

**LAPORAN TAHUN TERAKHIR
PENELITIAN PRODUK TERAPAN (PPT)**



**DESAIN MODEL PROTOTYPE DATA WAREHOUSE
SEBAGAI INTEGRASI DATA TERSEBAR UNTUK
MENDUKUNG SISTEM MONITORING DAN ANALISA
DATA SURVEILANS KESEHATAN MASYARAKAT
KABUPATEN GROBOGAN**

Tahun ke 2 dari rencana 2 tahun

Fikri Budiman, M.Kom (0604047201)

Muslih, M.Kom (0604057501)

**UNIVERSITAS DIAN NUSWANTORO SEMARANG
NOVEMBER - 2016**

Perpustakaan Universitas
Dian Nuswantoro

2016
FIK
D
10058

HALAMAN PENGESAHAN

Judul : DESAIN MODEL PROTOTYPE DATA WAREHOUSE
SEBAGI INTEGRASI DATA TERSEBAR UNTUK
MENDUKUNG SISTEM MONITORING DAN
ANALISA DATA SURVEILANS KESEHATAN
MASYARAKAT KABUPATEN GROBOGAN

Peneliti/Pelaksana

Nama Lengkap : FIKRI BUDIMAN M.KOM
Perguruan Tinggi : Universitas Dian Nuswantoro
NIDN : 0604047201
Jabatan Fungsional : Lektor Kepala
Program Studi : Teknik Informatika
Nomor HP : 08122804229
Alamat surel (e-mail) : fikri@dosen.dinus.ac.id

Anggota (1)

Nama Lengkap : MUSLIH M.Kom
NIDN : 0604057501
Perguruan Tinggi : Universitas Dian Nuswantoro
Institusi Mitra (jika ada) :
Nama Institusi Mitra : -
Alamat : -
Penanggung Jawab : -
Tahun Pelaksanaan : Tahun ke 2 dari rencana 2 tahun
Biaya Tahun Berjalan : Rp 50.000.000,00
Biaya Keseluruhan : Rp 107.173.000,00



Semarang, 21/11 - 2016
Kritca

(FIKRI BUDIMAN M.KOM)
NIP/NIK 0686.11.1995.070

Menyetujui,
Kepala Pusat Penelitian

(Prof. Vincent Didiek W.A. Ph.D)
NIP/NIK 0686.11.2014.606

RINGKASAN

Pengendalian dan pemberantasan penyakit memerlukan pengelolaan data kesehatan dalam bentuk surveilans epidemiologi yang baik. Kegiatan surveilans epidemiologi meliputi pengumpulan, pengolahan, analisis dan interpretasi data epidemiologi serta penyajian informasi. Untuk itu diperlukan data epidemiologi yang akurat, lengkap dan tepat waktu. Yang menjadi permasalahan sistem surveilans saat ini adalah tidak tersedianya sistem manajemen database yang dapat digunakan sebagai sarana koleksi dan integrasi data kesehatan. Data yang dikirim dari unit surveilans (puskesmas, rumah sakit) ke tingkat kabupaten/kota (dinas kesehatan) tidak lengkap, tidak akurat dan tidak tepat waktu. Selain permasalahan tersebut, data yang dikirim tidak memiliki standar yang sama dan bersumber dari sistem aplikaswi yang berbeda (*heterogen*). Secara teknis masalah yang dihadapi adalah *interoperability* atau kemampuan untuk mengintegrasikan data yang bersumber dari aplikasi yang berbeda *platform*. Untuk mengatasi permasalahan tersebut diperlukan metode XML Web Services, yaitu metode yang dapat mengintegrasikan aplikasi dan pertukaran data dalam format XML (Extensible Markup Language). Pertukaran data dalam format XML menggunakan teknologi SOAP (Simple Object Access Protocol) dan WSDL (Web Services Description Language). Web Services yang akan menjembatani perbedaan *platform* aplikasi dalam rangka integrasi ke data center.

Tujuan penelitian ini adalah untuk mengidentifikasi kebutuhan data, sumber data, mekanisme koleksi data (replikasi), desain model integrasi database dalam bentuk prototype *data warehouse* (gudang data) untuk mendukung sistem monitoring dan analisa data kesehatan. Pada desain model integrasi database dalam bentuk prototype *data warehouse*, pengelolaan sumber koleksi data yang berasal dari unit pengelola data kesehatan pada suatu wilayah kabupaten/kota, sangat mempengaruhi keberhasilan sistem surveilans epidemiologi. Mekanisme koleksi data antar unit surveilans memerlukan proses strukturisasi, sinkronisasi dan replikasi secara langsung. Dengan demikian data kesehatan akan selalu tersedia secara *up to date*, lengkap dan terintegrasi pada *data warehouse*. Ketersediaan data surveilans ini akan memudahkan sistem monitoring dan analisa kesehatan dalam rangka pengendalian dan pemberantasan penyakit. Dengan demikian penelitian ini memiliki urgensi dalam mendesain *prototype data warehouse* sehingga memiliki kontribusi langsung terhadap proses pengendalian pemberantasan penyakit melalui mekanisme penyediaan data untuk mempermudah monitoring dan analisa data kesehatan. Jenis penelitian ini merupakan penelitian kualitatif dengan metode *action research*. Penelitian dimulai dari studi pendahuluan, analisa kebutuhan, identifikasi masalah dan informasi yang dibutuhkan, merumuskan hipotesis tindakan, perancangan sistem integrasi basis data, desain model *prototype data warehouse*, uji coba dan simulasi *prototype data warehouse*. Hasil dari penelitian yang diperoleh merupakan desain level konseptual dan model *prototype data warehouse* yang dapat digunakan untuk mendukung monitoring dan analisa data kesehatan. Data surveilans epidemiologi dari masing-masing puskesmas dan rumah sakit akan terkumpul dan terintegrasi dalam suatu model prototype data warehouse. *Prototype data warehouse* ini akan dapat digunakan untuk melakukan monitoring dan analisa data kesehatan baik melalui sistem informasi ataupun dengan model *data mining*.

PRAKATA

Alhamdulillah, puji syukur peneliti panjatkan kehadirat Allah SWT yang telah memberikan limpahan rahmat dan hidayahNya sehingga peneliti dapat menyusun dan menyelesaikan laporan tahun terakhir penelitian produk terapan (PPT) ini. Penelitian yang berjudul DESAIN MODEL PROTOTYPE DATA WAREHOUSE SEBAGAI INTEGRASI DATA TERSEBAR UNTUK Mendukung Sistem Monitoring dan Analisa Data Surveilans Kesehatan Masyarakat Kabupaten Grobogan, merupakan penelitian produk terapan (PPT), yang mana skim sebelumnya bernama hibah bersaing yang dibiayai KEMENRISTEK-DIKTI selama dua tahun. Selama melakukan penelitian dan selesainya laporan akhir tahun ke-2 penelitian ini tidak terlepas dari bantuan dan dorongan dari berbagai pihak, baik secara moril dan materiel. Oleh karena itu peneliti mengucapkan terimakasih kepada :

1. Direktur Jendral Penguatan Riset dan Pengembangan Kemenristekdikti
2. Direktorat Riset dan Pengabdian Masyarakat
3. Koodinator Pertungguan Tinggi Swasta (Kopertis) wilayah VI Jawa Tengah
4. Rektor UDINUS Semarang
5. Direktur LP2M UDINUS Semarang
6. Dosen dan mahasiswa Udinus Semarang
7. Para kerabat atau keluarga besar yang tercinta

Meskipun sudah memperhatikan berbagai aspek yang berhubungan dengan penulisan laporan tahun terakhir penelitian produk terapan ini, peneliti menyadari masih banyak terdapat kekurangan dan kelemahan dalam penelitian PPT ini. Saran dan kritik yang bersifat membangun merupakan masukan yang peneliti harapkan. Semoga penelitian ini bermanfaat dan dapat dikembangkan ataupun disempurnakan oleh para peneliti lainnya.

Semarang, 20 November 2016
Ketua Peneliti

Fikri Budiman
NIDN. 0604047201

DAFTAR ISI

Halaman Pengesahan	i
Ringkasan	ii
Prakata	iii
Daftar Isi	iv
Daftar Tabel	vi
Daftar Gambar	vii
Daftar Lampiran	viii
Bab 1 Pendahuluan	1
Bab 2 Tinjauan Pustaka.....	4
2.1. Sinkronisasi.....	6
2.1.1. Pengertian Sinkronisasi.....	6
2.1.2. Protokol Sinkronisasi.....	6
2.2. Replikasi.....	6
2.2.1. Pengertian Replikasi.....	6
2.2.2. Jenis Replikasi.....	6
2.2.3. Teknik Replikasi.....	7
2.3. Basis Data Distribusi.....	7
2.3.1. Definisi Basis Data Distribusi.....	7
2.3.2. Desain Sistem Basis Data Distribusi.....	7
2.3.3. Fragmentasi Sistem Basis Data Distribusi.....	7
2.4. Gudang Data (Data Warehouse).....	9
2.5. XML (Extensible Markup Language)	10
2.6. Web Services.....	10
2.7. SOAP (Simple Object Access Protocol).....	12
2.8. WSDL (Web Services Description Language)	12
2.9. NuSOAP	13
3.0. Integrasi Aplikasi.....	13
Bab 3 Tujuan Dan Manfaat Penelitian.....	17
Bab 4 Metodologi Penelitian	
4.1. Rancangan Penelitian.....	19
4.2. Identifikasi Masalah.....	19
4.3. Merumuskan Hipotesa	20
4.4. Merancang dan Mendesain Prototype.....	20
4.5. Eksperimental Prototype.....	20
4.6. Evaluasi Prototype.....	21
4.7. Proses Berulang.....	21
4.8. Penerimaan Prototype.....	21
Bab 5 Hasil Yang Dicapai	
5.1. Hasil Analisis Dan Identifikasi Masalah	23

5.2. Hasil Identifikasi Desain Integrasi Dengan Web Services	29
5.3. Hasil Rancangan Arsitektur Integrasi Antar Host Unit Surveilans Dengan Dinkes.....	31
5.4. Hasil Rancangan Web Services	37
5.5. Hasil Perancangan Interface Integrasi Unit Survilans Dengan Data Center	35
 Bab 6 Kesimpulan dan Saran.....	 37
 Daftar Pustaka	
Lampiran	

