

IMPLEMENTASI SERVICE ORIENTED ARCHITECTURE PADA SISTEM PENGELOLAAN DATA ASURANSI BPJS KARYAWAN PT. TRANSPORTASI JAKARTA

Noor Adikrodadi Pakusadewa, Slamet Sudaryanto N., ST, M.Kom
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Imam Bonjol No. 207, Semarang, 50131, (024) 3517261
111201005380@mhs.dinus.ac.id, slametalica301@gmail.com

Abstrak

Perkembangan teknologi sangatlah pesat terutama dibidang teknologi informasi dan komputer (TIK). TIK sering dimanfaatkan dalam dunia bisnis adalah Sistem Informasi. Sistem Informasi memegang peranan penting dalam berbagai bidang pekerjaan seperti pendidikan, perdagangan, kedokteran, jasa, dan lain-lain. Dalam sebuah perusahaan keberadaan sistem informasi sangatlah diperlukan. Salah satu pemanfaatan sistem informasi adalah dapat digunakan sebagai pengolahan data seperti keuangan, penjualan dan karyawan. Untuk mengelola data asuransi karyawan dengan jumlah yang banyak PT. Transportasi Jakarta membutuhkan alat yang lebih efektif dan efisien dari cara konvensional. Selain itu terdapat masalah perilaku tidak tertibnya pembayaran iuran BPJS menyebabkan proses klaim asuransi dan rujukan menjadi kacau sehingga diperlukan sistem yang dapat mempermudah pengelolaan data dan rujukan untuk asuransi BPJS untuk karyawan PT. Transportasi Jakarta. Untuk mengatasi efektifitas dan efisiensi masalah pengelolaan data dan rujukan asuransi BPJS karyawan PT. Transportasi Jakarta peneliti mengusulkan untuk merancang system pengelolaan data dan rujukan asuransi karyawan dengan Service Oriented Architecture (SOA) berbasis web service. Tujuan dari penelitian ini adalah mengimplementasikan SOA dalam sistem pengelolaan data asuransi BPJS dan perusahaan terdaftar. Hasil dari penelitian yaitu SOA dengan web service berhasil diimplementasikan pada sistem BPJS. Dengan demikian komunikasi data antara perusahaan-perusahaan dapat dengan mudah dilakukan dan mempercepat proses dalam perujukan asuransi karyawan perusahaan.

Kata Kunci: Service Oriented Architecture, Web Service, Asuransi

Abstract

Technology is extremely rapidly development especially in the field of information and computer technology (ICT). ICT is often utilized in the business world is information systems. Information systems play an important role in various fields of work such as education, trade, medical, services, and others. In a company the existence of information system is absolutely necessary. One of the utilization of information system is can be used as the data processing such as finance, sales and employees. Insurance data to manage employees with great numbers of PT Transportasi Jakarta needs tools are more effective and efficient than conventional. In addition there are behavior problems do not cause the BPJS dues payment Martinet process insurance claims and references to be screwed so the needed system which can simplify data management and referrals for BPJS insurance for employees of PT Transportasi Jakarta. To address the effectiveness and efficiency of the management of data and issues the BPJS insurance reference for employees of PT Transportasi Jakarta researchers proposing to design a system for data management and referral of insurance employees with Service Oriented Architecture (SOA) based web service. The purpose of this research is to implement SOA in insurance data management of BPJS system and registered company. Results from research is SOA based web services successfully implemented on the system of the BPJS. Thus the data communication between companies can be easily done and speed up the process in the referring employee insurance company.

Keywords: Service Oriented Architecture, Web Service, Insurance

1. PENDAHULUAN

Saat ini perkembangan teknologi sangatlah pesat terutama dibidang teknologi informasi dan komputer (TIK). TIK yang sering dimanfaatkan dalam dunia bisnis adalah Sistem Informasi. Sistem Informasi memegang peranan penting dalam berbagai bidang pekerjaan seperti pendidikan, perdagangan, kedokteran, jasa, dan lain-lain. Dalam sebuah perusahaan keberadaan sistem informasi sangatlah diperlukan. Dengan sistem informasi dapat mempercepat pengolahan data dan dapat menghasilkan informasi yang diperlukan. Informasi yang digunakan sangat beraneka ragam sesuai dengan kebutuhan informasi yang dibutuhkan. Untuk menghindari kesalahan dalam informasi maka diperlukan suatu sistem informasi yang membantu kelancaran dan kemudahan dalam melaksanakan pekerjaan.

Salah satu pemanfaatan sistem informasi adalah dapat digunakan sebagai [1] pengolahan data seperti keuangan, penjualan dan karyawan. PT. Transportasi Jakarta merupakan perusahaan yang bergerak dibidang transportasi sebagai pengelola bus rapid transit yang mempunyai karyawan sebanyak 5504 orang. Setiap karyawan PT. Transportasi memiliki asuransi kesehatan yang terdaftar pada Badan Penyelenggara Jaminan Sosial (BPJS) Kesehatan. Untuk mengelola data asuransi karyawan dengan jumlah yang banyak PT. Transportasi Jakarta membutuhkan alat yang lebih efektif dan efisien dari cara konvensional. Selain itu terdapat masalah perilaku tidak tertibnya pembayaran iuran BPJS menyebabkan proses klaim asuransi dan rujukan menjadi kacau sehingga diperlukan sistem yang dapat mempermudah pengelolaan data dan rujukan untuk asuransi BPJS untuk

karyawan PT. Transportasi Jakarta.

