

Perancangan Portal Interoperabilitas *E-Government* Sebagai Platform Integrasi Sistem Informasi Pemerintahan Kota Denpasar

Mochammad Rizki Romdoni

Program Studi Sistem Informasi STT Indonesia Tanjungpinang

e-mail: m_rizki_r@yacanet.com

Abstrak

Pemerintah Kota Denpasar telah melewati fase perkembangan *E-Government* kedua yaitu *interactive government*. Untuk mencapai fase ketiga *integrated government* yang memiliki ciri adanya *multiple transaksi* yang melibatkan antar berbagai lembaga pemerintahan; telah terkonsep pada program kerja DISKOMINFO, namun belum terlaksana di lapangan. Berdasarkan hal tersebut penelitian ini ditujukan untuk mengembangkan sebuah arsitektur berbasis SOA (*Service Oriented Architecture*) yang diberi nama PIE (*Portal Interoperabilitas E-Government*), yang akan digunakan oleh SKPD (*Satuan Kerja Perangkat Daerah*) Pemerintah Kota Denpasar dalam berintegrasi dan berbagi sumber daya antar SKPD dengan mudah dan dapat diakses serta dimanfaatkan oleh masyarakat pengguna. Hasil penelitian adalah mengintegrasikan sistem informasi pemerintahan melalui PIE dengan mengikuti prinsip-prinsip dalam SOA.

Kata kunci: *interactive government, E-Government, service oriented architecture*

Abstract

Municipal city of Denpasar has passed the second phase of development of *E-Government*. Today, the existing phase is *interactive government*. To reach the third phase of *integrated government* where its characteristic is *multiple transactions* should involve various government agencies and until now DISKOMINFO remains continue to design but has not been done. Based on this research is aimed at developing SOA (*Service Oriented Architecture*) called PIE (*Portal of E-Government Interoperability*), which will be used by SKPD (*task force of government*) to integrate and share resources between SKPD easily and can be accessed and utilized by the community users. The results are to integrate the information systems of government through PIE based on the principles of SOA.

Keywords: *interactive government, E-Government, service oriented architecture*

1. Pendahuluan

Dinas Komunikasi dan Informatika Kota Denpasar telah memiliki program kerja rencana pengembangan *E-Government* tahun 2011-2015 yang salah satunya adalah "*terjalin komunikasi dan koordinasi antar instansi melalui jaringan komputer*" [1]. Salah satu hal yang belum tercapai dari program tersebut adalah belum adanya koordinasi pada level sistem informasi atau perangkat lunak diantara SKPD (*Satuan Kerja Perangkat Daerah*) atau Badan pemerintah Kota Denpasar [1], sehingga hal tersebut mengakibatkan duplikasi data kerap terjadi di setiap pengelola dan penyelenggara sistem informasi pemerintah (sumber daya yang *overlap* atau tidak terkoordinasi), sulitnya melakukan sinergi informasi digital, dan validasi data secara elektronik tidak dapat dilakukan untuk mendapatkan data yang akurat [2].

Solusi yang diusulkan untuk mengatasi permasalahan diatas telah banyak dilakukan misalnya mengintegrasikan sistem informasi dua SKPD yaitu Dinas Perijinan dan Kependudukan menggunakan salah satu teknologi yang ada pada *framework .Net* yaitu WCF (*Windows Communication Foundation*) [3], *single database* untuk setiap aplikasi [4], atau menggunakan

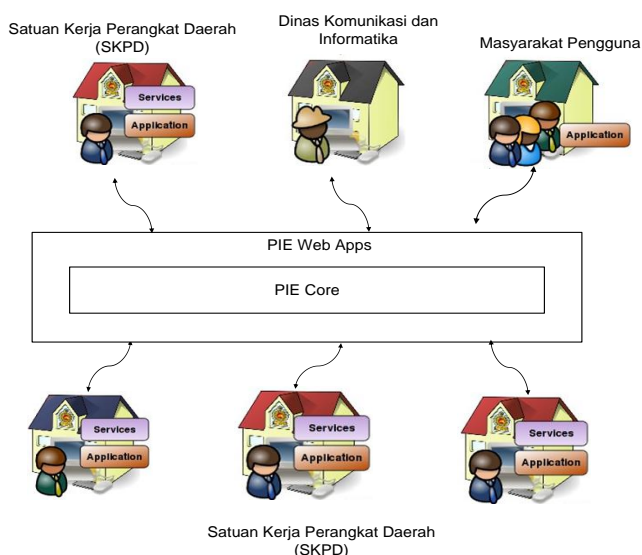
web service [5], dan lain-lain; namun kesemuanya itu masih menekankan pada satu teknologi interoperabilitas tertentu. Hal tersebut memiliki kelemahan yaitu bersifat *tightly-couple*, artinya ketika terjadi sebuah perubahan pada antar muka *service*-nya akan menimbulkan efek berantai ke perangkat lunak atau sistem informasi yang menjadi konsumernya untuk menyesuaikan dengan perubahan tersebut.

Integrasi sistem merupakan suatu keharusan, karena menjadi salah satu sasaran dari INPRES [6] yaitu pembentukan jaringan organisasi pendukung (*back-office*) yang menjembatani portal-portal informasi dan pelayanan publik dengan sistem pengolahan atau pengelolaan informasi yang terkait pada sistem manajemen dan proses kerja di instansi yang berkepentingan. Sasaran ini mencakup pengembangan kebijakan pemanfaatan dan pertukaran informasi antar instansi pemerintah pusat dan daerah [6]. Bila melihat tahap fase perkembangan *E-Government* pemerintah Kota Denpasar, maka saat ini telah sampai pada tahap ke dua yaitu "*interactive government*", yang bercirikan tersedianya isian formulir secara online dan dapat diunduh, kemudian dikembalikan melalui pos, fax, atau email; seperti situs bursa kerja online, *E-Procurement*, dan lain-lain. Oleh karena itu untuk peningkatan pelayanan publik terhadap masyarakat dan mengikuti prinsip-prinsip *good governance* sekarang ini sudah saatnya untuk meningkat menjadi "*integrated government*" yang memiliki ciri adanya *multiple* transaksi yang melibatkan antar berbagai lembaga.