DAFTAR TABEL

Tabel 1. Bagan Penelitian yang sudah dilakukan dan akan dilanjutkan.....	21
Tabel 2. Perancangan <i>web service</i> sistem Dinas Kesehatan.....	34
Tabel 3. Perancangan <i>web service</i> sistem Puskesmas.....	35
Tabel 4. Perancangan <i>web service</i> sistem Rumah sakit.....	36
Tabel 5. Perancangan <i>web service</i> sistem Dinas Kesehatan.....	37

DAFTAR GAMBAR

Gambar 1. Framework Sistem Tersebar.....	6
Gambar 2. Perancangan Secara Top Down.....	6
Gambar 3. Perancangan Secara Button Up.....	7
Gambar 4. Arsitektur Data Warehouse.....	8
Gambar 5. Arsitektur Web Service.....	11
Gambar 6. SOAP dengan standar XML.....	12
Gambar 7. Diagram Document WSDL.....	12
Gambar 8: Model point to point.....	14
Gambar 9. Model hub and spoke.....	15
Gambar 10. Model message bus.....	16
Gambar 11. Diagram Alir Penelitian.....	19
Gambar 12. Rancangan Database.....	24

DAFTAR LAMPIRAN

Lampiran 1. Instrumen

Lampiran 2. Personalia Tenaga Pelaksana Besrta Kualifikasinya

Lampiran 3. Artikel Ilmiah

Lampiran 4. HKI

Lampiran 5. Hasil Coding Servives Produk Penelitian

BAB 1

PENDAHULUAN

Pembangunan dibidang kesehatan merupakan bagian integral dan terpenting dalam pembangunan nasional. Tujuan diselenggarakan pembangunan kesehatan adalah untuk meningkatkan kesadaran, kemauan dan kemampuan hidup sehat bagi setiap orang agar terwujudnya derajat kesehatan masyarakat yang optimal. Hal ini sesuai dengan amanat Undang-Undang Dasar 1945 pasal 28 H ayat (1) bahwa setiap orang berhak hidup sejahtera lahir dan batin, bertempat tinggal dan mendapatkan lingkungan hidup baik dan sehat serta berhak mendapatkan pelayanan kesehatan. Seiring dengan sistem otonomi daerah maka penjabaran tujuan pembangunan nasional yang sesuai dengan amanat UUD 1945 tersebut menjadi tugas pokok pemerintah daerah, dimana parameter keberhasilan pembangunan daerah dapat dilihat dari pencapaian Indeks Pembangunan Manusia (*IPM*). Komponen penting yang mempengaruhi *IPM* adalah indikator kesehatan per kapita. Dengan indikator pencapaian derajat kesehatan perkapita menjadi upaya peningkatan kualitas sumberdaya manusia, sehingga secara tidak langsung akan mendukung percepatan pembangunan nasional.

Indikator kesehatan masyarakat sangat erat kaitanya dengan epidemiologi suatu kasus pada suatu daerah tertentu. Epidemiologi adalah wabah penyakit terutama yang menular secara cepat dan tak terduga pada suatu wilayah tertentu. Agar wabah tidak meluas ekskalasinya maka diperlukan sistem monitoring untuk mengembangkan suatu metode dalam menganalisis secara sistematis keadaan dan keberadaan suatu penyakit dalam upaya untuk mengatasi dan menaggulangi secara cepat dan terintegrasi. Untuk itu Departemen Kesehatan telah mengeluarkan keputusan menteri *No. 1479/MENKES/SK/X/2003* tentang : Pedoman Penyelenggaraan Sistem Surveilans Epidemiologi Penyakit Menular Dan Penyakit Tidak Menular Terpadu. Dalam pedoman surveilans tersebut menegaskan diperlukannya suatu Sistem Surveilans Terpadu (*SST*) dengan dukungan basis data yang setandar dimana sistem pengawasan utama epidemiologi meliputi semua unit pelayanan kesehatan (Puskesmas, Laboratorium, Rumah Sakit) di semua Pemerintah Daerah Propinsi dan Pemerintah Daerah Kabupaten/Kota dengan model : Sistem Pencatatan Pelaporan Puskesmas Terpadu (*SP2PT*) dan Sistem Pelaporan Rumah Sakit (*SPRS*). Di tingkat pemerintah daerah pelaksanaan operasional **SST** tersebut sepenuhnya diserahkan kepada dinas kesehatan daerah untuk bisa menjadi sistem informasi epidemiologi dalam rangka mendukung pemberantasan penyakit menular dan tidak menular secara nasional. Direktorat Jendral Pemberantasan Penyakit Menular dan Penyehatan Lingkungan (Dirjen PPM & PL Departemen Kesehatan) sebagai

lembaga pemerintah pusat yang mendapat tugas dan bertanggung jawab dalam bidang pengendalian maupun pemberantasan penyakit secara nasional.

Namun dalam pelaksanaan dan penyelenggaraan sistem surveilans terpadu (*SST*) tersebut ditingkat kabupaten/kota menghadapi suatu kendala dalam melakukan pengiriman data kesehatan ke dinas kesehatan kabupaten/kota sebagai penanggung jawab kesehatan di tingkat pemerintah daerah. Kendala ini disebabkan karena sumber data unit surveilans (puskesmas, laboratorium, rumahsakit) terletak secara geografis tersebar dan diperoleh dari beragam aplikasi dan database manajemen sistem (*DBMS*) yang beragam (heterogen). Sinkronisasi dan integrasi data antara unit surveilans (puskesmas, laboratorium, rumahsakit) dengan dinas kesehatan kabupaten/kota menjadi masalah utamanya. Hal ini disebabkan karakter masing-masing unit surveilans memiliki ketidak seragaman platform aplikasi dan database (*heterogen*). Untuk itu dibutuhkan suatu manajemen sistem interoperabilitas basis data yang memiliki pola strukturisasi, sinkronisasi dan integrasi data antar unit surveilans (skema local) dengan dinas kesehatan sebagai "*Data Center*" (skema global) dalam model *data warehouse* (gudang data). Model *data warehouse* ini akan dapat menjadi inovasi dan modal teknis dalam pengembangan sistem monitoring dan analisis data kesehatan dalam rangka pengendalian dan pemberantasan penyakit secara cepat dan tepat. Dalam jangka panjang model integrasi data dalam bentuk model *prototype data warehouse* ini dapat ditingkatkan dan dikembangkan menjadi sinergi basis data terdistribusi bertingkat untuk mendukung sistem informasi terpadu epidemiologi sebagai pelaksanaan sistem surveilans terpadu (*SST*) secara nasional, mulai dari unit surveilans puskesmas, rumah sakit, kabupaten/kota, propinsi sampai tingkatan nasional. Interoperabilitas data Sinkronisasi dan integrasi data antara unit surveilans dinas kesehatan kabupaten/kota menjadi masalah teknis utamanya. Setiap unit surveilans memiliki ketidak seragaman platform aplikasi dan database (*heterogen*). Permasalahan pokoknya adalah berkaitan dengan interoperabilitas atau kemampuan untuk mengintegrasikan aplikasi dan data yang berbeda antar unit surveilans. XML Web Service dapat menjadi solusinya.

XML Web Services adalah sebuah sitem perangkat lunak yang dirancang untuk mendukung inter-operasi dalam interaksi mesin ke mesin dalam sebuah jaringan (Erl,2007). Interaksi dilakukan melalui mekanisme atau protocol tertentu. Dengan demikian kemampuan web services dapat menambah kemampuan web untuk saling berkomunikasi dan bertukar informasi serta data dengan pola program-to-program. Dengan desain XML Web Services akan dapat mengintegrasikan sistem, bahasa pemrograman, basis data dan platform sistem

operasi yang berbeda dengan protocol HTTP (Hyper Text Transfer Protocol), sehingga aplikasi yang berbeda antar unit surveilans dapat berkomunikasi dengan data center dengan baik sehingga dapat mendukung data warehouse epidemiologi.

Tujuan penelitian ini adalah untuk melakukan analisis, perancangan, implementasi dan sinkronisasi integrasi data antara data unit surveilans (puskesmas dan rumah sakit) dengan data center (data warehouse epidemiologi) dalam suatu model aplikasi berbasis web. Manfaat dari penelitian ini adalah untuk memberi gambaran bagaimana langkah yang harus dilakukan dan dipersiapkan untuk mewujudkan integrasi data antar unit surveilans dalam mendukung data warehouse epidemiologi kesehatan.

BAB 2

TINJAUAN PUSTAKA

2.1. Sinkronisasi

2.1.1. Pengertian Sinkronisasi

Sinkronisasi adalah proses penyesuaian data terhadap skala waktu dari proses osilasi yang terjadi antara proses osilasi tersebut (*Balanov, 2010*). Pada dasarnya sinkronisasi terdiri dari dua jenis yaitu *one-way file synchronizaton* (sinkronisasi satu arah) dimana file-file yang telah mengalami perubahan pada bagian pusat (*source*) akan dibuat salinannya dan dipindah ke lokasi targetnya. Pada *one-way file synchronizaton* ini, tidak ada file dari target yang akan menuju ke bagian *source*. Sedangkan pada jenis yang kedua, *two-way file synchronizaton* (sinkronisasi 2 arah) proses pembuatan salinan dan pemindahannya dapat berjalan 2 arah baik dari *source* ke target maupun sebaliknya.

2.1.2. Protokol Sinkronisasi

Dalam teknik sinkronisasi dibutuhkan beberapa protokol yang digunakan mendukung komunikasi dan replikasi. Beberapa teknik sinkronisasi adalah *HotSyn, Intellisync, SyncML, CPISync*.

2.2. Replikasi

2.2.1. Pengertian Replikasi

Replikasi adalah suatu teknik untuk melakukan copy dan pendistribusian data dan objek-objek database dari satu database ke database lain dan melaksanakan sinkronisasi antara database sehingga konsistensi data dapat terjamin (*Dollimore, 2012*). Dengan menggunakan teknik replikasi ini, data dapat didistribusikan ke lokasi yang berbeda melalui koneksi jaringan lokal maupun internet.

