

Web Service Processor sebagai Penghubung Sistem Kiosk Medicom dengan SIM RS Kanker Dharmais

Barizana Sukmadewi Arifin, Mauldy Laya

Jurusan Teknik Informatika dan Komputer
Politeknik Negeri Jakarta

Jl. Prof. Dr. G.A Siwabessy, Kampus Baru UI, Beji, Kukusan, Beji, Kota Depok, Jawa Barat, Indonesia
sbarizana@gmail.com , dylaya@gmail.com

Diterima: 16 Agustus 2017. Disetujui: 12 September 2017. Dipublikasikan: November 2017

Abstrak - Badan Penyelenggara Jaminan Sosial (BPJS) merupakan badan penyelenggara jaminan kesehatan untuk masyarakat Indonesia. BPJS difungsikan untuk membantu dan mempermudah masyarakat untuk mendapatkan layanan kesehatan yang sesuai dari penyedia layanan kesehatan. Namun, hal baik ini berdampak pada peningkatan jumlah rata-rata pasien di Rumah Sakit Kanker Dharmais (RSKD) tiap harinya. Sementara penyisipan informasi pasien masih dilakukan secara manual oleh pegawai RSKD. Sistem KIOSK Medicom membantu untuk meningkatkan pelayanan pendaftaran pada RSKD. Pembuatan *web service* ini memungkinkan sistem KIOSK Medicom dapat mengakses dan melakukan pengecekan informasi pasien yang berada dalam SIM RSKD, pengecekan selesainya tiga status episode dari riwayat kegiatan pasien terdahulu, pendaftaran pasien ke dalam poli tujuan dan dokter yang dipilih, pencetakan SEP, dan penyimpanan data pendaftaran pasien. Komponen yang digunakan untuk perancangan *web service* ini adalah bahasa pemrograman Java dengan *framework* Spring, Eclipse arsitektur REST *web service*, Postman, dan *database* SQL Server.

Kata Kunci : bpjs, rumah sakit, *web service*, java, rest.

I. PENDAHULUAN

Pada era globalisasi hampir segala bidang kehidupan manusia telah tersentuh oleh Teknologi Informasi (TI), hingga sampai pada bidang kesehatan. Alasan utama TI merambah bidang ini ialah disebabkan oleh teknologi yang mampu meningkatkan kualitas pelayanan, khususnya dalam pelayanan dari penyedia layanan kesehatan. Badan Penyelenggara Jaminan Sosial (BPJS) merupakan salah satu badan penyelenggara jaminan kesehatan untuk masyarakat Indonesia. BPJS difungsikan untuk membantu dan mempermudah masyarakat untuk mendapatkan layanan kesehatan yang sesuai dari

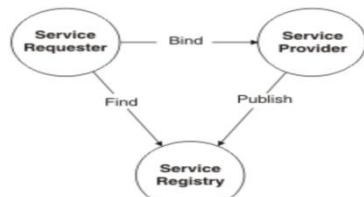
penyedia layanan kesehatan. Rumah sakit sebagai salah satu penyedia layanan kesehatan di Indonesia mendapat pengaruh dengan adanya bantuan jaminan kesehatan BPJS ini. Pengaruh tersebut adalah peningkatan jumlah pasien yang meningkat pesat, tidak terkecuali dengan Rumah Sakit Kanker Dharmais (RSKD). RSKD memiliki rata-rata jumlah pasien 800 sampai dengan 1000 pasien tiap harinya. Pasien-pasien tersebut harus didaftarkan terlebih dahulu sehingga dapat menerima pelayanan kesehatan. Namun, RSKD kewalahan dengan jumlah pendaftar yang ada. Sedangkan pelayanan kurang optimal dan terbilang lamban untuk sistem pendaftaran dan pencetakan Surat Eligibilitas Peserta (SEP) dengan jumlah pasien yang begitu besar. Hal itu disebabkan karena penyisipan informasi pasien masih dilakukan secara manual oleh pegawai RSKD.

Dengan adanya masalah tersebut, sistem KIOSK hadir untuk meningkatkan pelayanan pendaftaran pada RSKD. Sistem KIOSK berada di bawah naungan tim Medicom dari PT. Indo Medika Utama, yang bekerjasama dengan Panasonic Healthcare dari Jepang dan PT. Gobel Dharmas Nusantara. Untuk dapat melakukan pendaftaran melalui sistem KIOSK, diperlukan suatu media komunikasi. Salah satunya dengan penggunaan *web service*. *Web service* dapat membuat sistem KIOSK menjadi terintegrasi dengan Sistem Informasi Manajemen (SIM) RSKD. Pembuatan *web service* ini dibagi menjadi dua bagian, yaitu *web service processor* dan *web service adapter*. Penelitian ini menitikberatkan pada pengerjaan *web service processor*.