Proses dan sistem yang terdapat di pemerintah Kota Denpasar semakin besar, kompleks, dan heterogen. Untuk mengintegrasikannya membutuhkan *effort* yang besar karena berhadapan dengan platform teknologi yang tidak seragam; satu SKPD menggunakan Java, yang lain PHP atau .Net, dan lain-lain. Berdasarkan hal tersebut, penelitian ini menawarkan sebuah solusi yaitu membentuk sebuah kanal yang dijadikan sebagai media pertukaran informasi dalam konteks antar portal atau sistem informasi pemerintahan, yang diberi nama dengan Portal Integrasi *E-Government* (PIE). PIE menyediakan mekanisme pengelolaan integrasi dan interkasi antar sistem informasi SKPD atau Badan yang efisien sehingga diperoleh penyederhanaan dalam proses integrasi.

2. Arsitektur PIE

Gambar 1 menjelaskan gambaran umum arsitektur sistem yang akan dibangun secara keseluruhan. Arsitektur tersebut terdiri dari beberapa entitas yaitu SKPD sebagai *service provider* dan *consumer*; Masyarakat Pengguna sebagai *service consumer*; Dinas Komunikasi dan Informatika sebagai *administrator*; dan PIE sebagai platform integrasi.



Gambar 1. Arsitektur PIE

2.1 PIE Web

PIE menyediakan antar muka berbasis *web* yang dikembangkan menggunakan bahasa pemrograman PHP sebagai titik akses untuk menghubungkan masyarakat pengguna ke *E-Service* dan sebagai tempat untuk mengelola *E-Service*-nya masing-masing SKPD pemerintah Kota Denpasar; khusus bagi Dinas Komunikasi dan Informatika (DISKOMINFO) digunakan untuk mengelola dan memelihara PIE secara keseluruhan.

2.2 PIE Core

PIE memiliki komponen penting lain, yaitu *PIE Core*. *PIE Core* menyediakan interkoneksi dan *discovery capabilities E-Service* yang disediakan oleh SKPD, di samping itu memfasilitasi *location transparency*, *transport protocol conversion*, *message transformation*, dan *security*. *PIE Core* merupakan enkapsulasi terhadap ESB (*Enterprise Service Bus*). Untuk mempersingkat waktu pengembangan, pada penelitian ini digunakan ESB open source yang sudah tersedia, yaitu Mule ESB versi ke 3.2 (community edition).

2.3 SKPD

Pemerintah Kota Denpasar dalam menjalankan roda pemerintahannya didukung oleh perangkat alat daerah, diantaranya adalah delapan belas SKPD dan Sembilan badan pemerintah; sedangkan secara administratif pengelolaan Kota di bagi menjadi empat kecamatan [7]. Namun pada penelitian ini, tidak seluruh perangkat daerah pemerintah Kota Denpasar yang akan diintegrasikan, tetapi hanya akan diambil beberapa sampel sebagai *pilot project* yaitu [1]:

1. Dinas Kesehatan (DINKES)
2. Dinas Kependudukan dan Catatan Sipil (CAPIL)

Kedua SKPD tersebut telah memiliki sistem informasi, namun masih parsial [1]; sedangkan DISKOMINFO sendiri berperan sebagai pengelola, pemelihara, dan memberikan sosialisasi serta pelatihan mengenai arsitektur PIE kepada SKPD, masyarakat pengguna, atau badan pemerintah lainnya dalam rangka melaksanakan salah satu program *E-Government* DISKOMINFO. Berikut adalah daftar sistem informasi setiap SKPD di dalam Tabel 1.

Tabel 1. Sistem informasi setiap SKPD

SKPD	Sistem Informasi
Dinas Kesehatan	Sistem Informasi Apotik pada Rumah Sakit
Dinas Kependudukan dan Catatan Sipil	Sistem Informasi Administrasi Kependudukan

2.4 Masyarakat Pengguna

Masyarakat pengguna adalah entitas dalam bentuk *software* yang mengkonsumsi *E-Service* yang telah disediakan oleh SKPD. Entitas tersebut bisa dikembangkan oleh seorang pengembang mandiri, vendor *software*, rekanan proyek sistem informasi pemerintah, atau yang lainnya. Setiap entitas dapat mengkonsumsi satu atau lebih *E-Service* SKPD; hal ini dikenal dengan istilah *composite services*.

3. Metodologi Penelitian

Gambaran secara garis besar mengenai langkah-langkah yang dilakukan dalam penelitian ini; mulai studi literatur sampai dengan memberikan kesimpulan dan saran.

1. Studi Literatur

Studi literatur, dimana literatur-literatur diambil dari penelitian sebelumnya, jurnal ilmiah baik dari dalam negeri maupun luar negeri, dan beberapa buku mengenai prinsip-prinsip SOA; selanjutnya, di tambah dengan manual book atau panduan dari instrumen penelitian yang digunakan.

2. Investigasi & Analisis Sistem

Manfaat dari fase investigasi adalah untuk menentukan problem-problem atau kebutuhan yang timbul, sedangkan analisis digunakan untuk mendefinisikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi, kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

3. Perancangan Sistem

Tahapan perancangan adalah tahapan mengubah kebutuhan yang masih berupa konsep menjadi spesifikasi sistem yang riil. Dalam menentukan bentuk sistem riil yang akan dibangun, dibantu oleh metodologi MDSE. Tahapan ini merupakan tahap yang sangat kritis, karena bila salah menentukan spesifikasi atau format sistem akan berujung pada kegagalan.

