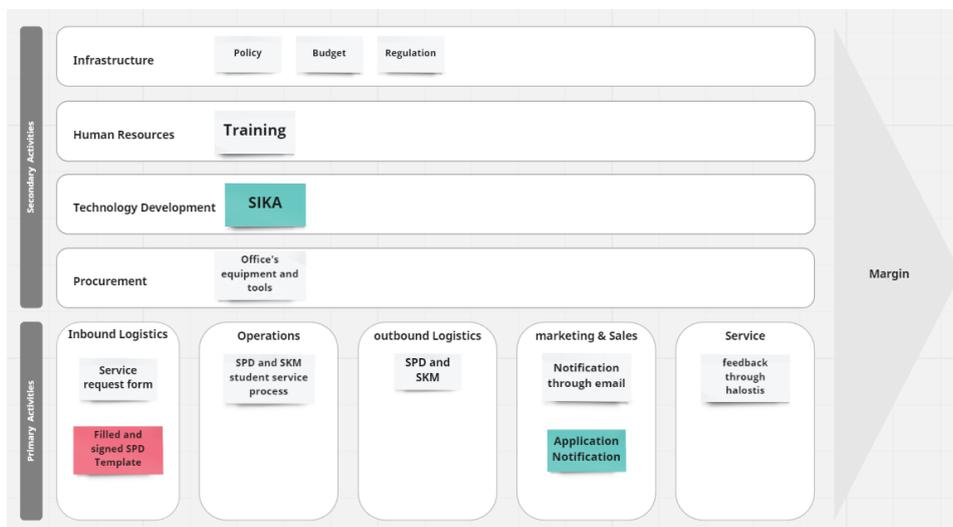
Dalam mendefinisikan ontologi bisnis, digunakan *Business Model Canvas (BMC)*, *Service Blueprint*, *Entity Model*, dan *Business Process Modeling Notation (BPMN)*. Berdasarkan *BMC* diketahui bahwa hubungan antara pelanggan dan system hanya terjadi pada saat pengguna mengirimkan form dan pemberitahuan informasi persetujuan serta belum terdapat sumber daya IT pada layanan SPD dan SKM sekarnag. Sehingga diusulkan perubahan model bisnis yaitu dengan menambah sumber daya IT untuk mengurangi biaya peralatan dan perlengkapan dan notifikasi aplikasi untuk meningkatkan transparansi progress. Namun infrastruktur IT menambah biaya dari sisi perawatan. Dengan adanya system terkomputerisasi melalui web, pengalaman pengguna menjadi meningkat karena proses lebih cepat. BMC usulan dapat terlihat pada gambar 1.

Proposed BAAK's Business Model (SPD and SKM Service)



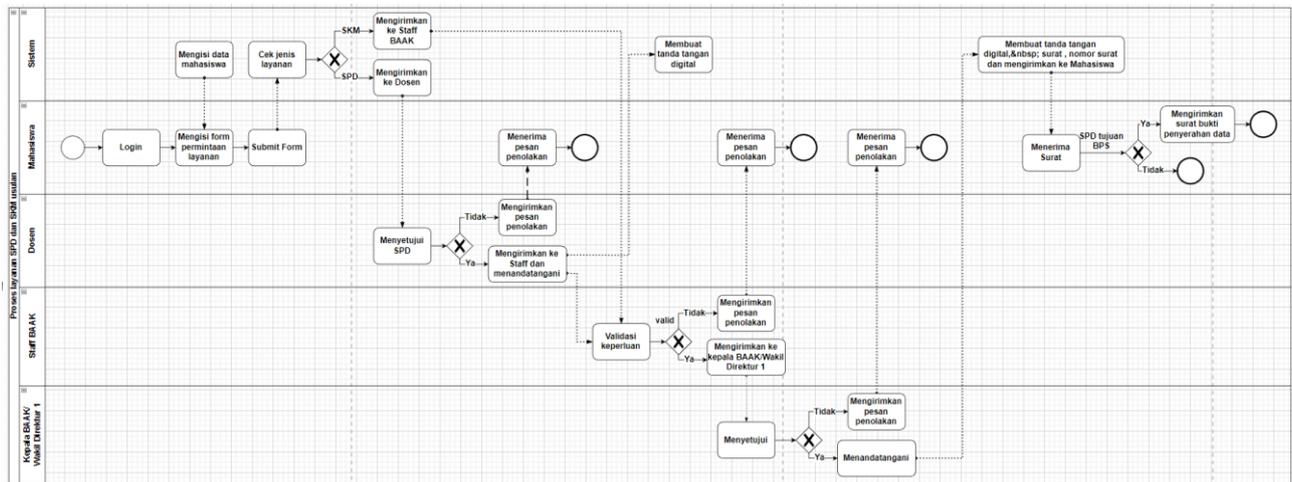
Gambar 4 BMC usulan.