II. TINJAUAN PUSTAKA

A. Web Service

Web service merupakan layanan atau aplikasi *client* dan *server*, yang berkomunikasi dan bertukar informasi melalui *World Wide Web (WWW)*. *Client* mengirimkan permintaan melalui *web* dan server menerima permintaan memprosesnya dan mengembalikan sebuah respon. *Web service* menyimpan data informasi dalam format *JSON* atau *XML*, sehingga memungkinkan interaksi pada sistem yang berbeda *platform*, sistem operasi, dan bahasa pemrograman[1]. Berikut arsitektur *web service* secara umum:



Gambar 1. Arsitektur Web Service[1]

Pada Gambar 1 terdapat tiga komponen arsitektur utama dari *web service* yaitu[1]:

- Service provider* sebagai penyedia *web service* yang berfungsi menyediakan kumpulan *web service* yang dapat diakses oleh pengguna.
- Service requestor*, yang merupakan aplikasi yang bertindak sebagai pengguna yang melakukan permintaan layanan (berupa *web service*) ke *service provider*.
- Service registry*, yang merupakan tempat dimana *service provider* mempublikasikan layanannya. Pada arsitektur *Web service*, *Service registry* bersifat opsional.

1. REST Web Service

REST web service merupakan seperangkat prinsip arsitektur yang dapat digunakan untuk merancang *web service* yang berfokus pada sumber daya (*resource*). Sama seperti arsitektur *client/server* web pada umumnya, *request* dan *response* digunakan untuk mengirim *resource* atau data. *REST* bersifat *stateless* dan *resource-oriented*. *REST web service* memiliki 4 prinsip utama [2], yaitu:

- Menggunakan *HTTP method* secara eksplisit. Salah satu karakteristik utama dari layanan *REST* adalah penggunaan *HTTP method* secara eksplisit dengan cara yang mengikuti protokol seperti yang didefinisikan oleh *Request for Comments (RFC)* 2616.
- Stateless* meningkatkan kinerja dan menyederhanakan desain aplikasi *REST web*

service. Hal ini dikarenakan dengan tidak adanya *state server* tidak perlu melakukan sinkronisasi *session data* dengan aplikasi klien.

- Mengekspose struktur direktori. *REST web service* harus memiliki *Uniform Resource Identifier (URI)* yang intuitif dan mudah ditebak. *URI* didefinisikan sebagai semacam dokumentasi diri antarmuka yang memerlukan sedikit, jika ada, penjelasan atau referensi bagi sumber daya yang terkait.
- Bertukar data dengan menggunakan *XML*, *Java Script Object Notation (JSON)* atau keduanya.

2. Postman

Postman adalah sebuah aplikasi (berupa plugin) untuk *browser chrome*, yang berfungsi sebagai *REST Client*, yang digunakan untuk melakukan uji coba *REST API* [3]. *Postman* juga dapat diunduh sebagai aplikasi *desktop*. *Postman* merupakan platform *GUI* yang *powerful* untuk membuat pengembangan *API* lebih cepat dan mudah, mulai dari membangun *API* melalui *testing*, dokumentasi, dan *sharing*. *Postman* direkomendasikan untuk sistem operasi *Mac*, *Windows* atau juga *Linux*.

B. Java

Java adalah sebuah bahasa pemrograman yang diciptakan oleh *James Gosling*, seorang *developer* dari *Sun Microsystem* pada tahun 1991. *Java* memiliki beberapa keunggulan seperti berbasis *GUI (Graphical User Interface)*, aplikasi *web*, *multiplatform*, bisa digunakan pada aplikasi jaringan terdistribusi serta *multithread*[4].

1. Apache Tomcat

Apache Tomcat merupakan sebuah *web server* yang bersifat *open-source* yang mendukung untuk penggunaan spesifikasi *Java EE* antara lain *Java Servlet*, *Java Server Pages (JSP)*, *Java Expression Language*, *WebSocket*, dan menyediakan sebuah *environment HTTP web server* murni *Java* dimana aplikasi berbasis *Java* dapat dijalankan.