2.2.2. Jenis Replikasi

Terdapat beberapa jenis replikasi diantaranya adalah :

1. Snapshot Replication : Mendistribusikan data yang dapat dilihat pada saat tertentu tanpa melakukan update.
2. Transactional Replication : Jenis replikasi ini lebih mementingkan dan memelihara kekonsistenan transaksi yang terjadi.
3. Merge Replication : Memungkinkan pengguna bekerja dan merubah data sesuai dengan wewenagnya. Pada saat server tidak dikoneksikan keseluruh lokasi dalam topologi, replikasi merubah data ke nilai yang sama.

2.2.3. Teknik Replikasi

Cara replikasi dalam DBMS terdistribusi dapat dikelompokkan dalam 2 teknik replikasi

www.learning.unl.ac.uk/csp003n/lectures/w021-ddb-arch.pdf :

1. Teknik Single Master Replicated : Dengan metode ini, salah satu komputer berfungsi sebagai master dan yang lainnya berfungsi sebagai slave. Pada prosesnya, komputer digunakan sebagai server akan dapat read dan write ke dalam database. Sedangkan komputer yang berfungsi sebagai slave, hanya akan read saja ke dalam basis data tersebut. Apabila kita melakukan perubahan data pada master, maka otomatis data pada slave akan berubah. Tetapi jika kita melakukan perubahan data pada slave, basis data pada master tidak akan berubah.
2. Teknik Multi Master Replicated Dengan metode ini, salah satu komputer berfungsi sebagai master server dan yang lainnya berfungsi sebagai master server juga. Pada prosesnya, setiap komputer akan dapat write dan read data dalam database. Apabila kita melakukan perubahan data pada master server 1, maka otomatis data pada master server 2 akan berubah. Begitu juga jika kita melakukan perubahan data pada master server 2, maka basis data pada basis data pada master server 1 akan berubah.

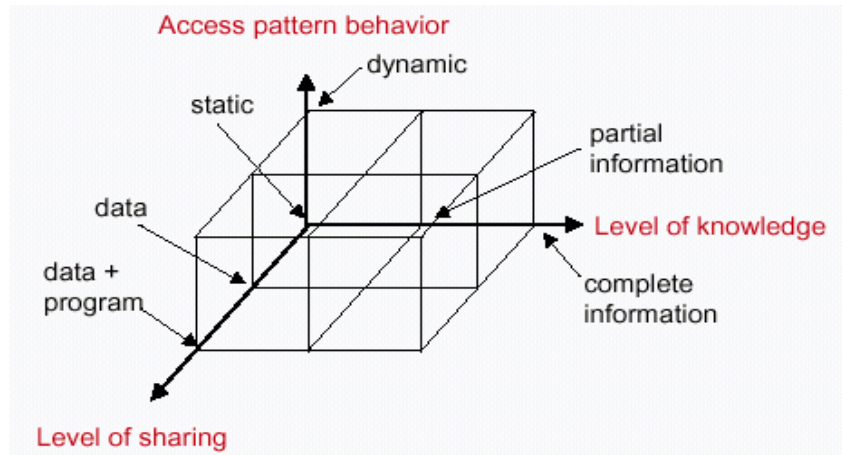
2.3. Basis Data Distribusi

2.3.1. Definisi Basis Data Distribusi

Basis data terdistribusi adalah basis data yang disebar pada sejumlah lokasi (A.Silberschatz, 2005). Setiap lokasi tersebut memiliki kewenangan sendiri dalam mengelola basis data. Masing-masing lokasi bisa melakukan transaksi lokal dan transaksi global. Basis data terdistribusi memiliki beberapa keuntungan seperti : meningkatkan performance, reliabilitas, ketersediaan data, memudahkan perluasan, dan meningkatkan Otonomi.

2.3.2. Desain Sistem Basis Data Distribusi

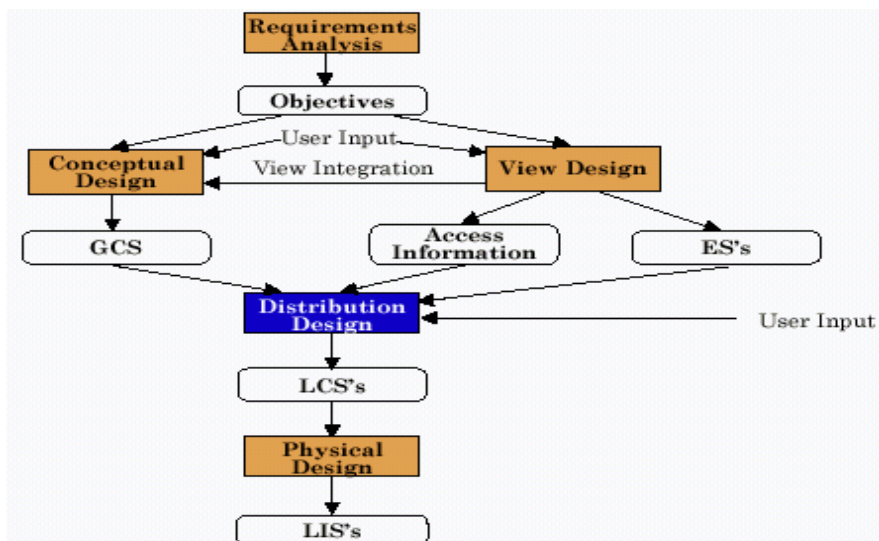
Desain sistem basis data terdistribusi meliputi bagaimana letak data dan program dalam suatu jaringan komputer. Dalam kasus sistem database tersebar, distribusi aplikasi meliputi 2 hal, yaitu distribusi software DBMS dan distribusi program aplikasi program aplikasi. Organisasi dari sistem tersebar dapat meliputi (Date C.J,2005) : Level Sharing, Pola Akses, Level Pengetahuan Pola Akses.



Gambar 1. : Framework Sistem Tersebar

Dalam strategi perancangan system database terdistribusi terdapat 2 pendekatan yaitu :

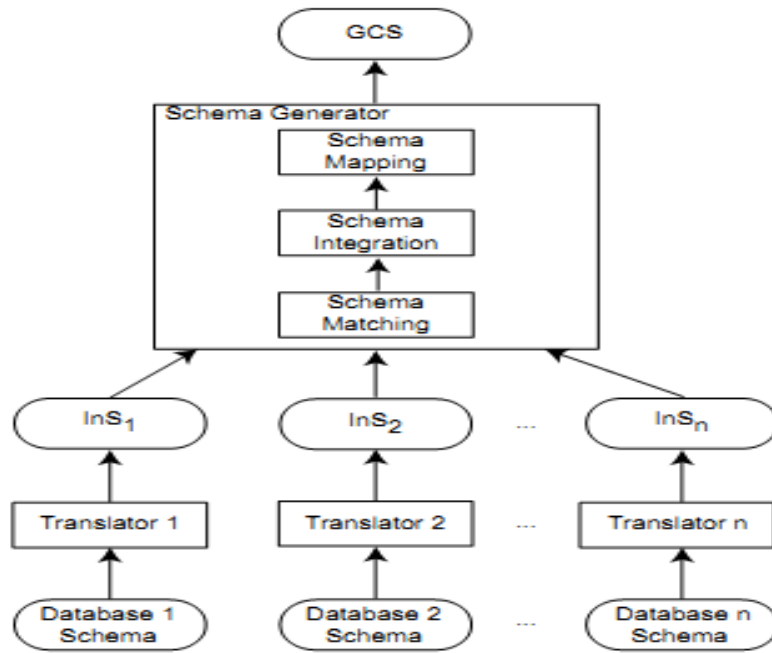
- a. Proses Perancangan Top Down dengan ciri – ciri pendekatan : Merancang system database dari awal, DBMS bersifat seragam (homogen)



Gambar 2. : Perancangan Secara Top Down

- b. Perancangan Button Up

Titik awal dari dari perancangan secara button up adalah individual local conceptual schema. Proses perancangan berupa proses integrasi local schema menjadi global conceptual schema. Ciri-ciri pendekatan ini adalah : database sudah ada dalam suatu situs, mengintegrasikan beberapa database heterogen menjadi satu database utama.



Gambar 3. : Perancangan Secara Button Up

2.3.3. Fragmentasi Sistem Multi Basis Data Distribusi

Fragmentasi data merupakan proses dimana basis data akan dipecah-pecah kedalam unit-unit logic yang disebut fragment yang kemudian akan disimpan dalam site yang berbeda (A. Tannenbaum, 2008). Suatu sistem basis data terdistribusi adakalanya dibentuk dari beberapa basis data yang berlainan. Sistem seperti ini disebut *multidatabase*, yang pembentukannya dilakukan dengan integrasi basis data. Dalam melakukan integrasi ini, boleh jadi ada ketidakseragaman antara basis data-basis data yang membentuknya dan mengakibatkan konflik, baik *konflik skema* (akibat ketidakseragaman skema relasi) maupun *konflik data* (akibat ketidakakuratan isi, misalnya presisi, besaran dan satuan, juga data yang tidak tepat atau sudah tidak berlaku [*obsolete*]). Oleh karena itu, perlu diperhatikan metode penyelesaian konflik sebelum terjadi integrasi, hal ini disebut restrukturisasi. Baik restrukturisasi table maupun atribut, proses ini jika dikaitkan dengan schema mapping dan shema matching disebut dengan *schema generator*.

1. Fragmentasi Horizontal : Fragmentasi berdasarkan tupel. Setiap fragment memiliki subset dari tupel relasi, Relasi r dibagi kedalam sejumlah subset r1, r2, r3....rn, masing-masing berisi dari sejumlah tupel relasi r. Masing-masing tupel relasi r harus merupakan satu dari fragment-fragment tersebut sehingga relasi awalnya dapat dibentuk kembali. Suatu

fragment didefinisikan sebagai seleksi pada relasi global r . Sebuah predikat P_i digunakan untuk menyusun fragmen r_i :

$$R_i = \sigma_{P_i}(r)$$

Pembentukan kembali dilakukan dengan mengembalikan seluruh fragment :

$$R = \bigcup_{i=1}^n r_i$$

2. Fragmentasi Vertikal : Fragmentasi vertikal dari r (R) melibatkan beberapa subset R_1, R_2, \dots, R_n dari sedemikian sehingga :

$$\bigcup_{i=1}^n R_i = R$$

Setiap fragment r_i dari r didefinisikan sebagai :

$$r_i = \pi_{R_i}(r)$$

Pembentukan kembali dengan menggunakan natural join $r = r_1 \bowtie r_2 \bowtie \dots \bowtie r_n$.