Untuk mengatasi efektifitas dan efisiensi masalah pengelolaan data dan rujukan asuransi BPJS karyawan PT. Transportasi Jakarta peneliti mengusulkan untuk merancang sistem pengelolaan data dan rujukan asuransi karyawan dengan Service Oriented Architecture (SOA) berbasis web service. SOA merupakan suatu pendekatan untuk perancangan aplikasi yang dibangun secara modular atau berupa service. Service yang dibangun dapat digunakan kembali sesuai dengan integrasi bisnis yang ada. Seiring dengan perkembangan sebuah aplikasi yang dituntut untuk memenuhi integrasi bisnis yang ada, perancangan aplikasi menggunakan pendekatan SOA yang akan dapat mengurangi effort untuk merubah sebuah aplikasi sesuai dengan integrasi bisnis, sehingga aplikasi tersebut dapat berubah dengan cepat serta menghemat usaha dalam perubahannya. Dengan pendekatan SOA, sebuah service dibangun dalam aplikasi juga dapat bekerja pada platform yang berbeda. Sedangkan web service merupakan bentuk implementasi SOA. Web service menggunakan seperangkat teknologi standar terbuka yang memungkinkan sebuah aplikasi dapat berkomunikasi dan menyediakan layanan bagi aplikasi lain melalui jaringan komputer [2].

Tujuan dari laporan tugas akhir yang dibuat oleh penulis adalah sebagai berikut :

1. Transportasi dalam proses pengelolaan data asuransi karyawan.

Mengimplementasikan Service Oriented Architecture dalam sistem pengelolaan data asuransi BPJS dan PT Transportasi.

2. SERVICE ORIENTED ARCHITECTURE

1. SOA

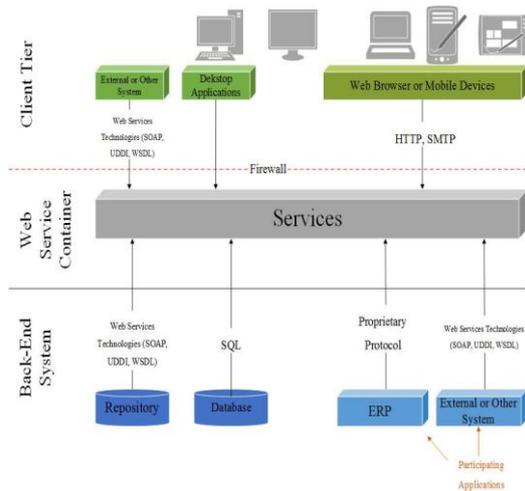
Menurut Thomas Erl (2005) Service Oriented Architecture (SOA) adalah bagian utama dari *service computing platform* yang membawa konsep, teknologi, dan tantangan baru. Menurut Thomas Erl ada tiga hal penting yang menjadikan sebuah infrastruktur dapat disebut sebagai *Service Oriented Architecture*, yaitu logika bisnis yang dikapsulasi sebagai *service*, dan proses komunikasi antar *service* dengan menggunakan *message*. Dalam hal ini, *service layer* akan menjembatani hubungan antara *business logic* dan *application logic* [2]. Menurut Kunal Mittal (2007), *Service Oriented Architecture* adalah sebuah kumpulan yang terdiri atas *tools*, teknologi, *framework*, dan *best practice* yang memudahkan implementasi sebuah *service* secara cepat. Proses dalam mengimplementasi SOA menggunakan metodologi yang mengidentifikasi *service* yang dapat dipergunakan kembali (*reusable*) dalam aplikasi dan organisasi suatu perusahaan. Dengan demikian, SOA adalah suatu ide, bukan merupakan teknologi, produk, ataupun standar. Arsitektur SOA difokuskan untuk mengidentifikasi, membangun, mengubah, dan memelihara proses bisnis suatu perusahaan sebagai sekumpulan *service*. Teknologi yang menggunakan SOA digunakan untuk mengurangi kompleksitas dalam membangun sebuah aplikasi atau *software*. SOA dapat mengantisipasi isu mengenai penggunaan *software* yang terdistribusi, penggunaan *platform* yang berbeda, dan integrasi aplikasi.

Menurut Ricardo LaRosa (2008) *Service Oriented Architecture* merupakan sebuah konsep yang sederhana. SOA adalah *blueprint* yang dapat membantu layanan-layanan dapat

beroperasi secara bersama-sama. Layanan yang dimaksud adalah sebuah unit *software* yang mendukung fungsi-fungsi bisnis agar dapat bekerja secara maksimal.

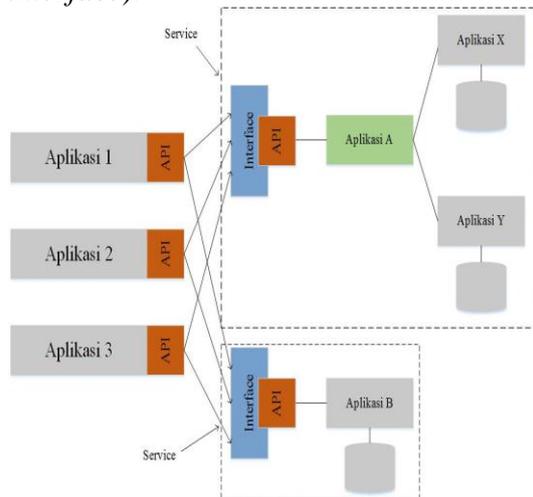
Dapat disimpulkan bahwa SOA adalah suatu cara mengorganisir perangkat lunak (*software*) sehingga organisasi dapat dengan cepat merespon perubahan kebutuhan. Teknologi tersebut berdasarkan *service* (layanan), yang terdiri dari unit-unit berdasarkan kebutuhan dari perangkat lunak yang berjalan pada jaringan. *Service* sendiri merupakan komponen umum yang digunakan oleh beberapa sistem aplikasi (*reusable*). *Service* dapat berupa modul program, aplikasi, atau gabungan dari beberapa aplikasi yang berhubungan. SOA merepresentasikan suatu model yang mana fungsi-fungsi dibagi menjadi beberapa unit-unit terpisah yang lebih kecil, yang dapat didistribusikan melalui jaringan dan dapat dikombinasikan dan digunakan secara bersama-sama untuk menciptakan aplikasi. *Service-service* tersebut berkomunikasi satu sama lain dengan cara mengirim data dari satu *service* ke *service* lainnya, atau dengan mengkoordinasikan suatu aktivitas antara dua atau lebih *service*. Sehingga SOA memungkinkan *service* yang *interoperable*, yang berarti *service-service* tersebut dapat berkomunikasi satu sama lain, meskipun pada implementasinya dibuat dengan bahasa pemrograman yang berbeda atau diakses melalui *transport protocol* yang berbeda yang memungkinkan pengintegrasian aset-aset sistem aplikasi dari suatu perusahaan.