2. Spring

Spring merupakan salah satu *framework* milik *Java* yang ringan, berkinerja baik, mudah diuji, dan kode dapat digunakan kembali. *Spring* juga memiliki kerangka kerja *open source*, yang diciptakan oleh *Rod Johnson* untuk menyelesaikan masalah desain sistem dalam pengembangan aplikasi *enterprise*.

3. Eclipse

Eclipse adalah sebuah *IDE (Integrated Development Environment)* untuk mengembangkan perangkat lunak dan dapat dijalankan di semua

platform[5]. Eclipse sering disebut dengan IDE untuk para *developer Java*. Eclipse merupakan salah satu produk yang dikembangkan dan diluncurkan oleh IBM pada tanggal 5 November 2001.

C. *SQL Server*

SQL Server adalah *Relational Data Base Management System (RDBMS)* yang dapat mendukung beragam pemrosesan transaksi, *Business Intelligent (BI)* dan aplikasi analisis di bagian TI dalam perusahaan. Basis data (*data base*) sebagai kumpulan data berisi informasi yang sesuai untuk sebuah perusahaan. Sistem manajemen *database* adalah kumpulan data yang saling berhubungan dan kumpulan program untuk mengakses data[6]. *SQL Server* merupakan salah satu dari tiga *database* terdepan dalam perkembangan pasar saat ini, bersama dengan *Oracle Database* dan *IBM's DB2*. Berikut ini adalah beberapa keunggulan *SQL Server*:

- a. Mampu untuk manajemen *user* dan tiap *user* dapat diatur hak akses terhadap suatu *data base* oleh *data base administrator*.
- b. Mampu untuk membuat *data base mirroring* dan *clustering*.
- c. Tingkat *security* data yang baik.
- d. Mampu untuk *back up* data, *rollback* data, dan *recovery* data.
- e. Cocok untuk perusahaan dengan skala kecil, menengah, dan besar mampu untuk mengolah data dengan jumlah yang besar.

Namun dengan keunggulan tersebut, untuk dapat mengakses pada *SQL Server* pada Bahasa pemrograman *Java* dibutuhkananya *driver*, yaitu *JDBC*. *JDBC* merupakan salah satu API (*Application Programming Interface*) *Java* yang secara khusus ditujukan untuk menangani koneksi ke *data base*[7].

D. *Surat Eligibilitas Peserta (SEP)*

Surat Eligibilitas Peserta (SEP) merupakan surat yang diterbitkan untuk mempermudah peserta *BPJS* memperoleh layanan kesehatan, khususnya pada fasilitas kesehatan (*faskes*) tingkat lanjutan seperti *faskes* tingkat II di Rumah Sakit Umum Daerah, Rumah Sakit Swasta serta Rumah Sakit Umum Pemerintah dan fasilitas III tingkat lanjutan seperti Rumah Sakit Nasional (*RSJP, RSKD dan RSCM*).

SEP harus ada salah satunya sebagai identitas dan tanda bahwa pasien telah diverifikasi status kepesertaannya dan secara administrasi sah sebagai pasien yang dijamin oleh *BPJS Kesehatan* [3]. *SEP* merupakan surat untuk membuktikan bahwa pasien sah secara klaim administrasi. Namun, bila *SEP* sudah terbit, pasien belum tentu dapat lolos verifikasi

klaim medis serta secara verifikasi administrasi. Sehingga *SEP* tidak dapat dijadikan sebagai bukti penjaminan peserta *BPJS*.

E. *UML*

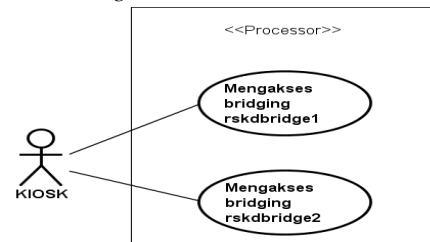
UML adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak [4]. *UML* merupakan bahasa pemodelan untuk sistem atau perangkat lunak yang berhubungan tentang kata “berorientasi objek” untuk menggambarkan model spesifikasi, konstruksi, visualisasi dan dokumentasi bagi alur sistem perangkat lunak yang akan dikembangkan. Beberapa diagram yang digunakan pada penelitian ini adalah : *Use case Diagram, Activity Diagram, Class Diagram, dan Sequence Diagram*.

III. HASIL DAN PEMBAHASAN

A. *Analisa dan Perancangan*

Pada bagian analisa dan perancangan akan dijelaskan melalui diagram-diagram seperti *Use Case Diagram, Activity Diagram, Class Diagram, Sequence Diagram* dan struktur *database* untuk menggambarkan kegiatan dan bisnis proses pada *web service processor*.