Berdasarkan *value chain diagram*, diketahui bahwa belum terdapat pengembangan teknologi, serta terdapat dua inbound logistic yaitu form SKM dan template SPD. Dari sisi *marketing* dan *sales*, saat ini pemberitahuan masih dilakukan melalui email. Kemudian diusulkan adanya pengembangan teknologi dan inbound logistic digabung menjadi satu form serta menambah komponen *marketing* yaitu notifikasi aplikasi. Perubahan *value chain* dapat terlihat pada gambar 2



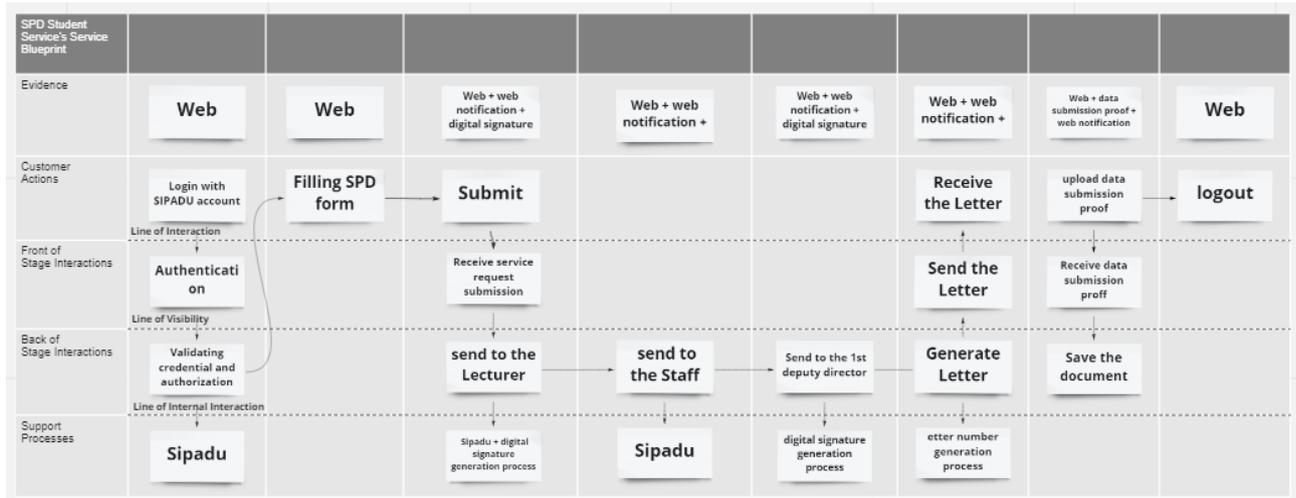
Gambar 5 Perubahan Value Chain.

Berdasarkan BPMN, diketahui proses bisnis yang terdapat pada layanan SPD dan SKM sistem berjalan, kemudian peneliti menggabungkan proses bisnis SPD dan SKM menjadi satu yang dapat terlihat pada gambar



Gambar 6 Proses Bisnis Usulan.

Berdasarkan *Service Blueprint*, diketahui informasi secara menyeluruh mengenai layanan, resource, proses yang terlihat dan tidak terlihat oleh pengguna. Berdasarkan *service blueprint system* berjalan, diketahui bahwa interaksi antara pengguna dan BAAK hanya terjadi di awal proses yaitu saat pengajuan form dan pengiriman surat. Diusulkan perubahan *service blueprint* yaitu adanya aplikasi web. Aplikasi ini meningkatkan interaksi antara pengguna dan system melalui notifikasi. Pada *service blueprint* usulan juga terdapat otomasi proses, yaitu penggunaan tanda tangan digital dan otomasi pembuatan surat sehingga proses bisa menjadi lebih cepat. *Service blueprint* usulan dapat terlihat pada gambar 4.



Gambar 7 Service Blueprin Usulan.

Berdasarkan *Service Blueprint* dan BPMN, diketahui entitas dan hubungan entitas bisnis pada layanan SPD dan SKM. Entitas dan hubungan antar entitas dapat terlihat pada tabel.

Tabel 2 Entitas Bisnis.