Ilustrasi konsep dari *Service Oriented Architecture* (SOA) dapat dilihat pada gambar dibawah ini:



Gambar 2.1: ilustrasi SOA (Thomas Erl, SOA: principles of service design 2008)

Service merupakan komponen umum yang digunakan oleh beberapa sistem aplikasi (*reusable*). *Service* dapat berupa modul program, aplikasi, atau gabungan dari beberapa aplikasi yang berhubungan. *Service* direalisasikan dengan menambahkan *interface (wrapper)* pada satu atau sekelompok sistem aplikasi. Sistem aplikasi berkomunikasi dengan *service interface* melalui API (*application programmer interface*).



Gambar 2.2: konsep service (Thomas Erl, SOA: principles of service design 2008)

Service merupakan infrastruktur karena:

- Dipakai bersama oleh struktur di atasnya (sistem-sistem aplikasi).

- Relatif lebih stabil/statis dibandingkan aplikasi-aplikasi penggunaannya.
- Memiliki *life cycle* tersendiri.
- Dikelola (dirancang, dikembangkan, dan dipelihara) oleh pihak yang berbeda dengan pengelola aplikasi di atasnya.
- Dapat di-*outsource*.
- Menerapkan *Service Level Agreement*.

Dengan menyediakan *service*, keuntungan yang didapat antara lain:

- Dapat memanfaatkan ulang keahlian
- Meningkatkan kecepatan (*agility*) pengembangan sistem aplikasi
- Menurunkan kompleksitas pengembangan sistem aplikasi
- Memudahkan penjaminan keandalan dan kinerja semua sistem aplikasi yang menggunakannya

Service yang ideal adalah yang:

- *Decoupled* (terpisah secara logis) dari aplikasi penggunaannya.
- Terpisah secara fisik dari aplikasi penggunaannya (hubungan melalui *network*).
- Menggunakan teknologi *interface* standar netral (*open standard*) yang dapat berjalan pada *platform* yang berbeda, dibangun dengan bahasa pemrograman yang berbeda, serta bukan *propriety* vendor tertentu.

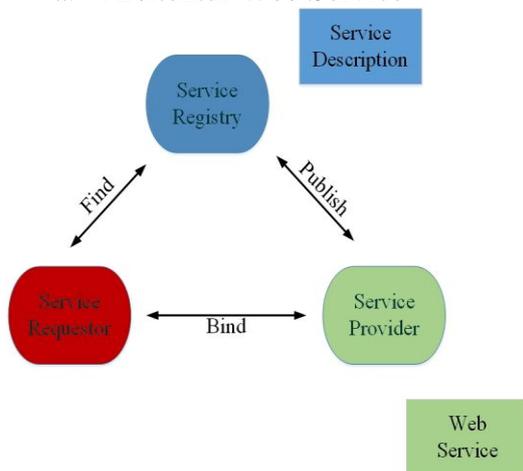
Sedangkan keuntungan utama dari standar SOA adalah memungkinkan *service* yang *interoperable*, yang berarti *service-service* tersebut dapat berkomunikasi satu sama lain, meskipun pada implementasinya dibuat dengan bahasa pemrograman yang berbeda atau diakses melalui *transport protocol* yang berbeda.

2. Web Service

Web service adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca oleh mesin. Sistem-sistem lainnya berinteraksi dengan *web service* menggunakan pesan SOAP yang umumnya dikirim melalui HTTP dalam bentuk XML dan JSON.

Definisi di atas diberikan oleh *World Wide Web Consortium (W3C)* yang merupakan badan yang menciptakan dan mengembangkan standar *web service*. Tetapi secara umum, *web service* tidak terbatas hanya pada standar SOAP saja. Salah satu pustaka yang mengulas lengkap tentang *web service* menyebutkan definisi yang lebih umum: *web service* adalah aplikasi yang diakses melalui *internet* menggunakan protokol standar *internet* dan menggunakan XML sebagai format pesannya.

a. Arsitektur Web Service



Gambar 2.3: arsitektur web service (Thomas Erl, SOA: principles of service design 2008)

Pada gambar di atas, ada tiga komponen yang membuat *web service* berjalan. Ketiga komponen itu adalah:

1. **Service provider**, merupakan pemilik *Web service* yang berfungsi menyediakan

kumpulan operasi dari *Web service*.

2. **Service requestor**, merupakan aplikasi yang bertindak sebagai klien dari *Web service* yang mencari dan memulai interaksi terhadap layanan yang disediakan.

3. **Service registry**, merupakan tempat dimana *Service provider* mempublikasikan layanannya. Pada arsitektur *Web service*, *Service registry* bersifat optional. Teknologi *web service* memungkinkan kita dapat menghubungkan berbagai jenis *software* yang memiliki *platform* dan sistem operasi yang berbeda.

b. Jenis Web Service

- REST

REST adalah salah satu jenis *web service* yang menerapkan konsep perpindahan antar *state*. *State* disini dapat digambarkan seperti jika *browser* meminta suatu halaman *web*, maka *server* akan mengirimkan *state* halaman *web* yang sekarang ke *browser*. Bernavigasi melalui *link-link* yang disediakan sama halnya dengan mengganti *state* dari halaman *web*. Begitu pula REST bekerja, dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Perintah HTTP yang bisa digunakan adalah fungsi *GET*, *POST*, *PUT* atau *DELETE*. Balasan yang dikirimkan adalah dalam bentuk XML sederhana tanpa ada protokol pemaketan data, sehingga informasi yang diterima lebih mudah dibaca dan diparsing disisi *client*.