1. *Use Case Diagram*



Gambar 2. *Use Case Diagram Web Service Processor*

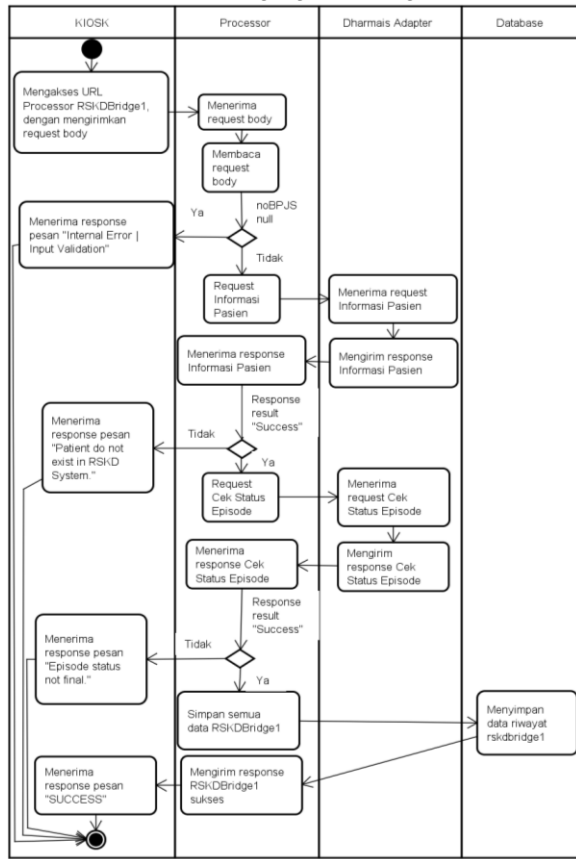
Gambar 6 merupakan gambaran dari kegiatan yang dapat dilakukan oleh aktor sistem *KIOSK Medicom* terhadap *web service processor*. Aktor dapat mengakses *web service processor* dengan cara memanggil URL yang memespesifikasikan proses *bridging*, agar terjadi pembatasan kegiatan sesuai dengan masing-masing fungsi, seperti proses *bridging rskdbridge1* dan *rskdbridge2*. *Web service processor* dapat digunakan sebagai media komunikasi antara sistem *KIOSK Medicom* dengan *web service RSKD* dan *web service BPJS* untuk bertukar informasi.

2. *Activity Diagram*

Berikut adalah dua diagram aktivitas yang menggambarkan kegiatan dari *use case* di atas. Dua diagram aktivitas berikut terdiri dari dua proses

kegiatan *bridging* yang dapat dilakukan oleh aktor, yaitu proses *bridging rskdbridge1* proses *bridging* dan *rskdbridge2* :

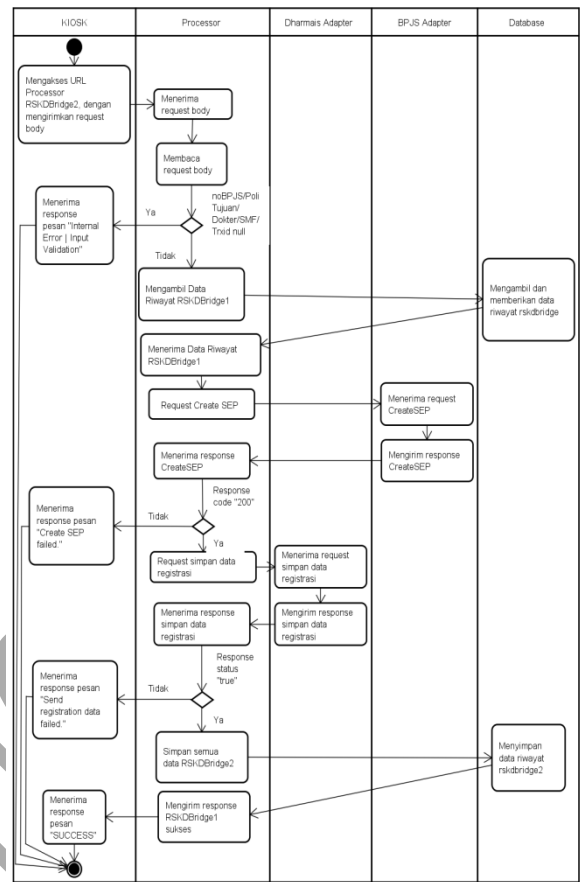
a. Aktivitas dalam *bridging rskdbridge1* :