Entitas	Deskripsi
Mahasiswa	Entitas yang memulai proses pengajuan layanan SPD dan SKM.
Pegawai/staff	Entitas yang meverifikasi pengajuan layanan dari mahasiswa.
Penyetuju	Entitas yang menandatangani pengajuan layanan dari pegawai dan mahasiswa. Penyetuju meliputi dosen pembimbing, kepala BAAK, dan wakil direktur 1.
Pengajuan Layanan SPD dan SKM	Entitas yang menyimpan data pengajuan dan status pengajuan.
Surat	Entitas yang menyimpan data SPD dan SKM. Entitas yang generate surat berdasarkan input data.

Kebutuhan Bisnis

Kebutuhan bisnis berasal dari proses bisnis usulan, *service blueprint* serta *user stories* dan didokumentasikan menggunakan *use case* diagram. *Use case* kebutuhan otomasi bisnis mendefinisikan solusi yang akan diimplementasikan oleh sistem. *Use case* yang digunakan tidak berfokus pada teknologi, melainkan berfokus pada kebutuhan bisnis. Kebutuhan bisnis sistem yang dibangun terlihat pada tabel 3.

Tabel 3 Kebutuhan bisnis

<i>Nama use case</i>	<i>Deskripsi</i>	<i>Aktor yang terlibat</i>
Mengajukan keperluan layanan SPD dan SKM	Use case ini mendeskripsikan event mahasiswa yang ingin menggunakan layanan SPD atau SKM. Data diinput melalui sistem dan data mahasiswa diinput secara otomatis dari sistem.	Mahasiswa
Login	Use case ini mendeskripsikan event pengguna ketika login	Mahasiswa, BAAK, Dosen, Penyetuju
Mengupload bukti penyerahan data	Use case ini mendeskripsikan event mahasiswa yang mengupload surat ke sistem, berlaku bagi mahasiswa yang menggunakan layanan SPD dengan tujuan BPS.	Mahasiswa
Verifikasi keperluan	Use case ini mendeskripsikan event staff BAAK yang meverifikasi pengajuan keperluan.	BAAK
Download surat	Use case ini mendeskripsikan event seluruh aktor yang ingin mendownload surat layanan SPD SKM. Use case ini bergantung pada use case generate surat dan baru dijalankan apabila surat sudah digenerate	Mahasiswa, BAAK, Dosen, Penyetuju
Menandatangani keperluan	Use case ini mendeskripsikan event penyetuju dan dosen menyetujui pengajuan layanan SPD SKM dan menandatangani persetujuan, tanda tangan digenerate oleh sistem begitu penyetuju dan dosen menyetujui dan akan metriger use case generate surat.	Penyetuju, Dosen

Menentukan Teknologi Arsitektur

Arsitektur teknologi yang digunakan ialah REST. Dalam mendefinisikan arsitektur teknologi, dilakukan pemodelan *Uniform Contract*. *Uniform contract* berguna sebagai standarisasi *service contract* service individu. Pemodelan uniform kontrak menghasilkan tiga hal, yaitu standar *resource identifier*, metode HTTP yang digunakan, *media-type* yang digunakan, struktur *request* dan struktur *response*.

Menentukan Cakupan Analisis (*service-oriented analysis*)

Cakupan analisis sudah dilakukan pada tahap *service inventory analysis* yaitu layanan SPD dan SKM kemahasiswaan BAAK.

Identifikasi sistem otomasi (*service-oriented analysis*)