Dalam pengaplikasiannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*. Maksud orientasi pada *resource* adalah orientasi yang menyediakan *resource-resource* sebagai layanannya dan bukan kumpulan-kumpulan dari aktivitas yang

mengolah *resource* itu. Alasan mengapa REST tidak digunakan dalam skripsi ini karena orientasi pada *resourcenya* itu. Selain itu, karena standarnya yang kurang sehingga tidak begitu cocok diterapkan dalam aplikasi yang membutuhkan kerjasama antar aplikasi lain, dimana standar yang baik akan sangat berguna karena berbicara dalam satu bahasa yang sama. Beberapa contoh *web service* yang menggunakan REST adalah: Flickr API (*Application Program Interface*), YouTube API, Amazon API.

- SOAP

SOAP merupakan protokol standar yang ringan dan ditujukan untuk pertukaran informasi dalam desentralisasi sistem. Protokol ini melakukan pemanggilan metode dalam bahasa XML (*Extensible Markup Language*). Namun, selain itu juga kita dapat menggunakan format pertukaran data lainnya yaitu JSON (*JavaScript Object Notation*). Dengan demikian, SOAP adalah suatu mekanisme yang sederhana untuk melakukan pertukaran struktur dan tipe informasi dalam lingkungan yang tersebar dan terdistribusi menggunakan XML/JSON. Dokumen SOAP yang digunakan untuk melakukan *request* disebut SOAP *request* sedangkan dokumen SOAP yang diperoleh dari web service disebut dengan SOAP *response*.

Secara garis besar, protocol SOAP dapat dibagi menjadi beberapa bagian berikut :

- a. SOAP Envelope

Untuk mendefinisikan apa yang terdapat dalam pesan (message) yang dikirim, dan kepada siapa pesan tersebut akan digunakan. Elemen ini merupakan root dari pesan dan bersifat wajib.

- b. SOAP Header

Elemen ini berisi informasi header dan bersifat opsional. Informasi yang ada pada header tidak ditampilkan di aplikasi.

- c. SOAP Body

Elemen ini berisi panggilan dan merespon informasi. Elemen ini wajib.

- d. SOAP Encoding Rules

Mendefinisikan mekanisme serialisasi yang dapat digunakan untuk pertukaran tipe data yang didefinisikan dari tiap aplikasi.

- e. SOAP RPC Representation

Menyediakan suatu standar yang dapat digunakan untuk mewakili remote procedure call (RPC) dan data hasil kembaliannya.

- f. SOAP Binding

Mendefinisikan standar yang digunakan untuk pertukaran SOAP Envelope antara peers dengan protokol yang umum untuk transportasi.

3. HASIL DAN PEMBAHASAN

1. Tahap Analisa

Analisa kebutuhan sistem perlu dilakukan agar tidak terjadi kesalahan dalam pembuatan sistem. Analisa sistem menentukan pengguna, data maupun proses pada sistem. Dengan analisis sistem diharapkan dapat diuraikan secara utuh menjadi komponen-komponen dasar dengan tujuan mengidentifikasi serta mengevaluasi permasalahan dan kebutuhan yang diharapkan. Analisa kebutuhan sistem meliputi kebutuhan fungsional dan kebutuhan non fungsional.

- Kebutuhan Fungsional

Kebutuhan fungsional berisi bagian dan proses yang harus ada pada sistem yang akan dibangun. Kebutuhan fungsional pada sistem dibagi dua yaitu untuk admin BPJS dan perusahaan yang telah terdaftar. Kebutuhan fungsional dari sistem yang diusulkan sebagai berikut:

- a. Data (Rumah Sakit, Perusahaan, Peserta)
- b. Fitur login admin BPJS

- c. Fitur kelola data perusahaan/ instansi
- d. Fitur lihat data karyawan / instansi
- e. Fitur approve request peserta / karyawan
- f. Fitur kelola data transaksi / rujukan
- g. Fitur kelola data rumah sakit
- h. Fitur kelola data tingkat

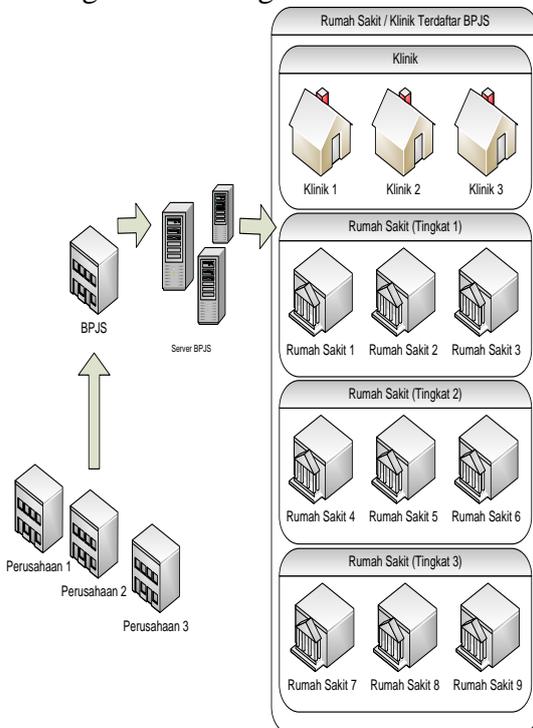
Kebutuhan fungsional bagi perusahaan yang telah terdaftar pada BPJS:

- a. Register karyawan
- b. Cek status karyawan
- c. Request rujukan untuk karyawan

2. Tahap Desain

a. Skema Sistem

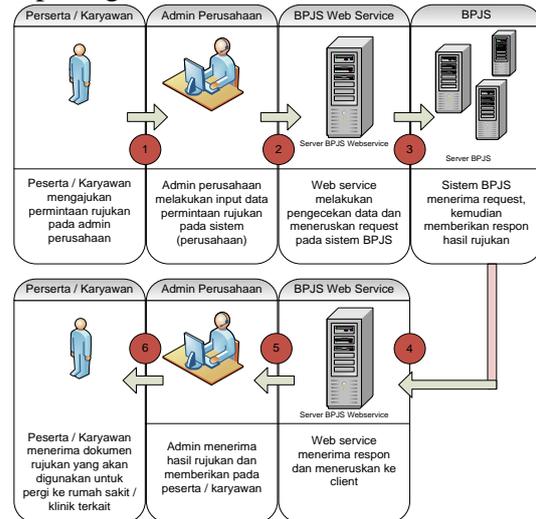
Pada tahap ini dilakukan analisa skema dari sistem yang ada pada BPJS. BPJS mempunyai daftar perusahaan dan rumah sakit atau instansi kesehatan yang terdaftar. Daftar instansi kesehatan mempunyai kategori berdasarkan tingkat yaitu poliklinik, tingkat 1, tingkat 2 dan tingkat 3.