Gambar 3. Diagram Aktivitas Web Service Processor Bridging Rskdbridge1

Gambar 7 menjelaskan proses *rskdbridge1* yang merupakan tahap awal dalam proses *bridging* pada *web service processor*. Dimulai dengan sistem KIOSK memberikan *request body* kepada *web service processor* yang akan diperiksa apakah *request* yang diberikan dapat dilanjutkan ke kegiatan berikutnya atau berhenti karena tidak mendapatkan respon yang seharusnya. *Web service processor* berinteraksi dengan *web service Dharmais adapter* untuk dapat melakukan *request* dan mendapatkan respon untuk mendapatkan informasi pasien RSKD dan cek status episode. Jika seluruh respon sesuai dengan yang diharapkan, maka seluruh data yang ada di dalam *rskdbridge1* akan disimpan di dalam database dengan nama tabel *rskdbridge1 history*.

b. Aktivitas dalam *bridging rskdbridge2* :



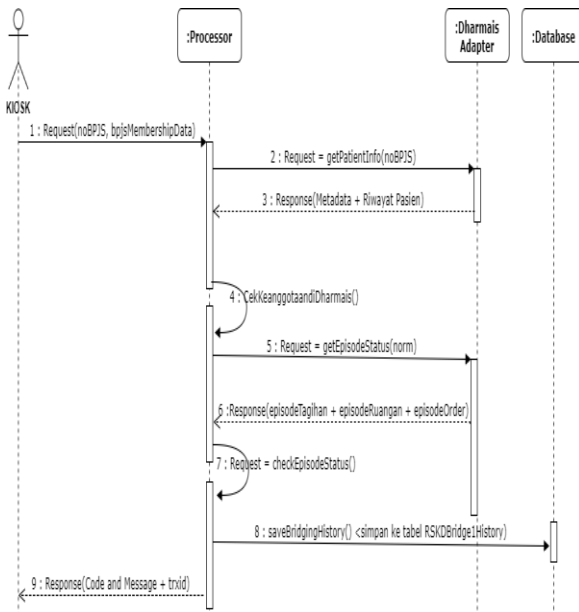
Gambar 4. Diagram Aktivitas Web Service Processor Bridging Rskdbridge2

Gambar 8 menjelaskan proses bisnis untuk proses *bridging rskdbridge2*, KIOSK Medicom melakukan *request* untuk mendapatkan riwayat data dari proses *bridging rskdbridge1*, pembuatan SEP, melakukan *request* untuk mengirimkan data registrasi ke RSKD, melakukan validasi respon, hingga menyimpan data ke dalam database. *Rskdbridge2* memerlukan bantuan dari *web service Dharmais adapter* dan *web service BPJS adapter* untuk mencetak SEP. Dan terakhir akan disimpan di dalam tabel *rskdbridge2 history*.

3. Sequence Diagram Web Service Processor

Berikut adalah dua *sequence diagram* yang terdiri dari dua proses kegiatan *bridging* yang dapat dilakukan oleh aktor, yaitu *rskdbridge1* dan *rskdbridge2* :

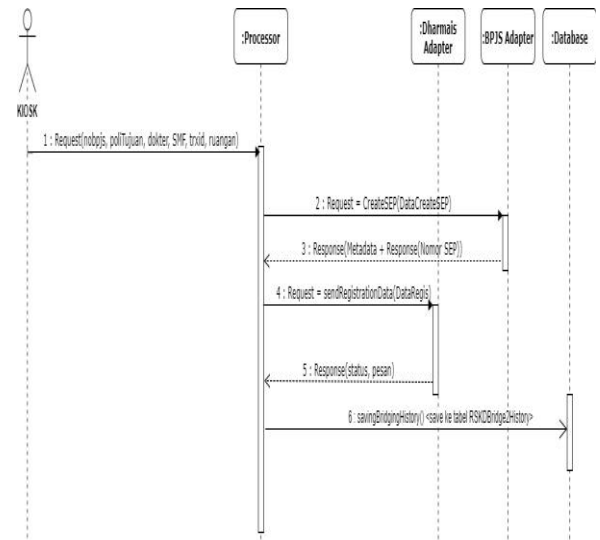
a. Alur dalam bridging rskdbridge1 :