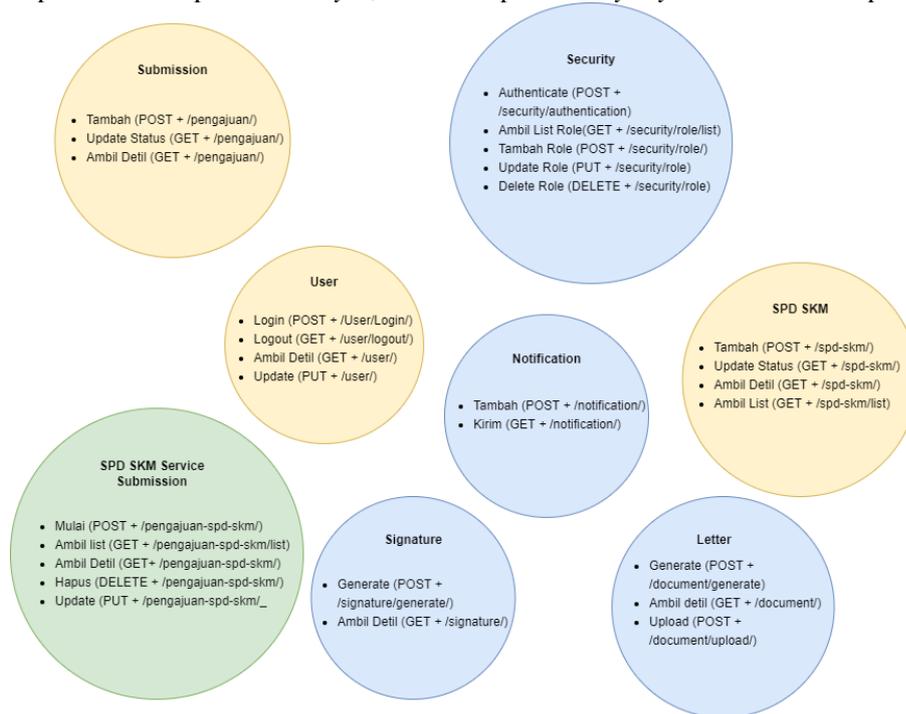
Keseluruhan proses bisnis pada system berjalan dilakukan secara manual. Berdasarkan *service blueprint* dan proses bisnis usulan, diketahui bahwa otomasi bisa dilakukan pada pembuatan surat, penandatanganan keperluan dan validasi data.

Perancangan

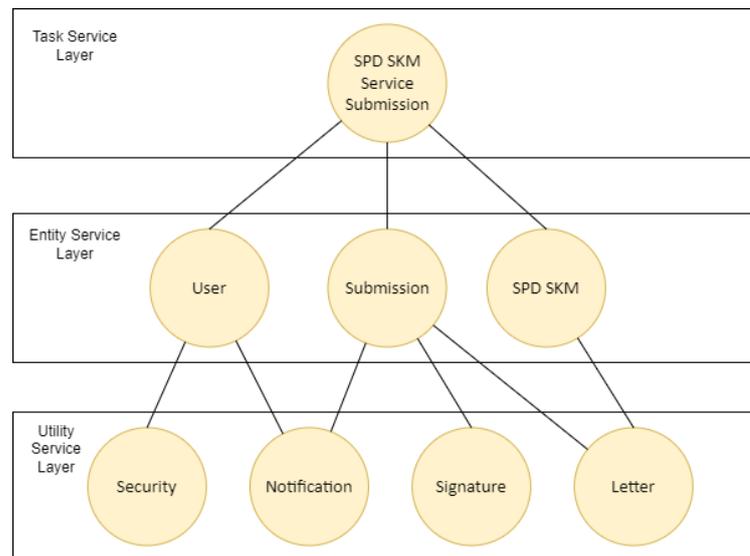
Service-oriented Modeling (service-oriented analysis)

Pada tahap ini dilakukan dekomposisi proses bisnis usulan, mengidentifikasi aksi yang bisa diotomasi, menghilangkan aksi yang tidak bisa diotomasi, mengidentifikasi aksi *agnostic*, menghilangkan aksi *non-agnostic*, menentukan *resource* dan entitas *service*, serta utilitas *service*, serta menerapkan *service-orientation*. Hasil dari tahapan ini ialah kandidat *service* dan kapabilitas *service* dan kandidat *service composition*. Gambar 7 menunjukkan kandidat kapabilitas masing-masing *service*. Gambar 8 menunjukkan kandidat *service composition*.

Service composition menunjukkan bagaimana *service* berinteraksi satu sama lain. Berdasarkan gambar 9, diketahui bahwa terdapat 1 *service* pada *task layer*, 3 *service* pada *entity layer* dan 4 *service* pada *utility layer*.



Gambar 8 Kandidat dan kapabilitas *service*



Gambar 9 Kandidat *Service Composition*.