Fragmentasi vertikal dibuat dengan menambahkan atribut khusus yaitu tuple-id yang merupakan alamat fisik atau logika untuk tupel dan menjadi kunci untuk skema.

a. Derajat Fragmentasi: Performansi eksekusi suatu query sangat tergantung dari database yang mana dan seberapa besar dari database tersebut didekomposisi dan dialokasikan ke beberapa site. Derajat fragmentasi bisa nol (tidak terfragmentasi sama sekali), fragmentasi horisontal dan fragmentasi vertikal.

b. Aturan dalam Fragmentasi (sama dengan prinsip normalisasi) :

- ❖ Komplit : Dekomposisi relation/table R menjadi beberapa fragmen R_1, R_2, \dots, R_n dikatakan komplit kalau setiap item data pada R dapat juga ditemukan di beberapa R_i .
- ❖ Rekonstruksi : Kalau ada suatu relation/table R didekomposisi menjadi beberapa fragment R_1, R_2, \dots, R_n , maka harus ada operator yang dapat mengembalikan fragmen-fragmen tersebut ke R . ($R = \bigvee R_i, \forall R_i \in F_R$)
- ❖ Disjointness : Kalau ada suatu relation/table R didekomposisi menjadi beberapa fragment R_1, R_2, \dots, R_n , dan item data d_i ada di R_j , maka d_i harus tidak boleh ada di fragment R_k ($k \neq j$)

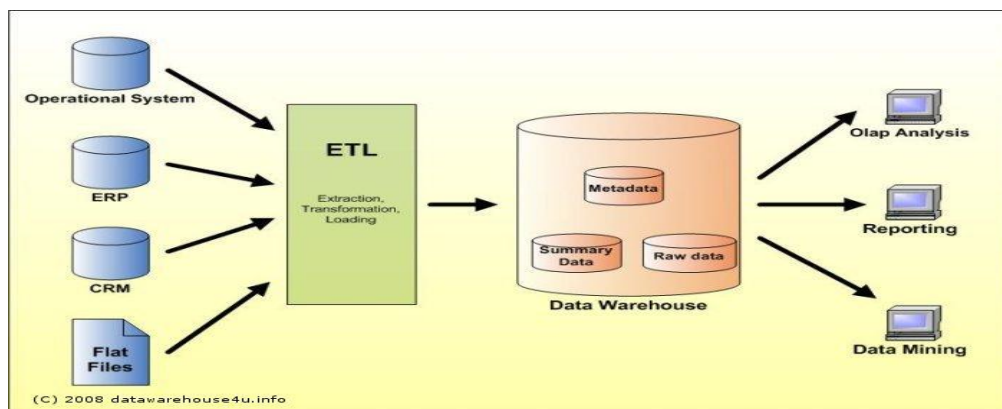
c. Teknik Alokasi fragmen ke beberapa site (Reliability & Efisiensi Query):

- ❖ Non-replicated (Partitioned Database): Setiap fragmen hanya diletakkan di satu site.
- ❖ Replicated: Fully Replicated: setiap fragmen ada di setiap site dan Partially Replicated: setiap fragmen ada di beberapa site.

2.4. Gudang Data (Data Warehouse)

Data warehouse merupakan suatu kumpulan data yang bersifat *subject-oriented*, terintegrasi, terus menerus dan *time variant* yang membantu *enterprise* atau organisasi dalam membuat keputusan. Sebagai pembuat keputusan maka dibutuhkan query beberapa nilai dari suatu subjek untuk melakukan proses analisis secara *real time*. *Data warehouse* dengan model multidimensi biasanya diimplementasikan dalam bentuk *star schema* agar memenuhi persyaratan. Pada model multi dimensional, *data warehouse* biasanya menyimpan data dalam bentuk *database* relasional (Efrem G, 2008).

Data warehouse didefinisikan sebagai sekumpulan data yang bersifat subject-oriented, terintegrasi, time variant, nonvolatile yang melayani sebagai implementasi fisik dari sebuah model data untuk mengambil keputusan dan menyimpan informasi untuk kebutuhan enterprise atas keputusan yang bersifat strategis. Teknologi dalam data warehouse meliputi data cleaning, integrasi data, dan OLAP sebagai teknik analisis dengan fungsi seperti menyimpulkan, konsolidasi dan agregasi sebaik kemampuan memandang informasi dari berbagai sudut (Reddy, dkk,2010).



Gambar 4. : Arsitektur Data Warehouse

2.5. XML (Extensible Markup Language)

XML (Extensible Markup Language) adalah bagian dari SGML (Standart Generalized Markup Language) XML merupakan bagian penting bagi pengembang yang ingin membangun web services. XML berbasis teks, sehingga bersifat *platform independent*. Sifat ini memungkinkan pertukaran data antar platform menjadi mungkin untuk dilakukan. Sebuah objek data dapat dikatakan sebagai dokumen XML jika memenuhi criteria well-formed. Adapun kriteria well –formed adalah sebagai berikut (Daconta,2007) : a) Satu dan hanya satu root element yang ada dalam satu dokumen, b) Setiap dokumen harus memiliki start-tag dan end-tag, c) Setiap elemen harus properly nested, d) Karakter pertama dari nama atribut harus berupa huruf alphabet atau underscore, e) Suatu nama atribut hanya boleh muncul satu kali pada start-tag yang sama. Terdapat dua

teknologi dominan yang dapat untuk mendefinisikan skema XML, yaitu Document Type Definition (DTD) dan XML Schema (Mawami, 2006).

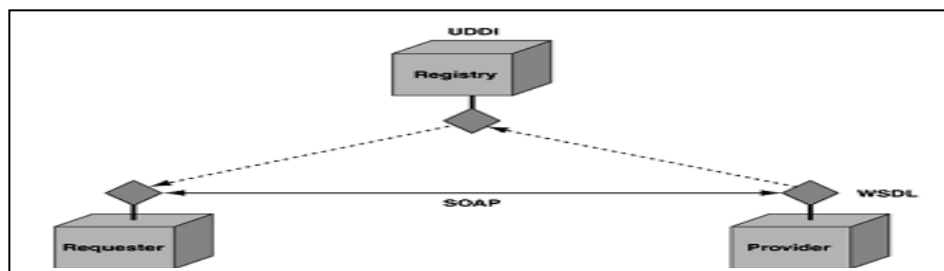
2.6. Web Services

Web services merupakan paradigma a arsitektur baru dalam aplikasi . Web services menerapkan layanan yang ada ke aplikasi lain (web services lain) melalui standar jaringan industry dengan interface dan protocol aplikasi. Suatu aplikasi dapat menerapkan layanan web services cukup melakukan request melalui jaringan . Web services menampilkan reuseable software software building block melalui alamat URL.

Teknologi web services menawaarkan cara untuk menjembatani aplikasi-aplikasi yang ingin berkomunikasi (Rusinawan,2004). Web services menyediakan staandar untuk berkomunikasi antara berbagai aplikasi yang berbeda dan dapat berjalan dalam platform maupun framework yang berbeda. Interoperabilitas web services menggunakan sekumpulan open standart yang terdiri dari HTTP, XML, SOAP, WSDL dan UDDI.

Untuk dapat menjalankan fungsi tersebut , web services memerlukan agen. Agen adalah potongan software yang mengirim dan menerima pesan. Agen dapat ditulis dengan berbagai bahasa pemrograman. Komunikasi memerlukan pembicara dan lawan bicara . Dalam web services dikenal dengan istilah provider s dan requesters. Web services dibangun dari tiga komponen utama yaitu services provider, services registry dan services requestor (Jorgensen, 2002). Ketiga komponen tersebut saling berinteraksi melalui komponen web services. Terdapat tiga macam operasi yang memungkinkan komponen tersebut untuk dapat saling berinteraksi, yaitu publish, find dan bind.

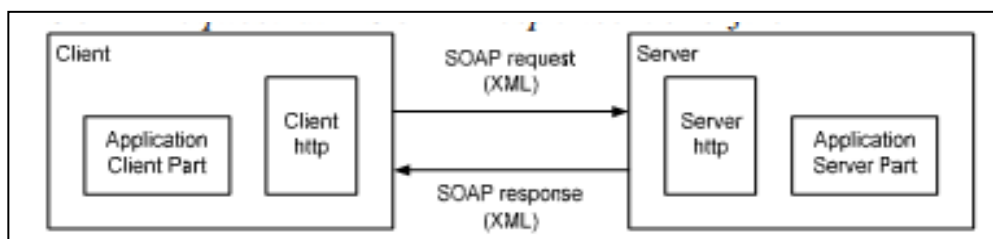
Dalam sekenario yang umum, services provider akan menetapkan sebuah modul perangkat lunak yang dapat diakses dalam sebuah jaringan, kemudian mendiskripsikan antarmuka yangsesui dengan layanan tersebut dan mempublikasikan secara langsung keservices requestor atau melaalui services registry. Pihak yang hendak mengkonsumsi web services akan menjalankan operasi find untuk menemukan diskripsi layanan yang diinginkan tersebut untuk mengikatkan diri ke services provider dan berinteraksi dengan implementasi web services. Keterkaitan anantara peranan dan operasi ketiga komponen tersebut dapat dilihat seperti gambar berikut :



Gamabr 5. Arsitektur Web Service

2.7. SOAP (Simple Object Access Protocol)

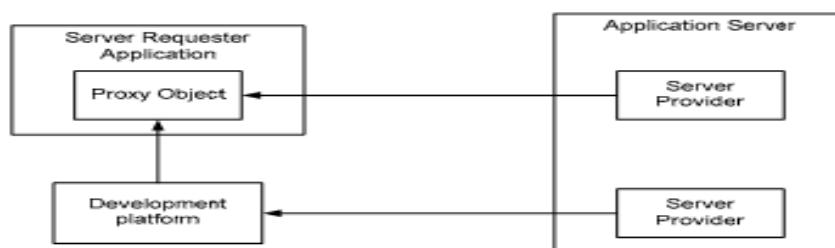
SOAP (Simple Object Access Protocol) adalah protocol untuk pertukaran informasi antara desentralisasi dan distribusi. Soap merupakan gabungan antara HTTP dengan XML , karena SOAP secara umum menggunakan protokol HTTP sebagai transport datanya dan data akan dipertukaarkan dalam format XML. SOAP mengatur bagaimana request dan respon dari suatu services bekerja. Proses SOAP request dan SOAP response dapat digambarkan seperti gambar di bawah ini.