Gambar 95. Sequence Diagram Web Service Processor Bridging Rskdbridge1

Gambar 9 menjelaskan proses bisnis untuk *bridging rskdbridge1*. Di mulai dari sistem KIOSK Medicom mengirimkan *request* dengan parameter noBPJS dan bpjsMembershipData yang dikirimkan kepada *web service processor*. Kemudian *web service processor* melakukan *request* kepada *web service Dharmais adapter*, untuk mendapatkan informasi pasien Dharmais lalu disaring respon yang didapat apakah dapat dilanjutkan ke tahap berikutnya, bila berhasil disaring akan dilakukan *request* kembali untuk mendapatkan tiga status episode, lalu respon kembali disaring. Sampai dengan proses tersebut, *web service processor* akan melakukan fungsi penyimpanan data untuk proses *bridging rskdbridge1* sebagai tanda bahwa proses *bridging* ini selesai dengan sukses. Data yang disimpan pada proses ini akan dijadikan informasi bagi proses *bridging rskdbridge2*, yaitu *trxid*.

b. Alur dalam bridging rskdbridge2 :



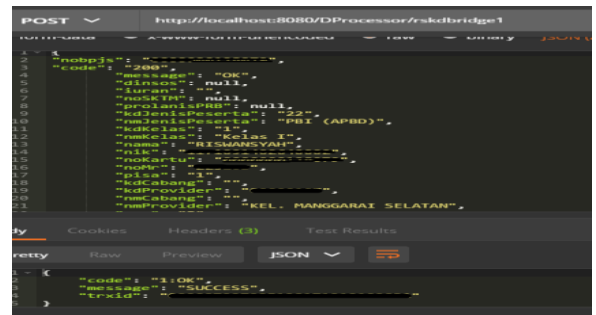
Gambar 60. Sequence Diagram Web Service Processor Bridging Rskdbridge2

Gambar 10 menjelaskan proses bisnis untuk *bridging rskdbridge2*, sistem KIOSK Medicom mengirimkan *request* dengan parameter nobpjs, poliTujuan, dokter, SMF, trxid, ruangan. *Webservice Dharmais adapter* dan *BPJS adapter* yang menerima *request* dan mengirimkan kembali respon untuk *web service processor* dan menyimpannya ke dalam tabel *bridging rskdbridge2 history*, sehingga dapat dinyatakan bahwa pendaftaran ke poli tujuan dan cetak SEP telah berhasil.

B. Implementasi Web Service Processor

Berikut adalah implementasi dari proses-proses yang dilakukan oleh *web service processor*:

a. Operasi *bridging rskdbridge*

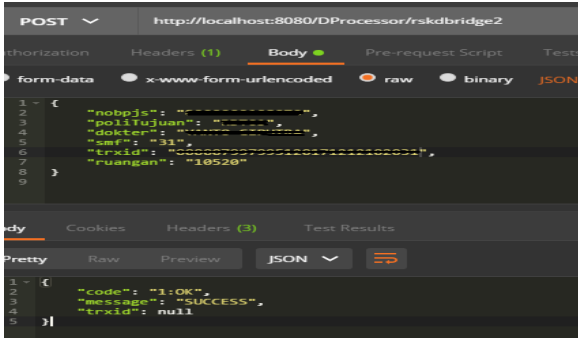


Gambar 11. Operasi Pemanggilan Bridging Rskdbridge1

Gambar 11 dari hasil pemanggilan fungsi proses *bridging rskdbridge1* pada *web service*

processor yang berfungsi untuk menghasilkan cek kepesertaan di dalam SIM RSKD, cek episode apakah sudah pasien sudah selesai melakukan episode order, episode kunjungan, dan episode ruangan. Hasil dari proses ini adalah *code*, *message*, dan *trxid*.

b. Operasi *bridging rskdbridge2*



Gambar 12. Operasi Pemanggilan Bridging *Rskdbridge2*

Gambar 12 merupakan hasil pemanggilan dari fungsi *bridging rskdbridge2* pada *web service processor* yang berfungsi untuk menentukan poli tujuan dan dokter. Hasil dari proses ini adalah *code*, *message*, dan *trxid*.