Gambar 4.1 : Skema Sistem BPJS

b. Proses Bisnis Permintaan Rujukan

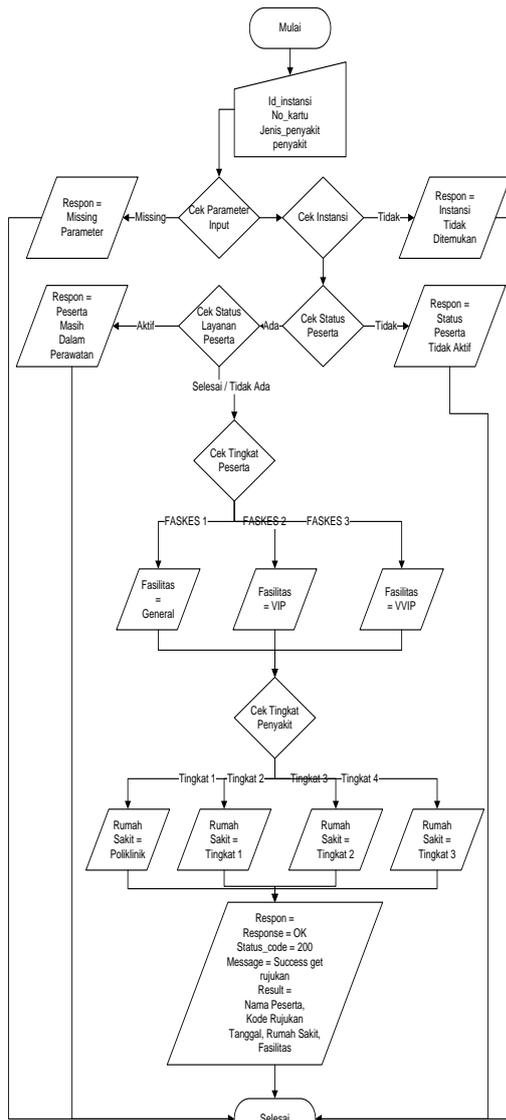
Tahap ini menggambarkan proses bisnis dari pengajuan rujukan oleh peserta. Pengajuan terdiri dari 6 tahapan. Mulai dari peserta mengajukan permintaan rujukan, proses rujukan dan peserta mendapat rujukan. Proses bisnis digambarkan pada gambar 4.5.



Gambar 4.5 : Proses Bisnis Permintaan Rujukan

c. Alur atau Flowcart Prosedur Rujukan

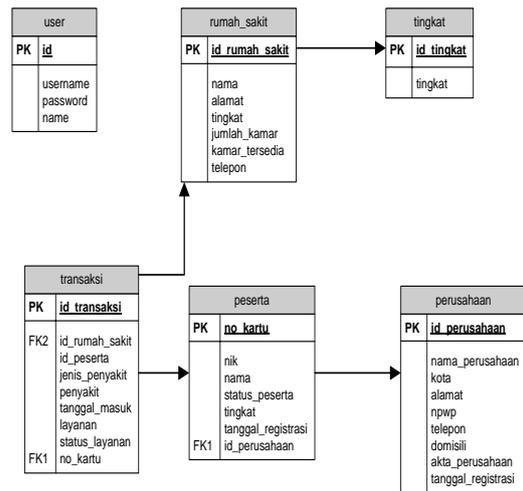
Pada tahap ini digambarkan alur prosedur mendapatkan rujukan secara teknis. Alur ditunjukkan dengan flowchart mulai dari tahap input sampai respon / hasil dari pengajuan rujukan.



Gambar 4.6: Flowchart Rujukan

d. Desain Database Sistem

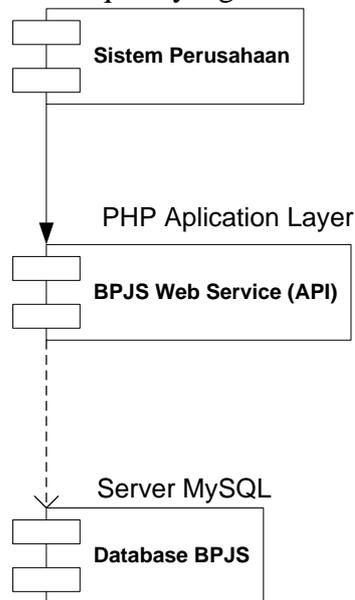
Tahap ini dilakukan desain database yang akan digunakan pada sistem. Database menggunakan MySQL sebagai Database Management System. Desain database yang telah dibuat terdiri dari 6 tabel yaitu user, rumah_sakit, tingkat, transaksi, peserta dan perusahaan.



Gambar 4.7: Desain Database

e. Deployment Diagram

Tahap ini dilakukan desain deployment diagram. Deployment diagram menggambarkan detail komponen dan alur yang ada pada sistem yang diusulkan. Perusahaan melakukan request pada sistem web service BPJS melalui protocol http, kemudian web service sistem BPJS merespon request dengan melakukan proses pengecekan pada database dan melakukan kembalian berupa respon atas request yang telah dilakukan.