Gambar 6. SOAP dengan standar XML

2.7. WSDL (Web Services Description Language)

WSDL (Web Services Description Language) adalah merupakan sebuah bahasa berbasis XML yang digunakan untuk mendefinisikan *web service* dan menggambarkan bagaimana cara untuk mengakses *web service* tersebut. Fungsi utama WSDL dalam *web service* adalah untuk mengotomasi mekanisme komunikasi *business-to-business* dalam *web service* melalui protokol internet. Dengan menggunakan WSDL *client* dapat memanfaatkan fungsi-fungsi publik yang disediakan oleh server (Putra,2009). Diagram WSDL dapat disajikan seperti gambar di bawah ini :



Gambar 7. Diagram Document WSDL

2.8. NuSOAP

NuSOAP adalah sebuah kumpulan *class-class* PHP yang memungkinkan user untuk mengirim dan menerima pesan SOAP melalui protokol HTTP. NuSOAP didistribusikan oleh NuSphere Corporation (<http://www.nusphere.com>) sebagai *open source toolkit* di bawah lisensi GNU GPL. Salah satu keuntungan dari NuSOAP adalah bahwa NuSOAP bukan

merupakan PHP *extension*, sehingga penggunaannya tidak membutuhkan registrasi khusus ke sistem operasi maupun *web server*. NuSOAP ditulis dalam kode PHP murni sehingga semua *developer web* dapat menggunakan *tool* ini tanpa tergantung pada jenis *web server* yang digunakan.

NuSOAP merupakan *toolkit web service* berbasis komponen. NuSOAP memiliki beberapa *class* yang menyediakan *method* seperti serialisasi variabel dan pemaketan *SOAP-Envelope*. Interaksi *web service* dilakukan dengan *class client* yang disebut dengan *class soapClient* dan *class server* yang disebut dengan *class soapServer*. *Class-class* ini mengizinkan *user* untuk melakukan proses pengiriman dan penerimaan pesan-pesan SOAP dengan bantuan beberapa *class-class* pendukung lainnya untuk melengkapi proses tersebut.

2.9. Integrasi Aplikasi

Integrasi aplikasi *enterprise* merupakan pendekatan strategis untuk membungkus beberapa sistem informasi bersama-sama dan mendukung kemampuan untuk bertukar informasi secara *real time*. Integrasi aplikasi ini adalah proses mengintegrasikan berbagai aplikasi independen, tujuannya adalah untuk berbagi data dan proses dapat dilakukan tanpa melakukan perubahan terhadap aplikasi-aplikasi dan struktur datanya, yang mana hal ini merupakan efektifitas dalam hal pembiayaan.

Terdapat beberapa pendekatan dalam integrasi aplikasi *enterprise* yaitu: *data integration* (integrasi data) yang digunakan untuk *sharing* informasi sederhana, *method integration* (integrasi metode) merupakan konsep pertukaran berbasis pesan yang disebut juga integrasi aplikasi web. *Process integration* (integrasi proses) yang disebut juga pendekatan otomatisasi proses yaitu terdapat penambahan kemampuan seperti proses *handling* dan simulasi kerja (Jorgensen, 2002).

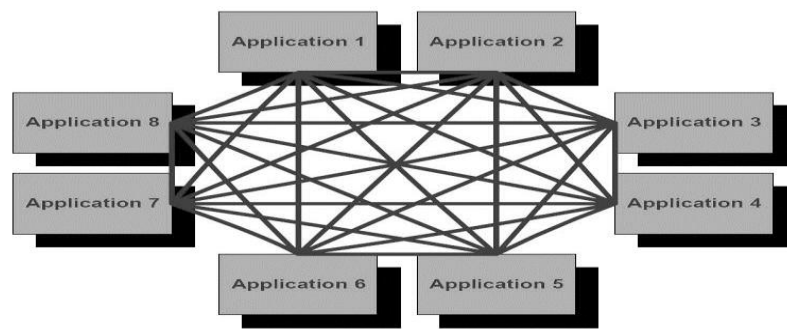
Solusi integrasi aplikasi *enterprise* menyediakan fungsionalitas melalui arsitektur integrasi menggunakan teknologi seperti *database-oriented-middleware, message oriented technologies, transaction based technologies, distributed object technologies, dan interface oriented technologies* (Jorgensen, 2002).

Ada beberapa teknologi yang digunakan untuk integrasi antar aplikasi diantaranya adalah sebagai berikut:

1. *Java Message Service (JMS)*.
2. *Remote Method Invocation (RMI)*.

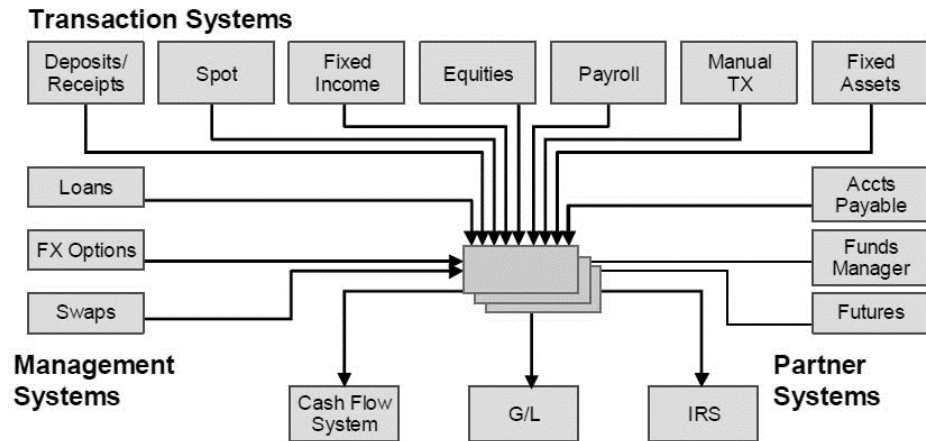
3. *Component Object Request Broker Architecture (CORBA)* .
4. *Web Service*.

Point-to-point merupakan model integrasi aplikasi yang paling lama, model ini digunakan untuk menyelesaikan masalah beberapa sistem *enterprise* yang saling terpisah. Yang mana pertukaran data menjadi kurang handal sehingga dibutuhkan secara manual untuk melakukannya. Idennya adalah membangun *interface object* dari dua sistem (*point*) yang diperlukan untuk melakukan pertukaran data. Data yang sesuai diambil oleh *interface object* dari sistem yang menjadi sumber data, untuk menyediakan sistem aplikasi sistem tujuan. Karakteristik dari sistem ini adalah tujuan *interface object* yang jelas, penyederhanaan transmisi format data, dan *performance* transmisi data yang tinggi. Tetapi memiliki *interface* yang kompleks dan penyambungan yang agak rumit, sehingga maintenance-nya rumit. Skema integrasi *point to point* dapat dilihat pada gambar ini.



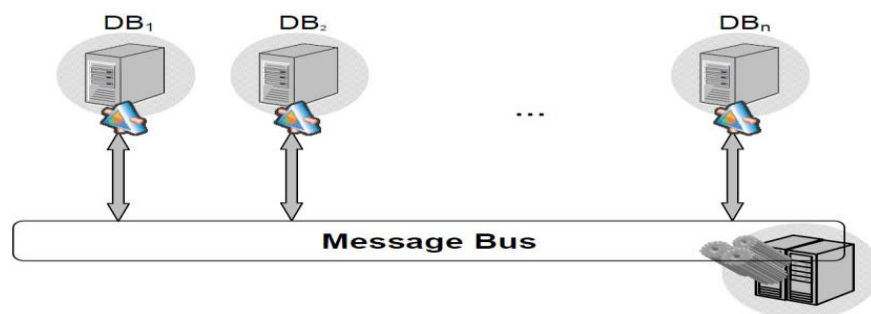
Gambar 8: Model *point to point*

Berikutnya ada model integrasi distribusi *star* diperkenalkan, model ini diperuntukkan untuk mengatasi *multi-interface* dan penyambungan yang berlebihan yang terdapat pada model *poin-to-point*. Model ini menggunakan *distributed control system (DCS)* dan *adapter*, yang mana memiliki konsep beberapa sistem akses ke perangkat sentral melalui *adapter*. Sehingga hanya ada satu *link* dalam arsitektur tersebut ke luar sistem. Tetapi redundansi data yang berbasis pada sinkronisasi data dibandingkan atas aplikasi berbasis bisnis merupakan sambungan yang lebih rapat dan memerlukan jaringan yang lebih tinggi. Model ini disebut sebagai model *hub and spoke (broker)*. Contoh skema model *hub and spoke* dapat dilihat pada gambar berikut ini[4].



Gambar 9: Model *hub and spoke*

Selanjutnya ada model *message bus* bertujuan untuk menyelesaikan masalah penyambungan berlebih, konektifitas dan skalabilitas. Ini menggunakan mekanisme pesan tak sinkron, yang mana memiliki banyak sistem yang terhubung ke *messagingbus* melalui *adapter* dan *converter*. Semua data dan informasi ditransfer ke dalam *form* dari *news*. *Message initiator* bertanggung jawab untuk melakukan *posting news* ke dalam *messagebus*, dan *news consumer* (pengguna) berlangganan *news* tertentu dari *message bus* untuk mendapatkan penyambungan yang lebih sedikit. Karena *message bus* menggunakan teknologi *publish/subscribe* untuk merespon pesan, maka kinerja sistem menurun di tingkat tertentu dan tidak dapat dicapai benar-benar secara simultan dan menjadi sulit untuk mencapai transaksi global antar berbagai sistem. Skema model *message bus* dapat dilihat pada gambar berikut ini.