4. Implementasi

Implementasi merupakan tahap penerapan rancangan supaya siap dioperasikan. Tahap ini meliputi kegiatan penulisan kode program *PIE Web* dan *Core*. *PIE Web* menggunakan bahasa pemrograman PHP dengan pustaka Prado; sedangkan *PIE Core* menggunakan bahasa Java dengan pustaka Mule ESB.

5. Pengujian

Pengujian dilakukan untuk mengetahui komponen-komponen sistem telah berfungsi dengan baik. Dalam menguji perancangan dilakukan dengan cara mengimplementasikannya ke dalam bentuk *software* yang diberi nama *PIE*. Selanjutnya *PIE* di uji dengan pendekatan *black-box*.

6. Kesimpulan & Saran

Terakhir adalah melakukan evaluasi arsitektur sistem *PIE* secara keseluruhan; kemudian diikuti dengan memberikan kesimpulan terhadap hasil penelitian yang telah dilakukan dan mengajukan saran-saran yang diperlukan untuk penelitian selanjutnya pada bidang *software engineering* pada domain SOA.

4. Kajian Pustaka

4.1 SOA (Service-oriented Architecture)

Sebagian orang mengartikan SOA dengan nada sarkasme "*Same Old Architecture*", tetapi ini jauh dari sebuah kebenaran [8]. SOA adalah sebuah konsep yang digunakan ketika akan membangun sistem yang besar dan terdistribusi dengan variasi platform teknologi yang berbeda-beda. SOA adalah sebuah paradigma atau cara berpikir yang berdasarkan pada tiga konsep utama yaitu *service*, *interoperability* melalui *enterprise service bus* (ESB), dan *loose coupling*[9].

SOA bukanlah sebuah arsitektur yang konkrit; SOA adalah sesuatu yang mengarah pada arsitektur yang kongkrit (*..architectural paradigm for dealing with business processes..*); SOA bisa dipanggil dengan sebuah *style*, paradigma, konsep, perspektif, filosofi, atau gambaran. Artinya, SOA bukan sebuah *tools* atau *framework* nyata yang bisa kita beli, jadi ini adalah sebuah pendekatan, cara berpikir, *value system* (*moral code*) yang mengarah pada keputusan-keputusan tertentu saat merancang sebuah arsitektur *software*[9]. Menurut[9], konsep teknis dari SOA terdiri dari tiga yaitu *services*, *interoperability*, dan *loose coupling*.

4.1.1 Services

Secara teknis *service* adalah sebuah antarmuka (*interfaces*) untuk sebuah atau beberapa *messages* yang mengembalikan informasi dan/atau merubah sebuah *state* suatu entitas yang terkait (*backend*) [9]. Tujuan utama dari *services* adalah untuk merepresentasikan langkah-langkah fungsionalitas bisnis, artinya sebuah *services* harus mewakili fungsionalitas mandiri yang sesuai dengan kegiatan dunia bisnis di dunia nyata [9].

4.1.2 Interoperability

Tujuan utama SOA adalah menghubungkan sistem-sistem yang heterogen dengan mudah, dikenal dengan '*high interoperability*'. Ide ini tidak baru, sebelumnya telah dikenal dengan *enterprise application integration* (EAI). Penjelasan terperinci mengenai konsep *interoperability* dapat dilihat di subbab 4.2.

4.1.3 Loose Coupling

Loose couple dalam pengembangan aplikasi mengacu pada ukuran tingkat ketergantungan komponen *software* satu sama lain [10], dalam konteks SOA *loose coupling* adalah sebuah prinsip dimana *consumers* dan *services* terisolasi dari perubahan teknologi dan lingkungan yang mendasarinya; dalam beberapa hal tertentu, prinsip *loose coupling* mendeskripsikan pemisahan logis sebuah permasalahan. Artinya, *consumer* secara sengaja dipisahkan koneksinya secara fisik atau langsung dengan *services* yang maksudnya adalah untuk melindungi integritas *consumers* atau *provider* dan menghindari ketergantungan fisik diantara *services* [11].

4.2 Enterprise Service Bus (ESB)

ESB terkadang disebut sebagai "*messaging middleware*" tidak lebih dari sebuah platform yang dapat menghantarkan data antara berbagai aplikasi yang berlainan. Data dibawa ke dan dari serangkaian pemberhentian, yang dikenal sebagai "*endpoint*". Internal sebuah ESB berisi mekanisme *routing* yang mengetahui bagaimana mengarahkan data tertentu dari titik A ke titik B [12].