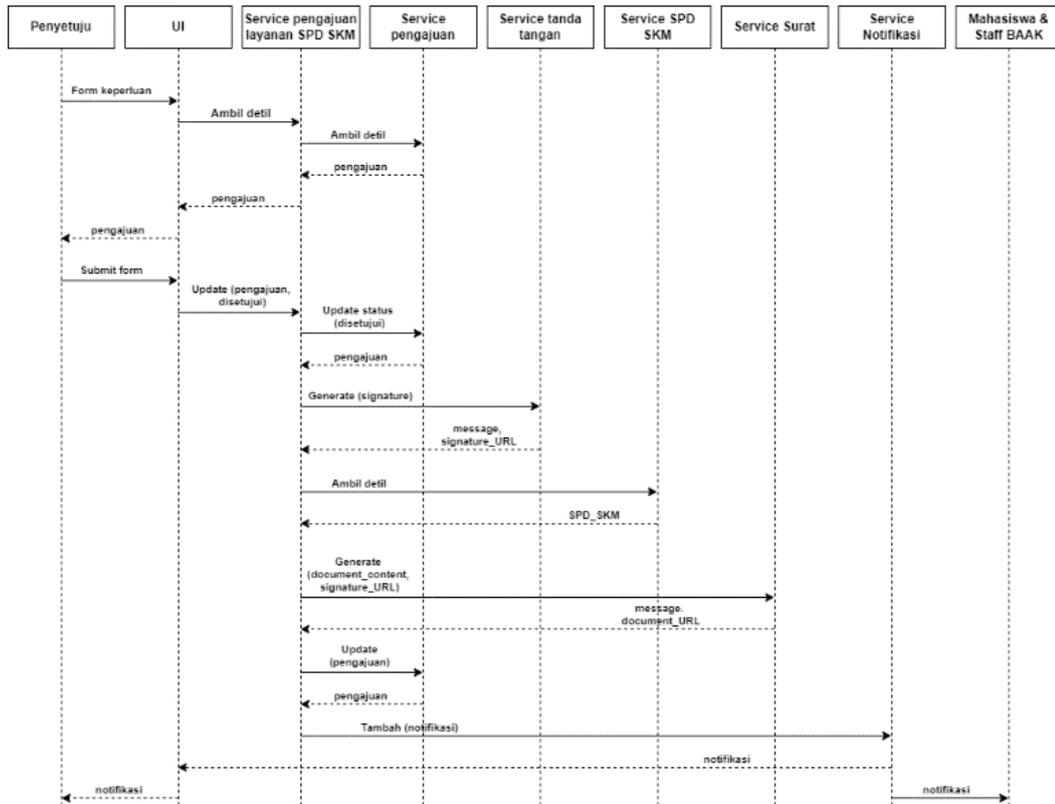
Service-Oriented Design

Pada tahap ini dilakukan rancangan *service contract* individu *service* yang mengacu pada *uniform contract* pada tahap *service inventory analysis*. *Service contract* individu mencakup metode HTTP yang digunakan untuk mengakses kapabilitas, struktur *request* dan syarat *body request*, *media-type* dan struktur *response*.

Service Logic Design

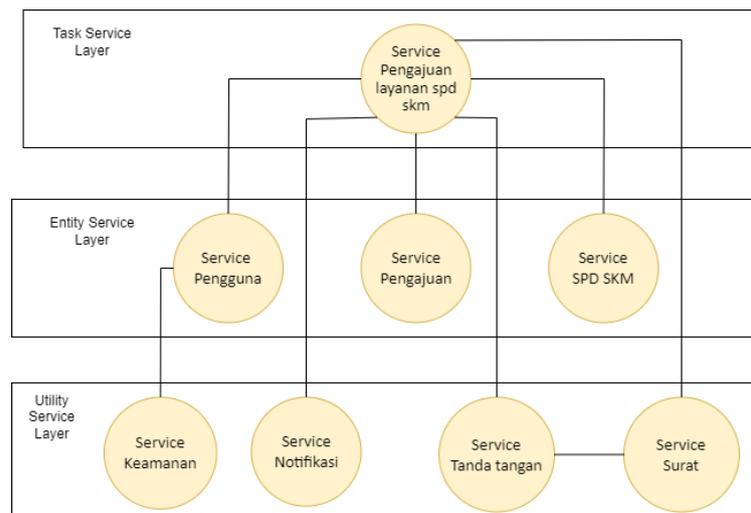
Pada tahap ini dihasilkan solusi dari kebutuhan otomasi bisnis yang ada pada *business requirement use case*. Seluruh kebutuhan otomasi bisnis memerlukan *service composition*. Interaksi *service* dalam menyelesaikan otomasi kebutuhan bisnis didokumentasikan menggunakan *sequence diagram*. Contoh interaksi *service* dapat dilihat pada gambar 10 untuk kasus penyetujuan, pembuatan tanda tangan digital dan pembuatan surat. Untuk

kasus tersebut terlihat interaksi antara *service* pengajuan layanan spd skm, pengajuan, tanda tangan, SPD SKM, surat, dan *service* notifikasi.



Gambar 10 *Service Interaction Diagram* untuk task penyetujuan, pembuatan tanda tangan dan surat digital.