Gambar 10: Model *message bus*

BAB 3

TUJUAN DAN MANFAAT PENELITIAN

Prinsip sinkronisasi dan integrasi data dalam penelitian ini tidak memerlukan keseragaman basis data dan aplikasi antar unit surveilans dalam mengelola data epidemiologi antar unit surveilans. Akan tetapi yang menjadi perhatian adalah bagaimana data kesehatan hasil dari transaksi masing - masing unit surveilans yang beragam bisa di sinkronkan dan selanjutnya diintegrasikan menjadi pusat data epidemiologi pada *data warehouse* epidemiologi dinas kesehatan untuk kepentingan sistem informasi surveilans epidemiologi. Untuk itu tujuan dalam penelitian ini adalah sebagai berikut :

- a. Mengidentifikasi masalah dan peluang pengembangan model *prototype data warehouse* melalui desain proses sinkronisasi, integrasi dan replikasi data antar DBMS unit surveilans yang tersebar dan beragam. .
- b. Mendapatkan gambaran tentang kebutuhan data, sumber data, mekanisme koleksi data (replikasi), mekanisme sinkronisasi, model integrasi data yang dapat di upayakan untuk mengembangkan model '*data center*' antar unit surveilans dalam model *prototype data warehouse*.
- c. Tersusunnya arsitektur data warehouse yang dapat digunakan sebagai '*data center*' dari berbagai unit surveilans sehingga dapat digunakan untuk monitoring analisis data kesehatan secara terpadu.
- d. Diperoleh desain model *prototype data warehouse* sebagai pola model sinkronisasi dan integrasi data yang terdistribusi secara heterogen untuk mendukung pelaksanaan sistem surveilans terpadu (SST) pada dinas kesehatan.
- e. Diperoleh diskripsi pola dan model integrasi data yang terdistribusi dari sumber skema local pada DBMS tersebar dan beragam dengan satu target skema global pada DBMS dinas kesehatan sebagai '*data center*' sehingga dapat mendukung pelaksanaan sistem surveilans terpadu (SST).
- f. Tersusunnya berbagai model analisa dan kebutuhan informasi yang melibatkan beberapa entitas unit surveilans dalam mendukung sistem informasi monitoring dan analisa data kesehatan secara terintegrasi dan terpadu.

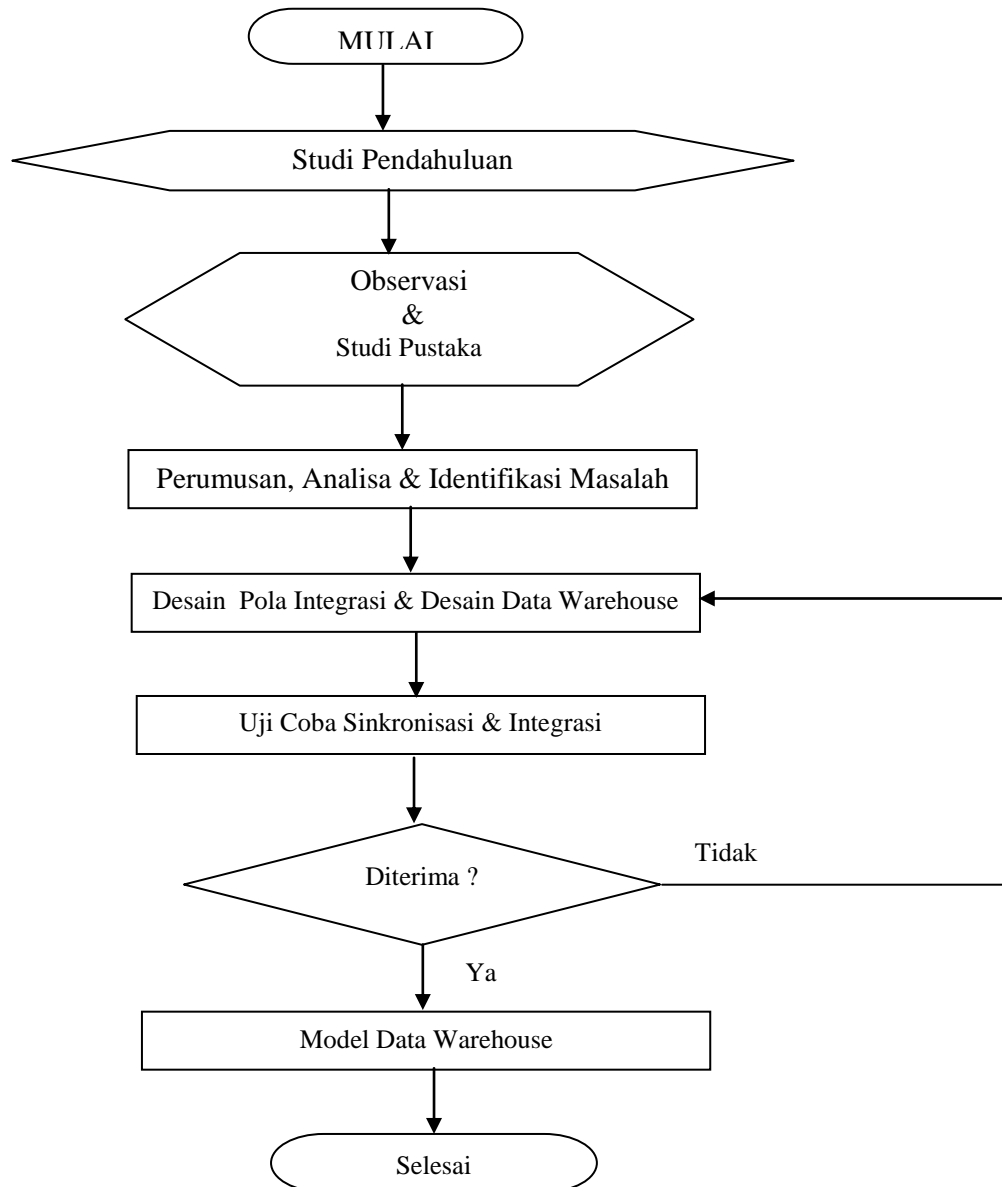
Sedangkan beberapa manfaat dalam penelitian ini adalah sebagai berikut :

- a. Memberikan diskripsi model yang efektif dengan mendesaian model data warehouse untuk dapat dimanfaatkan sebagai sarana teknologi untuk menghasilkan informasi epidemiologi yang berkualitas, tepat waktu dan berguna dalam mengambil keputusan menuju pembangunan kesehatan masyarakat yang berkelanjutan yang sesuai dengan misi *Millennium Development Goals (MDGs)*.
- b. Memberikan diskripsi model sinkronisasi dan integrasi data replikasi antar database unit-unit surveilans yang tersebar dan beragam sehingga tersedia data surveilans yang lengkap dan *up to date* dalam model *prototype data warehouse* (gudang data) untuk dapat dimanfaatkan oleh dinas kesehatan dalam pelaksanaan penyelenggaraan sistem surveilans terpadu (*SST*) untuk monitoring dan analisa data kesehatan.
- c. Memberikan masukan kepada unit pengelola sistem surveilans terpadu (*SST*) dinas kesehatan tentang pemanfaatan pola kaidah sinkronisasi dan integrasi dalam mengatasi masalah replikasi data antar unit surveilans sehingga bisa mendapatkan data yang sinkron dan sesuai untuk kepentingan sistem informasi epidemiologi secara bertingkat.
- d. Meningkatkan kultur dan atmosfir akademik lewat penelitian dalam rangka memanfaatkan ilmu atau teori yang bersifat akademik dalam membantu meningkatkan pengembangan teknologi yang bermanfaat bagi lingkungan atau masyarakat.

BAB 4 METODE PENELITIAN

4.1. Rancangan Penelitian

Pada penelitian ini digunakan pendekatan metode studi literatur (library research) dan studi lapangan (field research) untuk mendesain pola sinkronisasi. Adapun tahapan penelitian dapat dijelaskan dalam diagram dibawah ini :



Gambar 11. : Diagram Alir Penelitian

1. Studi Pendahuluan : Pada tahapan ini merupakan kegiatan untuk mengenali lebih lanjut tentang obyek penelitian beserta lingkungan yang terkait dalam rangka mendalami situasi

dan kondisi dari sinkronisasi dan integrasi yang akan dikembangkan. Studi pendahuluan dilakukan dengan mengumpulkan informasi mengenai pengelolaan sistem surveilans terpadu (*SST*).

2. Observasi dan Studi Pustaka : Pada tahapan ini akan dilakukan analisis kebutuhan dengan analisa diskriptif dengan cara melakukan kajian pustaka yang terkait dengan konsep sinkronisasi dan integrasi data dan keberadaan data dan database kesehatan yang digunakan untuk surveilans kesehatan pada masing-masing unit surveilans dengan dinas kesehatan sebagai pusat data (*data center*) .
3. Perumusan, Analisa dan Identifikasi Masalah : Tahap selanjutnya setelah mendapatkan permasalahan utama dari obyek penelitian dan dilengkapi dasar teori dari studi pustaka yang mendukung, adalah merumuskan permasalahan, menganalisa dan mengidentifikasi masalah yang berkaitan dengan obyek yang akan dieksplorasi dalam rangka menemukan pola baru dalam sinkronisasi dan integrasi data antar unit surveilans dengan database surveilans epidemiologi dinas kesehatan dalam bentuk *data warehouse* sebagai *data center*.
4. Desain Pola Integrasi dan Desain *Data Warehouse*: Pada tahapan ini dilakukan desain pola integrasi, sinkronisasi dan desain *data warehouse* sebagai model integrasi data berdasarkan kebutuhan dan identifikasi masalah yang ada kaitanya dengan sistem monitoring dan analisa data kesehatan. Pada tahap desain ini akan dihasilkan desain produk baik dari sisi konsep teori (*conceptual design*) dan desain produk teknis (*data warehouse design*) yang siap implementasi dalam bentuk model *prototype data warehouse* untuk sinkronisasi dan integrasi data sistem distribusi antar unit surveilans.
5. Pengujian : Pada tahap pengujian ini dilakukan evaluasi terhadap hasil desain *prototype data warehouse* yang memiliki proses sinkronisasi dan integrasi dengan aturan yang sesuai sehingga nantinya pola sinkronisasi database dapat digunakan secara maksimal dan sesuai (diterima) sebagai suatu pola baru untuk mendukung pengembangan system informasi epidemiologi kesehatan dari lingkungan database distribusi yang heterogen.
6. Hasil Pola Sinkronisasi dan Integrasi (*prototype data warehouse*) : Hasil dari pola model relasional sinkronisasi dan integrasi data ini merupakan sistem manajemen database yang menunjang proses sinkronisasi database berbasis proses replikasi data dari masing-masing unit surveilans yang heterogen (puskesmas, laboratorium, rumah sakit) menuju database epidemiologi dinas kesehatan sebagai *data center* kesehatan dalam bentuk *prototype data warehouse*. Sehingga model ini dapat mempermudah dan mendukung ketersediaan data

dan akses data untuk digunakan sebagai sistem monitoring dan analisa data kesehatan maupun untuk diolah menjadi system informasi epidemiologi kesehatan kabupaten/kota.