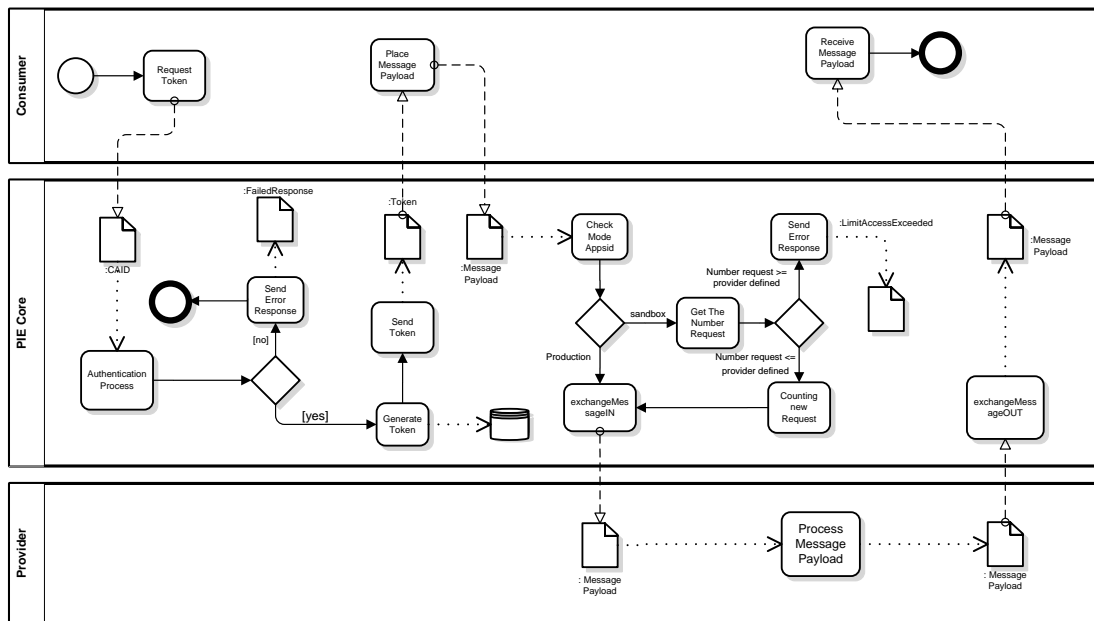
Gambar 4, mengilustrasikan sebuah ESB dengan bentuk sebuah saluran *logic* yang menjangkau masing-masing *endpoint*, yang memungkinkan data dapat dikirim atau diterima dari berbagai aplikasi melalui bus. Data ditransfer ke atau dari masing-masing *endpoint* menggunakan protokol tertentu, misalnya koneksi TCP atau HTTP. Namun ESB lebih dari sekedar protokol atau kanal komunikasi; tetapi merupakan sebuah *messaging framework* [12]. Intinya sebuah ESB adalah sebuah produk teknis untuk memecahkan permasalahan integrasi sistem.

5. Perancangan

Terdapat dua perancangan yaitu, arsitektur service PIE core dan PIE Web. PIE Core merupakan sebuah layer yang berfungsi sebagai jembatan atau kanal sistem informasi SKPD berinteroperabilitas; sedangkan PIE web bertugas sebagai interfaces dalam mengelola PIE.

5.1. Proses Bisnis PIE core

Gambar 2 menjelaskan proses bisnis konsumermengonsumsi *E-Service* yang disediakan oleh *provider*. Proses diawali oleh permintaan *token* dengan mengirimkan *message payload* yang berisi CAID (*Consumer Apps ID*). CAID diotentikasi, yang secara otomatis mendeteksi jenis transpor *message payload* dan menyesuaikan prosesnya; bila proses otentikasi berhasil, *token* di-generate menggunakan fungsi md5 dan disimpan kedalam *database*. Selanjutnya token dikirimkan kepada konsumen. *Token* tersebut digunakan untuk memelihara sesi komunikasi antara konsumen dan PIE Core. Menimbang faktor keamanan usia *token* dibatasi dalam waktu menit yang lamanya ditentukan oleh Admin PIE.



Gambar 2. Proses bisnis konsumsi E-Service provider (Diagram BPMN)

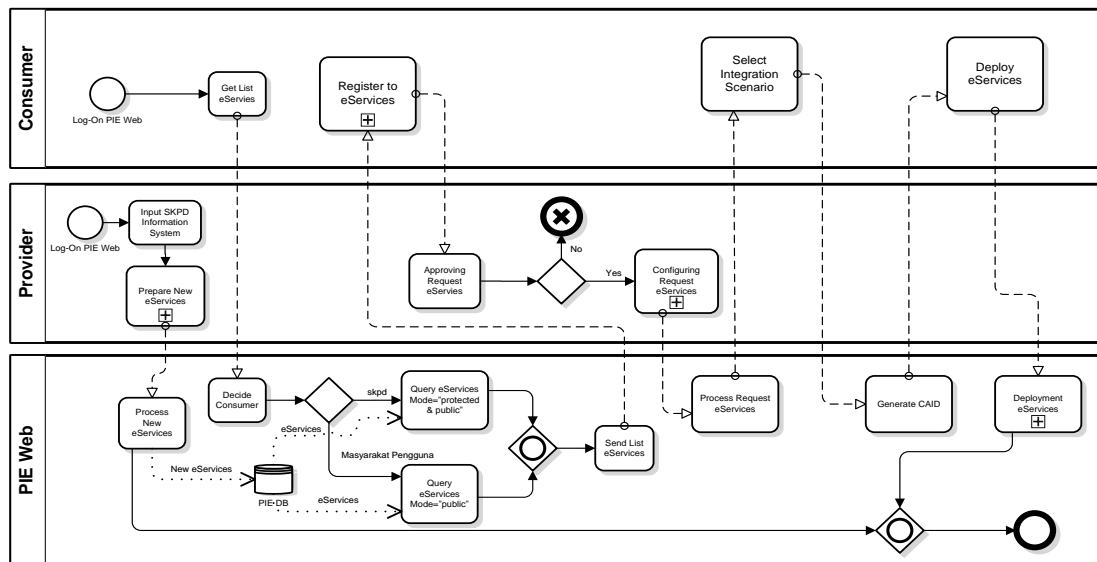
Konsumer dalam mengkonsumsi *E-Service* yang disediakan *provider*, tidak secara langsung ke *provider* tetapi melalui *PIE Core*. *PIE Core* disini bertindak sebagai *broker*. Konsumer mengirimkan *message payload* ke *PIE Core* dengan meletakkannya pada transpor tertentu, misalnya HTTP, FTP, JMS dan lain-lain. *PIE Core* melakukan pengecekan terhadap *E-Service* untuk mengetahui modenya; bila “*sandbox*” maka *task* “*get number request*” dieksekusi untuk mengetahui jumlah permintaanselama satu hari, seandainya hasilnya lebih besar dari yang telah di tetapkan oleh *provider* maka konsumsi *E-Service* dihentikan.