C. Pengujian Web Service Processor

Pengujian yang dilakukan menggunakan *test case* untuk mengetahui bagaimana hasil yang akan ditampilkan bila operasi *bridging* sesuai atau tidak sesuai dengan yang diinginkan. Format laporan pengujian disesuaikan dengan *template* yang telah dibuat sebagai panduan *web service* Medicom untuk PHC. Pengujian dilakukan terhadap *requirement* RSKD. Berikut adalah hasil pengujian dari *rskdbridge1* dan *rskdbridge2*:

a. *Rskdbridge1*

URL : <IP address>/DProcessor/*rskdbridge1*
 Fungsi : Mendapatkan data informasi pasien RSKD, mengecek status episode dan simpan data *bridging rskdbridge1* .

Method : POST

Format : JSON

TABEL 1. TEST CASE OPERASI BRIDGING RSKDBRIDGE1

No.	Test Case	Request Parameter	Response Message	Type
1.	<i>Rskd bridge1</i>	{ "nobpjs": "000xxxxxxxxxx", "code": "200", "message": "OK", "dinsos": null, "iuran": "", "noSKTM": null, "prolanisPRB": null, "kdJenisPeserta": "22", "nmJenisPeserta": "PBI (APBD)", "kdKelas": "3", "nmKelas": "KELAS III", "nama": "TRI ARNI", "nik": "xxxxxxxxxxxxxxxx", "noKartu": "000xxxxxxxxxx", "noMr": "151515", "pisa": "3", "kdCabang": "", "kdProvider": "09020107", "nmCabang": "", "nmProvider": "KEL. MANGGARAI SELATAN", "sex": "P", "keterangan": "AKTIF", "kode": "0", "tglCetakKartu": "2014-12-17", "tglLahir": "1965-09-29", "tglIAT": "2050-12-31", "tglTMT": "2014-01-09", "umurSaatPelayanan": "", "umurSekarang": "51 tahun ,11 bulan ,14 hari" }	{ code : "1:OK", message : "Success", trx_id: "00000799799512017" }	Normal test
2.	Chekk dhar mais mem bersh ip	{ "nobpjs": "000XXXXXXXXXXXXX", "code": "200", "message": "OK", "dinsos": null, "iuran": "", "noSKTM": null, "prolanisPRB": null, "kdJenisPeserta": "22", "nmJenisPeserta": "PBI (APBD)", "kdKelas": "3", "nmKelas": "KELAS III", "nama": "TRI ARNI", "nik": "xxxxxxxxxxxxxxxx", "noKartu": "000xxxxxxxxxx", "noMr": "151515",	{code : "0:101", message : "Patient does not exist", trx_id: "null" }	Negative test

		"pisa": "3", "kdCabang": "", "kdProvider": "09020107", "nmCabang": "", "nmProvider": "KEL MANGGARAI SELATAN", "sex": "P", "keterangan": "AKTIF", "kode": "0", "tglCetakKartu": "2014-12-17", "tglLahir": "1965-09-29", "tglTAT": "2050-12-31", "tglTMT": "2014-01-09", "umurSaatPelayanan": "", "umurSekarang": "51 tahun ,11 bulan ,14 hari"		
3.	Check episode status	{"nobpjs": "000XXXXXXXXXX", "code": "200", "message": "OK", "dinsos": null, "iuran": "", "noSKTM": null, "prolanisPRB": null, "kdJenisPeserta": "22", "nmJenisPeserta": "PBI (APBD)", "kdKelas": "3", "nmKelas": "KELAS III", "nama": "TRARNI", "nik": "XXXXXXXXXXXXXXXXXX", "noKartu": "000XXXXXXXXXX", "noMr": "151515", "pisa": "3", "kdCabang": "", "kdProvider": "09020107", "nmCabang": "", "nmProvider": "KEL MANGGARAI SELATAN", "sex": "P", "keterangan": "AKTIF", "kode": "0", "tglCetakKartu": "2014-12-17", "tglLahir": "1965-09-29", "tglTAT": "2050-12-31", "tglTMT": "2014-01-09", "umurSaatPelayanan": "", "umurSekarang": "51 tahun ,11 bulan ,14 hari" }	Negative test	

Method : POST
Format : JSON

TABEL 2. TEST CASE OPERASI BRIDGING RSKDBRIDGE2

No	Test Case	Request Parameter	Response Message	Type
1.	Rskdbridge2	{"nobpjs": "000XXXXXXXXXX", "poliTujuan": "umum", "dokter": "YANTO CIPUTRA", "trxid": "00000799799512017" }	{code : "1:OK", message : "Success", }	Normal test
2.	Create SEP	{"nobpjs": "000XXXXXXXXXX", "poliTujuan": "umum", "dokter": "YANTO CIPUTRA", "trxid": "00000799799512017" }	{code : "0:101", message : "Create SEP failed", }	Negative test
3.	Send registration data	{"nobpjs": "000XXXXXXXXXX", "poliTujuan": "umum", "dokter": "YANTO CIPUTRA", "trxid": "00000799799512017" }	{code : "0:102", message : "Send registration data failed.", }	Negative test