4.2. Bagan Penelitian

Tabel 1. : Bagan Penelitian

Tahap	Kegiatan	Luaran	Indikator Pencapaian
Tahun Ke 2	Detail Desain Pola Distribusi dan Pembuatan Dimensi Data	Tersusunnya pola arsitektur database distribusi homogen. Tersusunnya pola arsitektur database distribusi heterogen. Tersusunnya pola kolaborasi distribusi database heterogen dan homogen sebagai format data center pada prototype data warehouse.	Pola distribusi bisa digunakan simulasi distribusi data dari masing-masing unit surveilans ke server database epidemiologi dengan baik dan benar.
	Detail Desain Pola Sinkronisasi & Integrasi	Tersusunnya pola arsitektur sinkronisasi database homogen. Tersusunnya pola arsitektur sinkronisasi database heterogen. Tersusunnya pola kolaborasi sinkronisasi database heterogen dan homogen sebagai format data center .	Pola sinkronisasi dan integrasi bisa digunakan simulasi sinkronisasi data dari masing-masing unit surveilans ke server database epidemiologi dengan baik dan benar.
	Detail Desain Schema Data Warehouse	Diskripsi fungsi sinkronisasi yang sesuai dengan atribut dan batasan parameter database epidemiologi.	Memastikan bahwa model sinkronisasi dan replikasi distribusi dari sumber ke target dapat digunakan untuk sistem surveilans secara terpadu.
	Uji Prototype Datawarehouse untuk Sinkronisasi dan Replikasi Data Kesehatan	Implementasi sistem sinkronisasi yang sesuai dengan spesifikasi fungsi sinkronisasi dan spesifikasi fungsi replikasi yang sesuai dengan kebutuhan distribusi.	Model yang bisa digunakan dengan mudah untuk simulasi surveilans. Data sinkronisasi yang dihasilkan sesuai dengan kebutuhan user.
	Menyusun model data warehouse dan membandingkan antar alternatif pola integrasi	Nilai evaluasi ujicoba sebagai dasar refisi pemilihan pola integrasi dan sinkronisasi data.	Terdapat selisih nilai antara nilai ujicoba dari beberapa alternatif pola sinkronisasi.
	Pemilihan Pola Integrasi & Sinkronisasi	Didapat pilihan pola sinkronisasi dan integrasi yang paling tepat sebagai bagian dari proses data warehouse	Diskripsi performansi masing-masing alternatif pola sinkronisasi pada model prototype data warehouse.
	Data Hasil Penerimaan Model Data Warehouse dan Pola Integrasi	Kesimpulan penerimaan dalam bentuk jurnal. Pengajuan pengakuan model dalam HKI	Dokumentasi dalam Jurnal, pengajuan HKI, pola sinkronisasi yang bisa dipakai untuk mengelola distribusi data kesehatan.

BAB 5

HASIL YANG DICAPAI

5.1. Hasil Analisis Dan Identifikasi Masalah

5.1.1. Hasil Analisis Integrasi Unit Surveilans

Analisis dari sistem integrasi dari beberapa sub system (unit surveilans) yang heterogen adalah sebagai berikut:

a. Integrasi antara Sistem Kependudukan dengan Sistem Puskesmas

Sistem Dinas Kependudukan yang memiliki data penduduk. Untuk menjaga konsistensi data penduduk kemudian data kependudukan tersebut yang akan digunakan oleh sistem Puskesmas untuk mencatat data pasien.

b. Integrasi antara Sistem Kependudukan dan Sistem Rumah Sakit

Sama seperti pada integrasi antara sistem Kependudukan dengan sistem Puskesmas yaitu data penduduk dari sistem Dinas Kependudukan yang akan digunakan oleh sistem Rumah Sakit untuk mencatat data pasien.

c. Integrasi antara Sistem Puskesmas dengan Sistem Dinas Kesehatan

Sistem informasi Puskesmas yang menggunakan data penduduk akan mencatat data pemeriksaan pasien serta memiliki data penyakit yang sering terjadi pada Puskesmas itu. Data tersebut bisa dilihat secara *real time* oleh sistem Dinas Kesehatan kemudian data dari Puskesmas akan disimpan di basis data Sistem Dinas Kesehatan, Namun ada konflik penamaan tabel, atau struktur atribut tabel yang berbeda.

d. Integrasi antara Sistem Rumah Sakit dengan Sistem Dinas Kesehatan

Sistem informasi Rumah sakit yang telah mencatat data pemeriksaan pasien, dengan data pasien memanfaatkan data dari sistem Dinas Kependudukan. Kemudian data pencatatan dari pemeriksaan dapat di *monitoring* langsung oleh sistem Dinas Kesehatan, serta data pencatatan dan penderita penyakit akan di simpan pada basis data sistem Dinas Kesehatan, Namun ada konflik pada jenis basis data yang digunakan, penamaan tabel, atau struktur atribut tabel yang berbeda dari Puskesmas, Rumah Sakit dan Dinas Kesehatan.

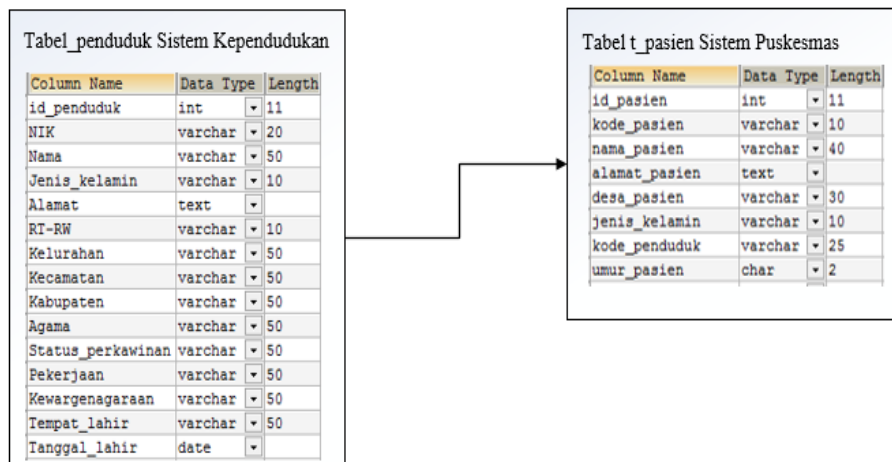
5.1.2. Hasil Identifikasi Masalah Integrasi

Adalah tindaklanjut dari adanya permasalahan (dari segi rancangan integrasi) yang timbul dari interaksi antar sistem tadi. Hasil rancangan struktur table hanya digunakan sebagai mekanisme integrasi dalam prototype data warehouse tanpa mengganggu dan mempengaruhi

system manual yang sudah ada. Adapun beberapa masalah yang teridentifikasi adalah sebagai berikut:

- a. Masalah integrasi antara sistem Dinas Kependudukan dan sistem Puskesmas dalam menggunakan data kependudukan yang akan digunakan menjadi data pasien di Puskesmas. Ada beberapa masalah yang timbul yaitu sebagai berikut:
 1. Untuk mengambil data penduduk tidak mungkin melakukan query langsung ke database sistem Kependudukan karena alasan privasi.
 2. Perbedaan nama tabel. Nama tabel pada sistem Kependudukan adalah tabel_penduduk, sedangkan tabel pasien yang ada pada sistem Puskesmas adalah t_pasien.
 3. Perbedaan struktur atribut tabel antara tabel_penduduk dengan t_pasien pada sistem Puskesmas.

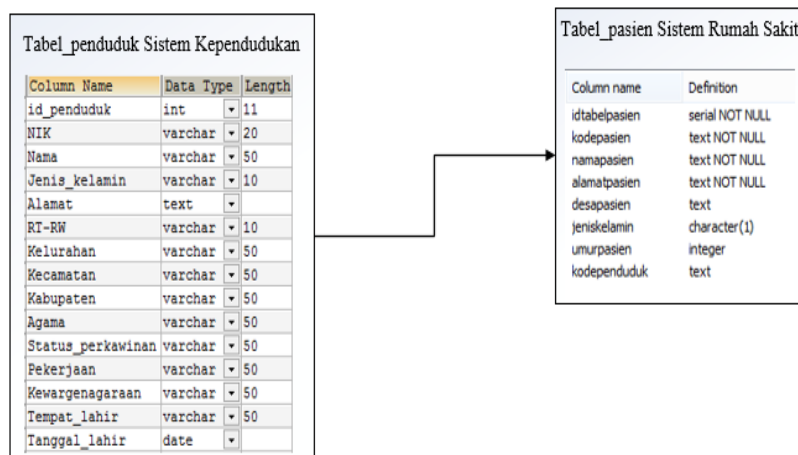
Berikut adalah hasil rancangan struktur atribut tabel penduduk pada sistem Kependudukan serta tabel pasien pada sistem Puskesmas.



Gambar 12: Rancangan nama dan struktur atribut tabel tabel_penduduk sistem Kependudukan dengan t_pasien sistem Puskesmas

- b. Masalah integrasi antara sistem Rumah sakit dengan sistem Kependudukan dalam menggunakan data penduduk yang akan digunakan untuk membuat data pasien di sistem Rumah sakit, ada beberapa masalah yaitu sebagai berikut:
 1. Untuk mengambil data penduduk tidak mungkin sistem Rumah sakit langsung melakukan query ke sistem Kependudukan, karena alasan privasi data.