Setelah *service logic design* telah dilakukan, dilakukan penyempurnaan terhadap *service composition*. *Service composition* final terlihat pada gambar 11. Terdapat perbedaan komposisi *service* yang sudah final dan pada kandidat komposisi (gambar 9), yaitu kurangnya dependensi antar *service utility* dan *entity service*. Sebelumnya *service* pengguna memiliki interaksi dengan *service* notifikasi, pada versi final, *service* notifikasi hanya berinteraksi dengan task *service*. *Service* tanda tangan tidak berkomunikasi dengan *service* pengajuan, namun terdapat tambahan interaksi dengan *service* surat.

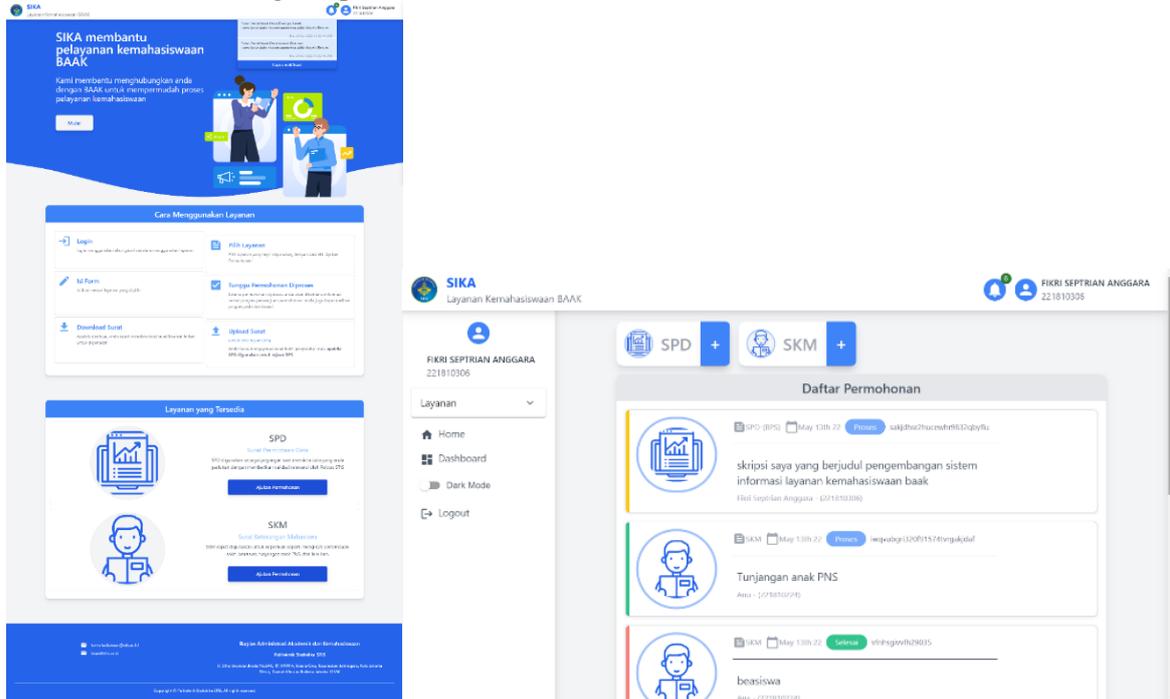


Gambar 11 *service composition* final

Koding

User Interface

Dari sisi tampilan, dihasilkan 3 halaman yaitu halaman Home, dashboard, dan detail permohonan. Halaman Home dan dashboard terlihat pada gambar 13 dan 14



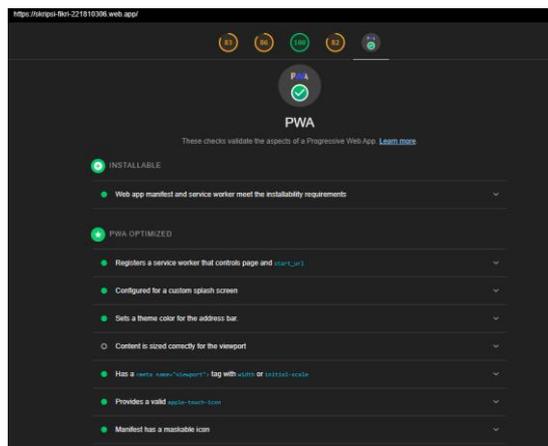
Gambar 13 Antar Muka Halaman Home (kiri); Gambar 14 Halaman Dashboard (kanan)

Pengujian

Pengujian dilakukan dengan *blackbox testing* terhadap *use case* kebutuhan bisnis (tabel 3). Semua *use case* sudah dapat terpenuhi oleh sistem.