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Kesimpulan dari rancang bangun *web service processor* untuk sistem KIOSK Medicom. *Web service* ini dapat menjadi penghubung antara sistem KIOSK Medicom dengan SIM RSKD. Sehingga KIOSK Medicom dapat mengakses dan melakukan pengecekan informasi pasien yang berada dalam SIM RSKD, pengecekan penyelesaian tiga status episode dari riwayat kegiatan pasien terdahulu, pendaftaran pasien ke dalam poli tujuan dan dokter yang dipilih, pencetakan SEP, dan penyimpanan data pendaftaran pasien.

b. Rskdbridge2

URL : <IP address>/DProcessor/rskdbridge2

Fungsi : Membuat SEP, mengirimkan data registrasi ke *web service* RSKD dan simpan data *bridging rskdbridge2*.

B. Saran

Sebaiknya dalam pengembangan proyek diketahui *user requirement* yang matang secara keseluruhan, sehingga dapat mengurangi waktu dalam implementasi dan tidak terjadi penambahan atau perubahan pengerjaan proyek.

REFERENSI

- [1] Anonim. 2017. SOAP *Web Services Architecture*. https://www.ibm.com/support/knowledgecenter/en/SSGMCP_5.2.0/com.ibm.cics.ts.webservices.doc/concepts/dfhws_model.html. [21 Desember 2017].
- [2] Taufik, R. dan Utomo, V.G. 2014. "Rancang Bangun Aplikasi Mobile Untuk Notifikasi Jadwal Kuliah Berbasis Android (Studi Kasus: Stmik Provisi Semarang)". *Jurnal Teknologi Informasi dan Komunikasi*, 48, 2087-0868.
- [3] Lewiani, Nia, Lisnawaty dan Akifah. 2017. "Proses Pengelolaan Klaim Pasien BPJS Unit Rawat Inap Rumah Sakit Dr. R. Ismoyo Kota Kendari Tahun 2016". *Jurnal Ilmiah Mahasiswa Kesehatan Masyarakat*, 4, 250-731.
- [4] Mallu, Satriawaty. 2015. "Sistem Pendukung Keputusan Penentuan Karyawan Kontrak Menjadi Karyawan Tetap Menggunakan Metode Topsis". *Jurnal Ilmiah Teknologi Informasi Terapan*, 38, 2407-3911.
- [5] Silvia, A.F., Haritman, E., dan Muladi, Y. 2014. "Rancang Bangun Akses Kontrol Pintu Gerbang Berbasis Arduino dan Android". *ELECTRANS*, 2, 1412 – 3762.
- [6] Daminto, E.P. dan Syahriani. 2014. *Perancangan Aplikasi Portal Berita E-News Update Berbasis Mobile* [Dikutip dari Analisis Dan Perancangan Sistem Basis Data Pada PT Jevin Multi Sarana. Penulisan Ilmiah Fakultas Ilmu Komputer Universitas Binadama Palembang]. Makalah dipresentasikan pada SNIPTEK 2014.
- [7] Setyorini dan Adha, F.A. 2017. "Sistem Informasi Perpustakaan E-Library Stmik Asia Malang Berbasis Java Server Page". *Jurnal Teknologi Infomasi*, 12, 2086-2989.
- [8] Imbar, R.V. dan Bratawijaya, P.I.K. 2017. "Sistem Informasi Rekrutmen dan Rekomendasi Penilaian Pelamar Menggunakan *ProfileMatching*". *JUISI (Vol. 03)*, 12-13, 2460-1306.
- [9] Widanti, Puspa., 2015. Sistem Informasi Laporan Keuangan Penjualan Barang pada CV Vanda Jaya Abadi Tangerang [Online] Available at: <https://widuri.raharja.info/index.php/TA1123368830> [21 Desember 2017].
- [10] Novita, R. dan Sari, N. 2015. "Sistem Informasi Penjualan Pupuk Berbasis E-Commerce". *Jurnal TEKNOIF*, 3-4, 2338-2724.
- [11] *Object Management Group*. 2011. *OMG Unified Modelling Language (OMG UML)*, Superstructure. U.S: Author.