Mode CAID bertipe “*production*” atau jumlah *request* kurang dari yang telah ditetapkan, maka proses “*exchangeMessageIN*” dieksekusi yang hasilnya dikirimkan ke *provider*; *message payload* tersebut diproses, yang hasilnya dikirimkan kembali ke *PIE Core*; *PIE Core* pada “*exchangeMessageOUT*” mengirimkan hasilnya kembali ke konsumer. *Task* “*exchangeMessageIN*” dan “*exchangeMessageOUT*” dilakukan oleh Mule ESB yang meliputi transformasi *message payload* dari satu format ke format lainnya, misalnya JMS ke JSON atau JSON ke JMS. Mule ESB juga mengatur rute transpor untuk aliran *message payload* dari konsumer ke *provider* atau sebaliknya.

5.2. Proses Bisnis PIE Web

Didalam *PIE Web* terdapat proses bisnis yaitu pengelolaan *E-Service* yang di ilustrasikan pada Gambar 3. Proses bisnis tersebut terdiri dari tiga partisipan yaitu “konsumer”, “*provider*”, dan “*PIE Web*”. Didalam gambar diagram BPMN tersebut, “konsumer” dan “*provider*” di asumsikan telah terdaftar di dalam *PIE Web*. Proses ini, dimulai dari *log on* ke dalam *PIE Web*; setelah berhasil, *provider* menginputkan sistem informasi yang dibagi dengan entitas lain. *Provider* membuat dan mengkonfigurasi *E-Service* yang merupakan perwakilan satu fungsionalitas dari setiap sistem informasi. Setiap sistem informasi dapat memiliki *E-Service* lebih dari satu. *PIE Web* memproses *E-Service* baru dengan menyimpannya ke dalam *database PIE*.

Konsumer disaat membutuhkan informasi *E-Service* untuk dikonsumsi, melakukan *query* melalui *PIE Web* ke dalam *database PIE*. Hasil dari *query* tersebut tergantung pada jenis konsumernya; jika “SKPD” maka *E-Service* di-*filter* berdasarkan mode *protected* dan *public*; jika “Masyarakat Pengguna” yang bertipe *public*.

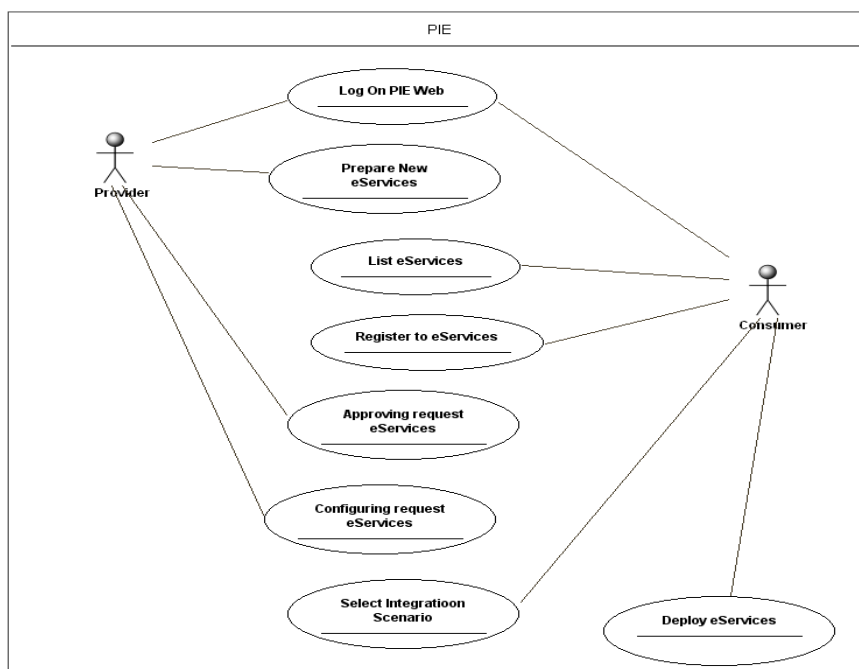


Gambar 3. Proses bisnis pie pengelolaan E-Service (Diagram BPMN)

Konsumer melakukan registrasi pada E-Service yang telah diperoleh, kemudian “provider” memberikan persetujuan apakah diterima atau ditolak; bila diterima provider mengkonfigurasi E-Service yang diminta oleh konsumer dan PIE Web menyimpan konfigurasi tersebut; terakhir konsumer memilih skenario integrasi dan men-deploy E-Service, yang kemudian dieksekusi oleh PIE Core. PIE Web men-generate CAID yang menjadi inputannya adalah AppsID E-Service dan ID registrasi.

5.3. Use Case PIE Web

Diagram use case adalah deskripsi kemampuan atau lingkungan sistem dari sisi setiap entitas (organisasi, divisi, software, dan lain-lain). Bagi developer sistem, usecase adalah sebuah alat yang digunakan untuk mengumpulkan kebutuhan dari sistem dari sudut pandang user yang terlibat. Gambar 4 diagram use case PIE Web.



Gambar 4. Diagram use case PIE web

6. Implementasi

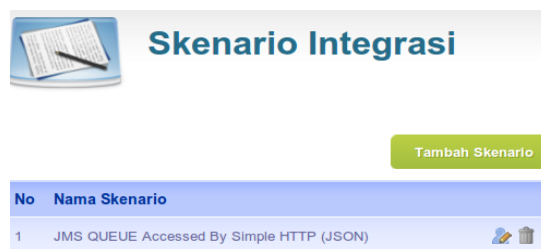
Implementasi adalah perwujudan atau realisasi dari rencana ide, model, spesifikasi desain, atau kebijakan. Dalam implementasi digunakan bahasa pemrograman Java, bahasa Visual Basic .Net, dan PHP; sedangkan untuk pustaka program menggunakan Mule ESB pada dan PRADO (*PHP Rapid Application Development Object-oriented*).