Evaluasi

Evaluasi dilakukan terhadap persentase kriteria penerimaan *user stories*, pemenuhan kriteria PWA, dan evaluasi prinsip inti SOA. Persentase kriteria penerimaan yang berhasil dipenuhi oleh system sebesar 90,77% (59 dari 65 kriteria). Persentase stories yang berhasil dipenuhi oleh system adalah 75% (12 dari 16 *stories*). Berdasarkan evaluasi pemenuhan syarat PWA, system telah berhasil memenuhi syarat sebagai PWA. Pemenuhan kriteria PWA dapat dilihat pada gambar 16.



Gambar 16 Pemenuhan Kriteria PWA

Tabel 4 Evaluasi syarat implementasi dan karakteristik desain prinsip inti SOA

Prinsip Service-orientation	Syarat Implementasi dan Karakteristik Desain	Evaluasi syarat implementasi dan karakteristik desain
<i>Autonomous</i>	<ul style="list-style-type: none"> • <i>Service</i> memiliki kontrak yang dirancang dengan batasan yang jelas yang tidak tumpang tindih dengan <i>service</i> lain. • Lingkungan deployment yang terdistribusi sehingga <i>service</i> bisa dipindahkan, atau dikomposisikan sesuai kebutuhan. 	<ul style="list-style-type: none"> • <i>Service contract</i> dirancang berdasarkan dekomposisi proses bisnis menjadi satu aksi. Setiap aksi yang berbeda diasosiasikan dengan satu kapabilitas dan satu kapabilitas yang berbeda dimiliki oleh satu <i>service</i> sehingga tidak ada tumpang tindih kapabilitas sehingga memiliki batasan yang jelas. • Deployment tidak termasuk pada penelitian ini, namun <i>service</i> dirancang untuk bisa dideploy pada lingkungan yang independen. (bisa menggunakan containerization)
<i>Loose coupling</i>	<ul style="list-style-type: none"> • Adanya <i>service contract</i> • Syarat konsumsi kapabilitas yang minimum • Konteks fungsional <i>service</i> tidak bergantung pada logika bisnis 	<ul style="list-style-type: none"> • <i>Service contract</i> individu <i>service</i> telah dirancang pada tahap <i>service oriented design</i>. • Syarat konsumsi kapabilitas dirancang seminimal dan seumum mungkin. • Konteks fungsionalitas tidak bergantung pada logika bisnis, contohnya pada pembuatan surat, <i>service</i> surat memiliki syarat body request yang general. Tidak spesifik pada SPD dan SKM.
Abstraction	<ul style="list-style-type: none"> • <i>Service contract</i> hanya berisikan informasi yang penting terkait kapabilitas <i>service</i>. • <i>Service</i> mengabstraksi informasi mengenai teknologi, logika dan fungsi. • <i>Service contract</i> memiliki syarat interaksi yang ringkas. 	<ul style="list-style-type: none"> • <i>Service contract</i> hanya berisikan kapabilitas dan deskripsi kapabilitas, input yang diperlukan serta output yang dihasilkan. • <i>Service</i> tidak memberikan informasi mengenai teknologi, logika dan fungsi. Konteks fungsionalitas terdapat pada daftar <i>service contract</i> masing masing <i>service</i>. • Syarat untuk bisa menggunakan kapabilitas ialah menggunakan metode HTTP standar dan media type sederhana.
<i>Formal contract</i>	<ul style="list-style-type: none"> • <i>Service</i> yang berada di dalam satu <i>service inventory</i> yang sama memiliki standar <i>service contract</i>. • <i>Service contract</i> disediakan oleh masing masing <i>service</i>. • <i>Service contract</i> distandarisasi dengan mengaplikasikan standar <i>design</i>. 	<ul style="list-style-type: none"> • Pada penelitian ini hanya dibuat satu buah <i>service inventory</i>. <i>Service inventory</i> yang dibuat memiliki kontrak standar yang telah ditetapkan pada tahap uniform contract modelling. • Setiap <i>service</i> memiliki <i>service contract</i> nya masing masing. • <i>Service contract</i> masing masing <i>service</i> mengacu pada <i>uniform contract</i> yang telah dibuat pada tahap awal menentukan teknologi arsitektur pada tahap <i>service inventory analysis</i>. <i>Uniform contract</i> berperan sebagai standar perancangan <i>service contract</i>.
<i>Reusability</i>	<ul style="list-style-type: none"> • <i>Service</i> didefinisikan dengan konteks fungsional agnostik • Logika <i>service</i> dibuat seumum mungkin • Terdapat <i>service contract</i> yang generic 	<ul style="list-style-type: none"> • Dalam mendefinisikan <i>service</i> dilakukan dekomposisi proses bisnis menjadi aksi aksi lalu diidentifikasi aksi yang bersifat agnostik dan non <i>agnostic</i>. Aksi <i>agnostic</i> menjadi kapabilitas <i>entity service</i> dan <i>utility service</i>, sementara yang spesifik terhadap <i>logic</i> proses bisnis (non-<i>agnostic</i>) dijadikan sebagai kapabilitas <i>task service</i>. • <i>Logic service</i> dibuat segeneric mungkin terutama untuk <i>utility service</i>. • Syarat untuk mengkonsumsi kapabilitas dibuat <i>segeneric</i> mungkin.