Gambar 4.8: Deployment Diagram

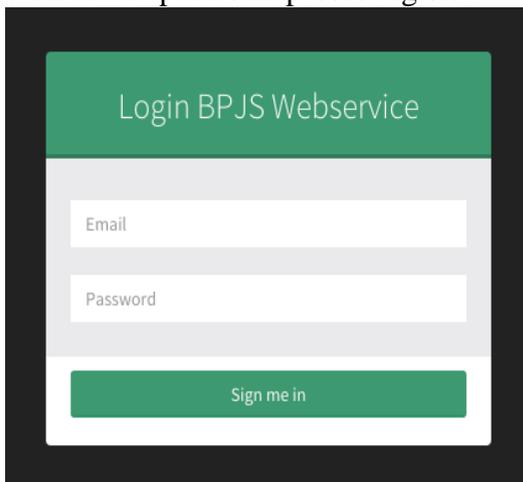
3. Implementasi

Sistem yang telah diimplementasikan dengan bahasa pemrograman PHP dan MySQL. Hasil implementasi sebagai berikut:

a. Implementasi Sistem BPJS

- Halaman Login

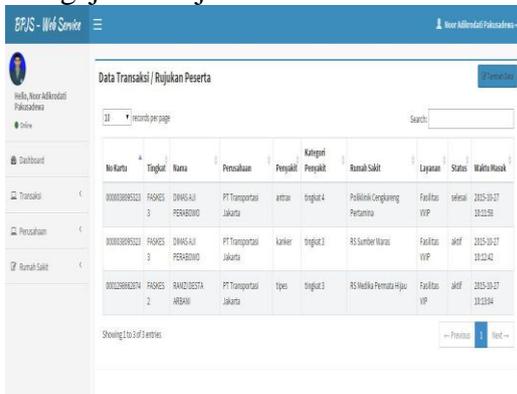
Halaman ini digunakan pengguna untuk login kedalam sistem. Pengguna memasukan email dan password yang didaftarkan pada saat proses register.



Gambar 5.1 : Halaman Login

- Halaman Dashboard

Setelah login sistem akan mengarahkan pengguna pada halaman dashboard. Terdapat menu antara lain dashboard, transaksi, perusahaan dan rumah sakit. Pada halaman dashboard terdapat data transaksi yang ditampilkan. Data transaksi berisi data peserta yang telah mengajukan rujukan.



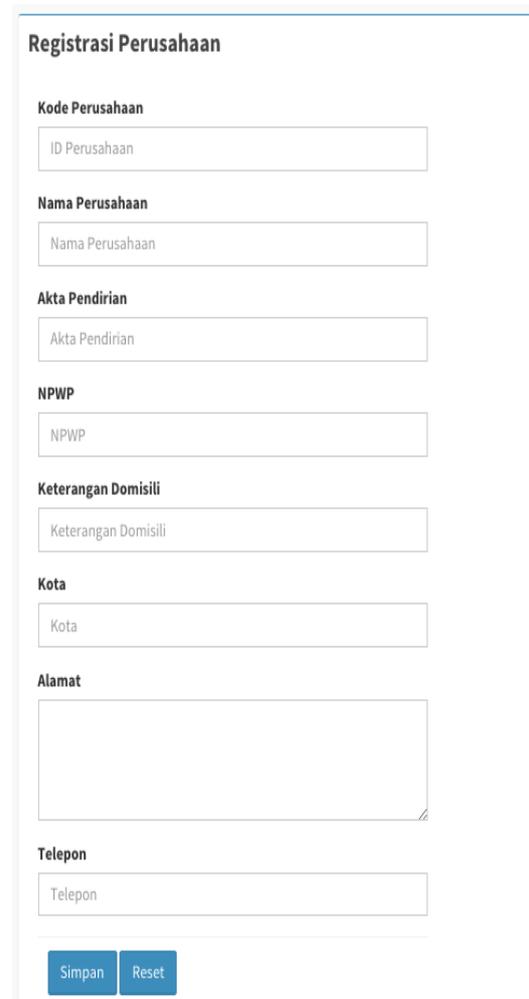
Gambar 5.2: Halaman Dashboard

- Halaman Perusahaan

Halaman ini digunakan admin BPJS untuk mengelola data perusahaan yang terdaftar. Admin dapat menambahkan perusahaan baru yang mendaftar pada BPJS.



Gambar 5.3 : Halaman Perusahaan



Gambar 5.4 : Halaman Input Perusahaan

- Halaman Peserta

Halaman ini digunakan admin untuk melihat data peserta yang terdaftar sebagai peserta BPJS atas nama instansi yang telah terdaftar.

Daftar Peserta

10 records per page

No. Kartu	NIK	Nama	Status Peserta	Nama Instansi	Tingkat	Tanggal Registrasi
0000036110961	3175012302010008	DONI PANCA SAPUTRA	aktif	PT Transporasi Jakarta	FASKES1	2015-10-23 02:37:17
0000037624288	3171071503880005	ARBI ACHYAR	aktif	PT Transporasi Jakarta	FASKES1	2015-10-23 02:37:17
0000037855878	3175070306900008	MOFA YUDA	aktif	PT Transporasi Jakarta	FASKES3	2015-10-23 02:37:17
0000038095323	3276021502020005	DIMAS ALI PERABOWO	aktif	PT Transporasi Jakarta	FASKES3	2015-10-23 02:37:17
0000039790001	3175072701920007	HERU RISWANTO	aktif	PT Transporasi Jakarta	FASKES1	2015-10-23 02:37:17
0000040537899	3275081201920004	SALASA SOLVI MUJI RAHARJA	aktif	PT Transporasi Jakarta	FASKES1	2015-10-23 02:37:17
000004284501	3175092707030002	TRI YULIANTO S	aktif	PT Transporasi Jakarta	FASKES1	2015-10-23 02:37:17
000004248812	3276041804910005	ARYO PAUNGKAS TRANSGIORD	aktif	PT Transporasi Jakarta	FASKES1	2015-10-23 02:37:17
0000043618803	3271052103860006	MUHAMMAD AWANTO WAGIMAN	aktif	PT Transporasi Jakarta	FASKES3	2015-10-23 02:37:17
000004992923	3175056208940002	AGNIS NUR KARTIKA SARI	aktif	PT Transporasi Jakarta	FASKES2	2015-10-23 02:37:17

Showing 1 to 10 of 28 entries

Gambar 5.5: Daftar Peserta

- Halaman Rumah Sakit

Halaman ini digunakan admin untuk mengelola data instansi kesehatan yang terdaftar pada BPJS. Instansi kesehatan berupa rumah sakit atau poliklinik.