6.1 Antar Muka PIE Web

Antar muka merupakan bagian yang paling penting dari sebuah sistem. Antar muka merupakan tempat dimana komponen visual dan non visual diletakan untuk membentuk sebuah aplikasi. Setiap user (Masyarakat Pengguna, SKPD, Admin) memiliki antar mukanya masing-masing.

6.2.1 Halaman Admin

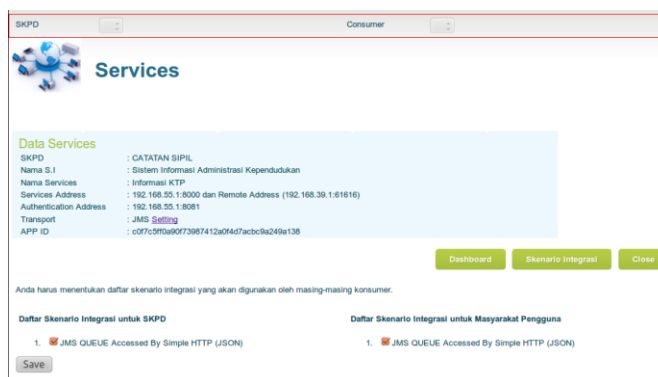
Halaman Admin terdiri dari beberapa menu penting yaitu “skenario integrasi, “SKPD”. Gambar 5, menunjukan daftar skenario integrasi yang telah di inputkan. SKPD berperan sebagai *provider* atau konsumer. Data SKPD tersebut ditambahkan oleh Admin.



Gambar 5. Daftar skenario integrasi

6.2.2 Halaman Provider (SKPD)

Provider adalah pihak yang menyediakan *E-Service* untuk di konsumsi oleh konsumer. Pada PIE yang bertindak sebagai *provider* adalah SKPD. Antar muka pada halaman *provider* dikaitkan dengan proses bisnis “*Prepare new eServices*”. Dalam proses bisnis tersebut dimulai dengan memilih sistem informasi SKPD yang sebelumnya telah di inputkan oleh *provider*. Gambar 6 menampilkan halaman untuk mengatur skenario integrasi yang digunakan oleh konsumer (SKPD dan Masyarakat Pengguna). Disamping itu terdapat informasi mengenai data *E-Service*, seperti nama *E-Service*, alamat, dan lain-lain.



Gambar 6. Halaman pengaturan skenario integrasi untuk konsumer

Gambar 7 menampilkan daftar konsumen yang mendaftar pada *E-Service*. *Provider* sebelum menyetujui, mengatur mode *E-Service* yang dikonsumsi oleh konsumen apakah *sandbox* atau *production*. Antarmuka ini sesuai dengan proses bisnis “*Configuring request eServices*”.

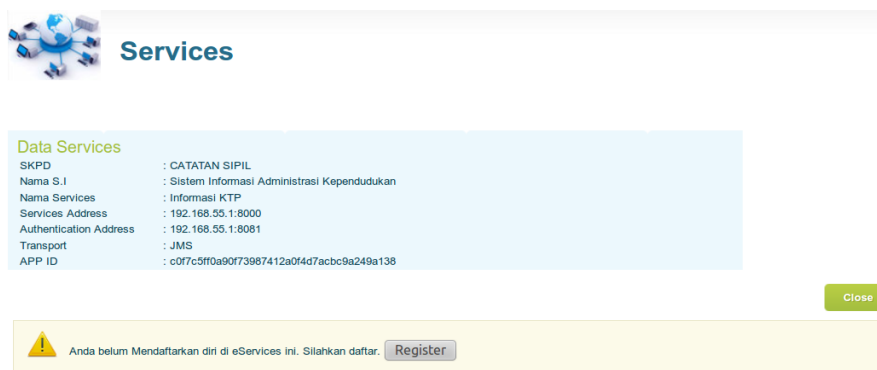


Gambar 7. Provider menyetujui request e-service dari konsumen

Provider menentukan jumlah *request* yang dapat diakses oleh konsumen perharinya. *Provider* pada halaman tersebut juga dapat mengubah mode *E-Service* *sandbox* atau *production*.

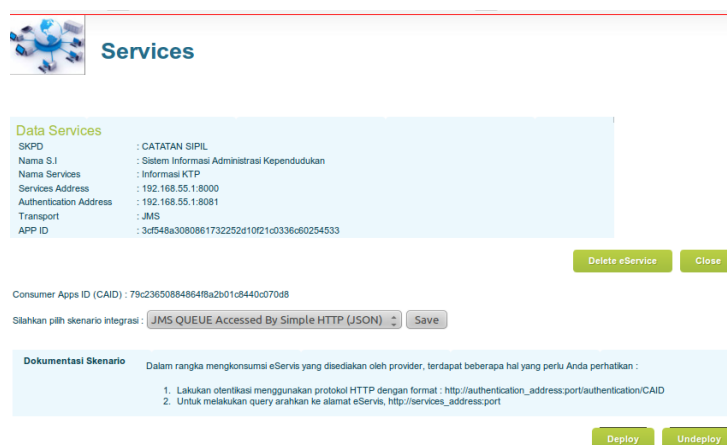
6.2.3 Halaman Konsumer

Konsumer (Masyarakat Pengguna dan SKPD), sebelum memanfaatkan sebuah *E-Service* harus terlebih dahulu mendaftar pada *E-Service* (gambar 8). Antarmuka ini berkaitan dengan proses bisnis “*Register to eServices*” dan “*Deploy eServices*”.