Evaluasi prinsip inti SOA mengacu pada syarat implementasi dan karakteristik desain SOA (Erl, Carlyle, Pautasso, & Balasubramanian, 2012). Berdasarkan evaluasi prinsip inti SOA, peneliti telah memenuhi syarat implementasi dan karakteristik desain prinsip inti SOA.

KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, dapat ditarik kesimpulan dari hasil dan pembahasan penelitian bahwa sistem yang telah dibangun mampu memenuhi kebutuhan proses bisnis layanan Kemahasiswaan SPD dan SKM BAAK. Namun dari sisi kebutuhan pengguna, sistem mampu memenuhi 75% kebutuhan pengguna dan memenuhi 90,77% kriteria penerimaan sistem. Sistem yang telah dibangun telah memiliki interoperabilitas dan kegunaan ulang karena sudah mengadopsi *Service-Oriented Architecture*. Selama perancangan arsitektur, peneliti telah mengaplikasikan prinsip utama *service-orientation* yaitu *autonomy*, *abstraction*, *formal contract*, *loose coupling*. Sistem yang dibangun sudah memenuhi syarat PWA. Untuk penelitian selanjutnya, peneliti menyarankan untuk dilakukan evaluasi antar muka dan arsitektur sistem menggunakan metode yang sudah ada serta memperluas *service inventory* untuk melengkapi layanan yang ada pada kemahasiswaan BAAK.

DAFTAR PUSTAKA

- Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change Second Edition*. Massachusetts: Addison Wesley Professional.
- Brocke, J. v., & Hevner, A. (2020). *Design Science Research. Cases*. Cham: Springer.
- Erl, T. (2009). *Service-Oriented Architecture: Concepts, Technology and Design*. Indiana: Prentice Hall.
- Erl, T., Carlyle, B., Pautasso, C., & Balasubramanian, R. (2012). *SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST*. New Jersey: Prentice Hall.
- IBM. (2021, April 7). *SOA (Service-Oriented Architecture)*. Gogilwe go tswa IBM: <https://www.ibm.com/cloud/learn/soa>
- OMG. (2012). *Service oriented architecture Modeling Language (SoaML) Specification*. Massachusetts: Object Management Group.
- Othman, A., & Gloaguen, R. (2013). River Courses Affected by Landslides and Implications for Hazard Assessment: A High Resolution Remote Sensing Case Study in NE Iraq–W Iran. *Remote Sensing*, 5(3), 1024–1044.
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach*. New York: McGraw-Hill.
- Richard, S. (2020, February 24). *web.dev*. (Google) Gogilwe July 5, 2022, go tswa <https://web.dev/pwa-checklist/>
- Richard, S., & LePage, P. (2020, January 6). *What are Progressive Web Apps*. Gogilwe go tswa [web.dev: https://web.dev/what-are-pwas/](https://web.dev/what-are-pwas/)
- Rosen, M., Lublinsky, B., T.Smith, K., & Balcer, M. J. (2008). *Applied SOA: Service-Oriented Architecture and Design Strategies*. Indianapolis: Wiley Publishing, Inc.
- STIS, P. S. (2020). *RENCANA STRATEGIS POLITEKNIK STATISTIKA 2020-2024*. DKI Jakarta: Politeknik Statistika STIS.
- Whitten, J. L., & Bentley, L. D. (2007). *System Analysis & Design Methods*. New York: McGraw-Hill/Irwin.
- Wijayanto, A. W., & Suhardi. (2014). Service Oriented Architecture Design using SOMA for Optimizing Public Satisfaction in Government Agency. *International Conference on ICT for Smart Society (ICISS)* (dits. 1-20). Bandung: Information Technology Research Group Institute of Technology Bandung.