Daftar Rumah Sakit

Nama	Tingkat	Jumlah Kamar	Kamar Tersedia	Alamat	Telepon	Aksi
Klinik IMAT Kallides	Tingkat	200	200	Jl. Pen. Setan Raya Haldere Indah I Blok A No. 2	948027	[Edit] [Hapus]
Publikasi Canggih Permana	Tingkat	200	140	Jl. Dan Raga Prens I	7802536	[Edit] [Hapus]
RS Medika Permata Husa	Tingkat	200	0	Jl. Raya Kembangan Lama No. 64 Jakarta	830528	[Edit] [Hapus]
RS IM Thonni Canggih	Tingkat	200	98	Jl. Dan Raga Prens I, Canggih	634821	[Edit] [Hapus]
RS Paka Pekanbaru	Tingkat	200	0	Jl. A.S. Tahan No. 52-54	530802	[Edit] [Hapus]
RS Puri Mandiri Medika	Tingkat	200	0	Jl. Medya Raya No. 2 Medan Jarak	830289	[Edit] [Hapus]
RS Raga Utama	Publikasi	2	2	Jl. Dan Raga Prens I, 34 Sengul	8898319	[Edit] [Hapus]
RS Sitan Saha Medika	Publikasi	2	1	Jl. Raya Pajajaran No. 1 Medan Jarak Jakarta Barat	830888	[Edit] [Hapus]
RS Sumber Wisesa	Tingkat	800	99	Jl. Raya Tugu No. 1 Sengul Jakarta Barat	8802212	[Edit] [Hapus]
RSI Henna Dan Raga	Tingkat	200	3	Jl. Kiriwani Raya No. 12 Perumahan Dan Raga Sitan Jakarta Barat	949089	[Edit] [Hapus]

Showing 1 to 10 of 26 entries

Gambar 5.6 : Halaman Rumah Sakit

Input Rumah Sakit

Nama

Nama RS

Tingkat

Poliklinik

Jumlah Kamar

Jumlah Kamar

Kamar Tersedia

Kamar Tersedia

Alamat

Telepon

Telepon

Simpan Reset

Gambar 5.7: Input Rumah Sakit

b. Implementasi Web Service

- Service Register Peserta

Tabel 5.1: Dokumentasi API Register

Nama	Register (POST)	
url	api/postRegister/	
paramter		
field	type data	deskripsi
		nomor induk
Nik	String / Text	kependudukan peserta
nama	String / Text	nama peserta
tingkat	String / Text	tingkat (FASKES1, FASKES2, FASKES3)
id_instansi	String / Text	id instansi / perusahaan

• Contoh Request Register

Gambar 5.8: Contoh Request Register

• Hasil Request Register (Success)

Gambar 5.9: Hasil Request Register

• Service Daftar Peserta

Tabel 5.2: Dokumentasi API Peserta Terdaftar

nama	GetPeserta (GET)	
url	api/getPeserta/{id_instansi}	
paramter		
field	tipe data	deskripsi
id_instansi	String / Text	id instansi / perusahaan

Gambar 5.10: Contoh Request Peserta

• Hasil Request (Success)

Gambar 5.11: Hasil Request Success

• Hasil Request (Failed)

Gambar 5.12: Hasil Request Gagal

• Service Request Rujukan

Tabel 5.3: Dokumentasi API Rujukan

nama	getRujukan (POST)	
url	api/getRujukan/	
paramter		
1.		
field	tipe data	deskripsi

id_intansi	String / Text	id instansi / perusahaan
no_kartu	String / Text	nomor kartu peserta
jenis_penyakit	String / Text	jenis penyakit peserta(tingkat 1, tingkat 2, tingkat 3, tingkat 4)
penyakit	String / Text	penyakit yang diidap

• Contoh Request Rujukan

Normal Basic Auth Digest Auth OAuth 1.0 No environment

http://localhost:8080/web-service/api/getRujukan POST URL params Headers (0)

form-data x-www-form-urlencoded raw

id_instansi P121314 Text

no_kartu 000112347773 Text

jenis_penyakit tingkat 3 Text

penyakit paru Text

Key Value Text

Send Preview Add to collection Reset

Gambar 5.13: Contoh Request Rujukan

• Hasil Request Rujukan (Success)

Normal Basic Auth Digest Auth OAuth 1.0 No environment

http://localhost:8080/web-service/api/getRujukan POST URL params Headers (0)

form-data x-www-form-urlencoded raw

id_instansi P121314 Text

no_kartu 000112347773 Text

jenis_penyakit tingkat 3 Text

penyakit paru Text

Key Value Text

Send Preview Add to collection Reset

Body Cookies (1) Headers (1) STATUS 200 OK TIME 168 ms

Pretty Raw Preview JSON XML

```

1 {
2   "response": "OK",
3   "status_code": 200,
4   "message": "get data success.",
5   "result": {
6     "nama_peserta": "TIAS PRATIWI",
7     "nik": "327504448870011",
8     "tingkat": "FASIKES 3",
9     "status_peserta": "aktif",
10    "kode_rujukan": "RUCP-20-10-15",
11    "tanggal": "20-10-15",
12    "rumah_sakit": "RS Sumber Haras",
13    "alamat": "Jl. Kyai Tapa No. 1 Grogol Jakarta Barat",
14    "fasilitas": "Fasilitas VIP"
15  }
16 }

```

Gambar 5.14: Contoh Hasil Request Rujukan

• Hasil Request Rujukan (Failed)

Normal Basic Auth Digest Auth OAuth 1.0 No environment

http://localhost:8080/web-service/api/getRujukan POST URL params Headers (0)

form-data x-www-form-urlencoded raw

id_instansi P121314 Text

no_kartu 000112347773 Text

jenis_penyakit tingkat 4 Text

penyakit paru Text

Key Value Text

Send Preview Add to collection Reset

Body Cookies (1) Headers (1) STATUS 200 OK TIME 94 ms

Pretty Raw Preview JSON XML

```

1 {
2   "response": "FASIKES",
3   "status_code": 400,
4   "message": "Peserta masih dalam masa perawatan"
5 }