Gambar 8. SKPD Mendaftar pada E-Service SKPD Lainnya

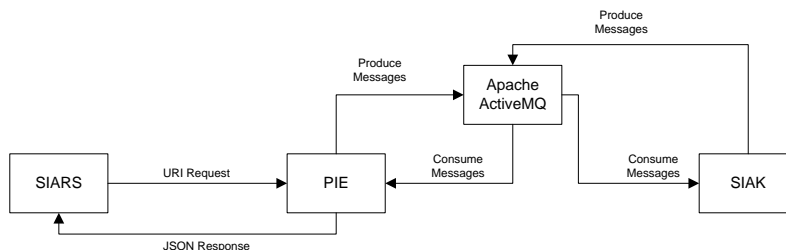
Setelah proses registrasi *E-Service* disetujui oleh *provider*, maka selanjutnya konsumen memilih skenario integrasi yang sesuai dengan proses bisnis internalnya masing-masing (Gambar 9). Konsumer mendapatkan CAID. CAID digunakan untuk mendapatkan *token* dari PIE. Terakhir konsumen, men-*deploy* *E-Service* tersebut.



Gambar 9. Konsumer memilih skenario integrasi dan men-deploy E-Service

7. Pengujian Integrasi Sistem

Test case utama dalam penelitian ini menggunakan dua buah prototipe sistem informasi (Gambar 10). Gambar tersebut mengilustrasikan test case pertukaran data dan protokol oleh partisipan PIE. Platform teknologi untuk berintegrasi yang digunakan oleh partisipan, menggunakan teknologi yang populer di pakai pada saat ini yaitu JMS dengan Apache ActiveMQ dan Web Service.



Gambar 10. Skenario integrasi SIARS dan SIAK melalui PIE

Sistem Informasi Apotik Rumah Sakit (SIARS) adalah sistem yang digunakan untuk membantu Apoteker dalam mengelola Apotik. SIARS dikembangkan menggunakan bahasa pemrograman Visual Basic .Net. Salah satu program pemerintah Kota Denpasar adalah menggratiskan biaya pengobatan beserta resepnya di rumah sakit pemerintah. Program ini memiliki syarat dan ketentuan yaitu hanya berlaku bagi warga Kota Denpasar yang dibuktikan dengan cara menunjukan KTP. Supaya program ini tepat sasaran maka diperlukan verifikasi keabsahan KTP ke SIAK (Sistem Informasi Administrasi Kependudukan) di Dinas Catatan Sipil.

SIAK (Sistem Informasi Administrasi Kependudukan) dikembangkan menggunakan bahasa pemrograman Java dengan Netbeans IDE. Sesuai skenario, SIAK mempublis E-Service dalam bentuk JMS ke Apache ActiveMQ. Dalam rentang waktu tertentu (detik) PIE mengkonsumsi message dari Active MQ yang selanjutnya dikirimkan kembali ke SIAK melalui protokol HTTP dengan format data JSON. Pengujian integrasi sistem SIAK dan SIARS dilakukan dengan cara pemanggilan E-Service SIAK oleh SIARS yang secara umum dilakukan untuk mengetahui apakah PIE Core berhasil mentransformasikan dan mengirimkan message payload dari konsumer ke provider dan sebaliknya. Dalam pengujian ini dikembangkan prototipe SIARS seperti Gambar 11 dan SIAK seperti Gambar 12.

Sistem Informasi Apotek pada Rumah Sakit

The screenshot shows a software window titled "Form_Pasien" with a sub-header "Data Pasien". It features several input fields: "No Pasien" (P12060001), "Tgl Lahir" (06/06/2012), "No. KTP", "Nama Pasien", "Jenis Kelamin", "Umur", "Alamat", and "Pekerjaan". Below these fields is a table with columns: "No_Pasien", "Nama_Pasien", "Jenis_Kelamin", "Tgl_Lahir", and "Umur". The table lists several patient records, with the first one highlighted. At the bottom, there are buttons for "Simpan", "Edit", "Hapus", "Batal", and "Keluar".

No_Pasien	Nama_Pasien	Jenis_Kelamin	Tgl_Lahir	Umur
P12042001	I Ketut Jompo	Pria	22/01/1994	18
P120427001	Dewik	Wanita	12/04/1967	25
P120501001	Wawan Setiawan	Pria	20/06/1981	31
P120501002	kadek.yudastraw...	Pria	12/10/1985	27
P120511001	I Komang Mahar...	Pria	18/08/1989	23

Gambar 11. Form input data pasien

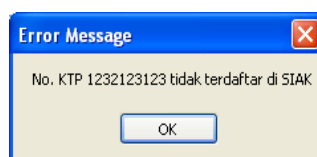
The screenshot shows a window titled "(SIAK) Sistem Informasi Administrasi Kependudukan" with a sub-header "Master Penduduk". It contains a "Refresh" button and a table with columns: "NO", "NIK", "Nama", "JK", and "Alamat". The table lists 12 residents with their respective identification numbers and addresses.