```

Gambar 5.14: Contoh Request Rujukan (Gagal)

• Hasil Request Rujukan (Failed)

Normal Basic Auth Digest Auth OAuth 1.0 No environment

http://localhost:8080/web-service/api/getRujukan POST URL params Headers (0)

form-data x-www-form-urlencoded raw

id_instansi P121314 Text

no_kartu 000112347773 Text

jenis_penyakit tingkat 4 Text

penyakit paru Text

Key Value Text

Send Preview Add to collection Reset

Body Cookies (1) Headers (1) STATUS 200 OK TIME 94 ms

Pretty Raw Preview JSON XML

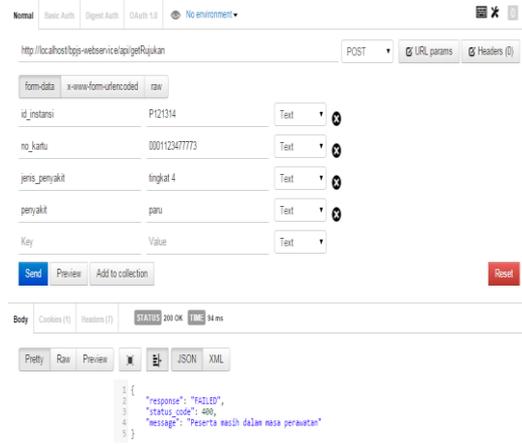
```

1 {
2   "response": "FASIKES",
3   "status_code": 400,
4   "message": "Peserta masih dalam masa perawatan"
5 }

```

Gambar 5.15: Contoh Request Rujukan (Gagal)

● Hasil Request Rujukan (Failed)



Gambar 5.16: Contoh Request Rujukan (Gagal)

4. Pengujian

Metode pengujian yang di gunakan adalah pengujian blackbox. pengujian blackbox dimana pengujian ini dilakukan untuk memastikan tanggapan/respons atas suatu event atau masukan akan menjelaskan proses yang tepat dan menghasilkan keluaran/output sesuai dengan rancangan yang telah dibuat. Uji ini dilakukan pada menu utama dengan hasil sebagai berikut::

Tabel 5.1 Pengujian Blackbox

Input Event	Output	Hasil
Klik tombol Login (masukan data username dan password Valid)	Menampilkan Login Sukses dan menampilkan Form Menu Utama	Sesuai / OK
Klik tombol Login (ketika data username atau password salah)	Menampilkan pesan Login salah.	Sesuai / OK
Input data perusahaan	Menampilkan data perusahaan	Sesuai / OK

Input data rumah sakit	Menampilkan data rumah sakit	Sesuai / OK
Klik Menu peserta	Menampilkan daftar peserta	Sesuai / OK
Klik menu transaksi	Menampilkan daftar transaksi / rujukan	Sesuai / OK
Request register peserta	Berhasil register peserta	Sesuai / OK
Request data peserta	Response menampilkan daftar peserta	Sesuai / OK
Request Rujukan Peserta	Response menampilkan data rujukan	Sesuai/ OK

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan pengujian dalam metode Service Oriented Architecture dengan web service, pada akhir laporan penulis dapat memberikan kesimpulan sebagai berikut :

1. Penelitian ini berhasil merancang dan membangun perangkat lunak yang dapat menjadi jembatan bagi PT. Transportasi dan perusahaan-perusahaan yang terdaftar pada BPJS dalam pengelolaan data asuransi karyawan.
2. Metode Service Oriented Architecture berhasil diimplementasikan pada sistem BPJS. Dengan demikian komunikasi data antara perusahaan-perusahaan dapat dengan mudah dilakukan dan mempercepat proses dalam perujukan asuransi karyawan perusahaan.

Berdasarkan kesimpulan dan analisis laporan, saran dari peneliti untuk penelitian lebih lanjut yaitu :

1. Penelitian lebih lanjut dapat menerapkan sistem keamanan dalam proses request pada web service. Karena sistem berbasis SOA perlu adanya keamanan pada service yang disediakan. Karena service yang disediakan dapat dikonsumsi oleh pihak diluar BPJS, maka perlu adanya sistem keamanan untuk menjaga agar service hanya dapat digunakan oleh pihak yang mempunyai otorisasi.
2. Dikembangkan penelitian yang lebih mendalam dalam penyediaan service dalam Service Oriented Architecture seperti penambahan service yang lebih lengkap dan penggunaan format hasil respon yang terstandarisasi.

2010.

- [5] a. kadir, Konsep Elemen Dasar Sistem., 2003.
- [6] S. .. A.S Rossa, Rekayasa Perangkat Lunak, Bandung: Modula, 2011.
- [8] P. Group, "PHP: Hypertext preprocessor," 2006. [Online]. Available: <http://php.net/>.
- [7] L. Welling dan L. Thomson, PHP and MySQL Web development, Sams Publishing, 2003.
- [9] A. Sunyoto, Ajax Membangun Web dengan Teknologi Asynchronous JavaScript&XML, Penerbit Andi, 2007.
- [10] Jogyanto, Analisis dan Desain Sistem Informasi pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis, dan E-Commerce, Yogyakarta: Andi Offset, 2005.

DAFTAR PUSTAKA

- [1] Jogyanto, Konsep Dasar Sistem, Yogyakarta, 2005.
- [2] T. Erl, Soa: principles of service design, Upper Saddle River: Prentice Hall, 2008.
- [3] I. Y. S. T. S, "Perancangan Basis Data dan Layanan Akses Berbasis Service Oriented Architecture (SOA) Untuk Dinas Kesehatan Kabupaten Sleman," dalam *Jurnal Buana Informatika*, Jogjakarta, 2010.
- [4] M. Arif Firmansyah S.Si, "Implementasi Services Oriented Architecture (SOA) dalam Sistem Transaksi Perbankan di Perguruan Tinggi Studi Kasus : Universitas Padjadjaran," *Development Center of Information System and Technology for Education and Management (DCISTEM) Universitas Padjadjaran Bandung*,