NO	NIK	Nama	JK	Alamat
1	5102213	MARIA TURMA DEVITA P	L	JL PANTAI IMPIAN GG KURAU
2	6101080	FAISAL	L	JL. GUDANG MINYAK NO. 22 TANJUNGPIN...
3	6102274	YOUKE LEANVANTORY DHWIETAMA	L	JL DURA 1 NO.14 PRUMNAS SEJANG
4	10101096	LILI DHARMAVANTI	P	JL. RAWASARI NO.17
5	10101096	APRILIA EKA FITRIANI	P	JLN. IR. SUTAMA RT.4/RW. 4 NO.16 TPI
6	10101058	SINNY	P	JL. RAWASARI NO 17 AA
7	10102001	JULIANTI	P	JL. SULTAN MACHMUD NO. 13
8	11101026	PUJIYANTO	L	PERUM LEMBAH ASRI BLOK D2 NO.05
9	11102114	HARIL ISKANDAR	L	JL. KARVA KP SIDO JASA BATU IX
10	11102118	MISRAWATI D	P	KAMPUNG MEKAR JAWA BT 8
11	28328323	MOHAMMAD RIZKI ROMDONI	P	JL. TUKAD BANYUSARI GG. XI NO. XI
12	2147485647	ANGGRAHA PUTRA YOANDA	L	PERUM CITRA PRATWI NO.1 BLOCK A K...

Gambar 12. Form daftar master penduduk

Sesuai proses bisnis konsumsi *E-Service* (Gambar 3), SIARS melakukan otentikasi dengan melakukan *request* menggunakan protokol HTTP ke alamat dan format berikut "http://192.168.55.1:8081/authentication/CAID" yang nilai kembaliannya adalah sebuah *token*. Langkah selanjutnya SIARS mengkonsumsi *E-Service*, dengan cara menyertakan *token* dalam setiap *request*-nya ke alamat dan format berikut "http://192.168.55.1:8000/token/noktp"; informasi alamat *E-Service* dan port-nya diperoleh dari data *E-Service* (Gambar 9).

Petugas apoteker di rumah sakit menginputkan data pasien melalui *form* data pasien. Didalam form tersebut terdapat beberapa isian diantaranya nomor pasien, nomor KTP, nama pasien, jenis kelamin, dan lain-lain. Di saat Petugas menekan tombol *save* maka SIARS menghubungi PIE untuk mendapatkan validitas no. KTP yang outputnya bila gagal, tampil sebuah *message box* seperti Gambar 13. PIE *Core* mentransformasi *message payload* melalui transpor HTTP menjadi JMS, kemudian mengirimkannya ke ActiveMQ. Secara *random* SIAK melakukan pengecekan Queue di ActiveMQ, bila terdapat *message* di Queue maka akan di konsumsi; hasilnya dikirimkan kembali ke ActiveMQ, selanjutnya PIE melakukan transformasi ke dalam bentuk JSON.



Gambar 13. Message Box No. KTP Tidak Terdaftar di SIAK

Melalui dua prototipe tersebut validitas fungsional PIE diuji yaitu dengan mengintegrasikan sistem informasi yang berbeda *platform* telah memberikan hasil yaitu respon kegagalan untuk proses validasi no KTP dengan nilai input bertipe *string* yang di *request* dari SIARS ke SIAK; hal ini disebabkan karena no KTP yang di *request* tidak terdaftar di SIAK.

8. Simpulan

Berdasarkan hasil uji coba yang telah dilakukan dapat disimpulkan hal-hal sebagai PIE berhasil mengintegrasikan dua sistem informasi pemerintahan yang berbeda *platform* yaitu SIARS (DINKES) dan SIAK (CAPIL). Pemilihan pustaka Mule ESB sebagai *core* dari PIE berperan dengan baik dalam mendukung arsitektur PIE. PRADO sebagai *framework* PHP dapat mempercepat proses pengembangan PIE *Web*.

Daftar Pustaka

- [1] Ketut Agus Indra Diatmika, S.Kom., Staf Pos dan Telematika Dinas Komunikasi dan Informatika Kota Denpasar. (komunikasi pribadi, 02 Januari 2012).
- [2] <http://biogen.litbang.deptan.go.id/wp/terbitan/Presentasi-Herry%20Abdul%20Aziz.pdf>, [diakses tanggal 24 September 2012]
- [3] Nofian Adi Prasetyawan, Rancang Bangun Framework Berbasis .NET Framework Menggunakan Konsep SOA Studi Kasus: E-Government Pada Dinas Perijinan dan Dinas Kependudukan, Tugas Akhir, Institut Teknologi Sepuluh November, <http://www.apitika.kominfo.go.id/forumegov2011/unduh.php?name=pengelolaan+integrasi+informasi+dan+pertukaran+data.pdf>, [diakses tanggal 30 september 2012]
- [4] Ali Nasrun, Rully Agus Hendra, Muhammad Priandi, Urgensi Integrasi Sistem Informasi Akuntansi Instansi Pemerintah, Jurnal Teknik ITS. Volume 1, 2012.
- [5] Franco Arcieri, Elettra Cappadozzi, Enrico Nardelli, Maurizio Talamo. SIM : a working example of an E-government service infrastructure for mountain communities. Database and Expert Systems Applications (DEXA). Munich, 2001: 407-41.
- [6] Instruksi Presiden RI No. 3 Tahun 2003 tentang Kebijakan dan Strategi Nasional pengembangan E-Government.
- [7] <http://www.denpasarkota.go.id/instansi/?cid===wN&s=menu&id=563>, [diakses tanggal 5 Januari 2012].
- [8] <http://tutorials.jenkov.com/soa/soa.html>, diakses tanggal 22 Nopember 2011
- [9] Nicolai M. Josutti, SOA in Practice: The Art of Distributed System Design. California: O’Rielly Media, Inc, 2007.
- [10] Tom Yuan Gao, The Complete Reference To Professional SOA With Visual Studi 2005 (C# & VB 2005) .NET 3.0. US: Lulu Press, 2007.
- [11] James Bean, SOA and Web Services Interfaces Design: Principles, Techniques, and Standars. Burlington : Elseveir, 2010.
- [12] Peter Delia, Antoine Borg, Ricston Ltd. Mule 2 A Developer’s Guide. Apress, 2008.