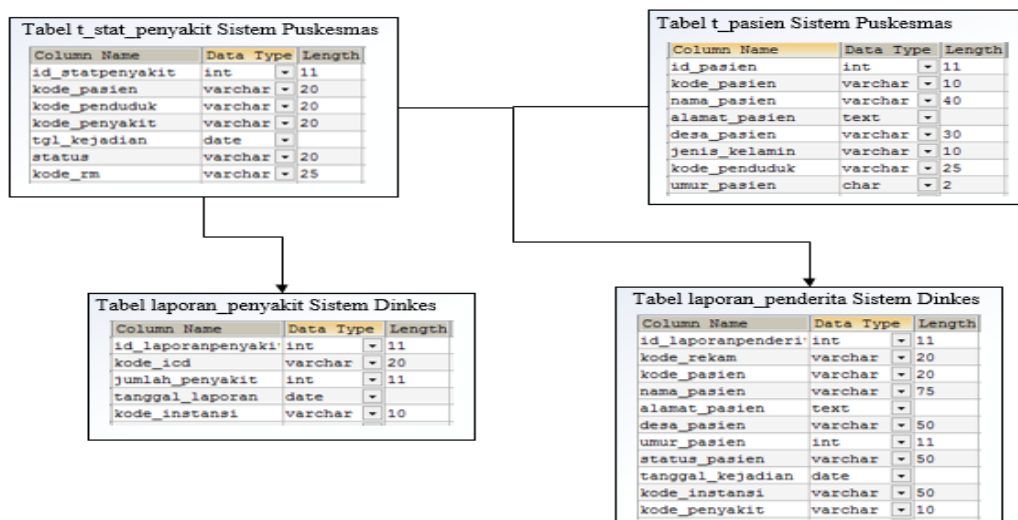
2. Perbedaan jenis Database Management System (DBMS), pada sistem Kependudukan DBMS yang digunakan adalah MySQL sedangkan sistem Rumah sakit menggunakan DBMS PostgreSQL.
3. Perbedaan nama tabel, nama tabel pada sistem Kependudukan yaitu tabel_penduduk, sedangkan nama tabel pada sistem Rumah sakit adalah tabel_pasien.
4. Perbedaan struktur atribut tabel antara tabel_penduduk dan tabel_pasien. Berikut adalah rancangan struktur atribut tabel penduduk pada sistem Kependudukan.



Gambar 13: Rancangan nama dan struktur atribut tabel_penduduk sistem Kependudukan dengan tabel_pasien sistem Rumah sakit

- c. Masalah integrasi dan sinkronisasi laporan dari Puskesmas ke Dinas Kesehatan. Ada beberapa masalah yaitu sebagai berikut:
 1. Berikut adalah jenis laporan yang dikirim ke Dinas Kesehatan oleh Puskesmas yaitu sebagai berikut:
 - a) Laporan W1 / KD
Yaitu laporan wabah atau penyakit harian dari pencatatan diagnosa penyakit pasien.
 - b) Laporan W2
Yaitu laporan wabah atau penyakit mingguan dari pencatatan diagnosa penyakit pasien.
 - c) Laporan LB1
Yaitu laporan data kesakitan keseluruhan yang direkap tiap bulan.

- d) Laporan Pasien W1
Yaitu laporan yang berisi data pasien yang menderita penyakit pada laporan W1.
- e) Laporan Pasien W2
Yaitu laporan yang berisi data pasien yang menderita penyakit pada laporan W2.
- f) Laporan Pasien LB1
Yaitu laporan yang berisi data pasien yang menderita penyakit pada laporan LB1.
- Untuk mengambil data – data laporan tidak memungkinkan untuk melakukan query langsung ke database sistem Puskesmas.
 - Untuk melakukan integrasi dan sinkronisasi data laporan dan data penderita penyakit yang akan disimpan di data center sistem Dinas Kesehatan ada perbedaan arsitektur sistem, struktur atribut tabel dan penamaan tabel. Berikut adalah gambar skema integrasi dan sinkronisasi serta perbedaan arsitektur, penamaan tabel, serta struktur atribut tabel dari database sistem Puskesmas dengan sistem Dinas Kesehatan.

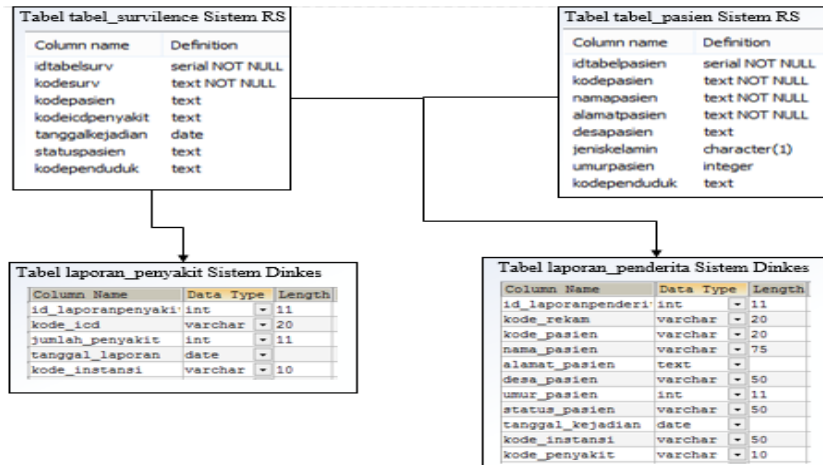


Gambar 14 : Rancangan arsitektur, penamaan tabel, dan struktur atribut antara sistem Puskesmas dengan sistem Dinkes

- d. Masalah integrasi dan sinkronisasi antar sistem Rumah sakit dengan sistem Dinas Kesehatan adalah sebagai berikut :

1. Ada beberapa laporan yang akan dibagikan datanya ke sistem Dinas Kesehatan yaitu sebagai berikut:
 - a) Laporan W1
Yaitu laporan wabah atau penyakit harian hasil diagnosa penyakit pasien di Rumah Sakit.
 - b) Laporan W2
Yaitu laporan wabah atau penyakit mingguan yang terjadi pada Rumah Sakit.
 - c) Laporan LB1
Yaitu laporan data kesakitan keseluruhan yang terjadi tiap bulannya.
 - d) Laporan Pasien W1
Yaitu berisi data pasien yang menderita penyakit sesuai dengan laporan W1.
 - e) Laporan Pasien W2
Yaitu berisi data pasien yang menderita penyakit sesuai dengan laporan W2.
 - f) Laporan Pasien LB1
Yaitu berisi data pasien yang menderita penyakit sesuai dengan laporan LB1.
2. Untuk mengambil data – data laporan tidak memungkinkan untuk melakukan query langsung ke database sistem Puskesmas.
3. Ada perbedaan jenis Database Management System (DBMS), pada sistem Rumah sakit menggunakan DMBS PostgreSQL sedangkan pada sistem Dinas Kesehatan menggunakan DBMS Mysql.
4. Ada perbedaan pada arsitektur sistem, penamaan tabel yang berbeda, serta struktur tabel yang berbeda.

Berikut adalah gambar skema integrasi dan sinkronisasi serta perbedaan arsitektur, penamaan tabel dan struktur atribut antara sistem Rumah sakit dengan sistem Dinas Kesehatan.

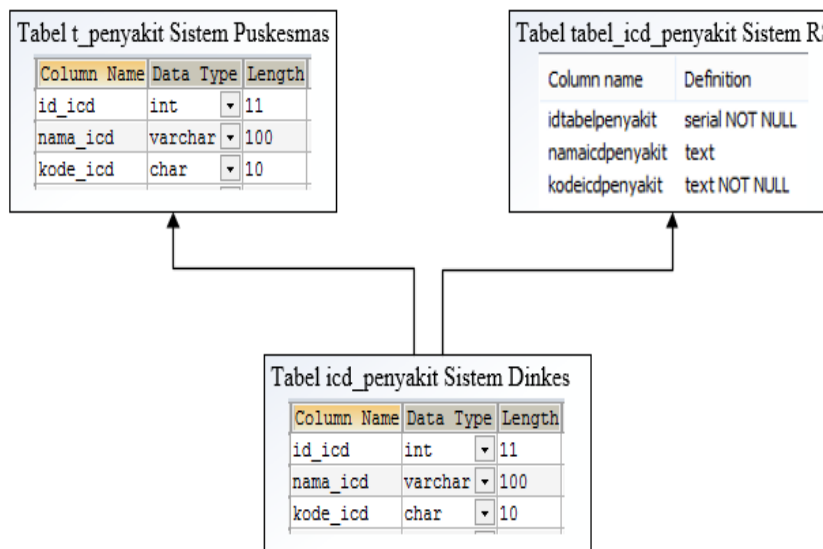


Gambar 15: Rancangan arsitektur, penamaan tabel, dan struktur atribut antara sistem Rumah sakit dengan sistem Dinkes

e. Masalah integrasi antar Sistem Dinas Kesehatan untuk memerikan data ICD penyakit yang telah distandarkan adalah sebagai berikut:

1. Untuk mengambil data – data laporan tidak memungkinkan untuk melakukan query langsung ke database sistem Dinas Kesehatan.
2. Adanya perbedaan penggunaan Database Management System dari sistem yang akan mengambil data.
3. Perbedaan penamaan tabel, serta struktur atribut tabel yang berbeda.

Berikut adalah gambar skema model integrasi dari tabel dan basis data yang heterogen.



Gambar 16: Rancangan nama dan struktur atribut tabel dari tiga tabel yang menyimpan ICD penyakit

5.2. Hasil Identifikasi Desain Integrasi Dengan Web Services

- a. Integrasi data penduduk antara sistem Kependudukan dengan Sistem Puskesmas dan Sistem Rumah Sakit dibuat *service* untuk membagi data penduduk ke sistem lain menggunakan *web service* SOAP dengan komunikasi data menggunakan XML. Berikut adalah web service yang diberikan oleh sistem Kependudukan.

1. getPenduduk

Yaitu service yang digunakan untuk mendapatkan data penduduk agar pada proses input data pasien tidak menambah waktu lagi karena sudah ada data penduduk yang berada pada sistem informasi pada Dinas Kependudukan.

- b. Integrasi dan sinkronisasi antara sistem Puskesmas dengan sistem Dinas Kesehatan dalam proses monitoring data laporan menggunakan *service* data laporan dan *service* data penderita menggunakan web service SOAP dengan komunikasi data XML. Berikut adalah web service sistem Puskesmas untuk melihat data laporan penyakit dan laporan penderita penyakit.

1. getLaporanW1

Yaitu service untuk memberikan laporan W1 ke sistem Dinas Kesehatan yang berisi data penyakit per hari, dengan memiliki parameter tanggal.

2. getLaporanW2

Yaitu service untuk memberikan laporan W2 ke sistem Dinas Kesehatan yang berisi data penyakit per minggu, dengan memiliki parameter tanggal.

3. getLaporanLB1

Yaitu service untuk memberikan laporan LB1 ke sistem Dinas Kesehatan yang berisi data penyakit per bulan, dengan parameter bulan dan tahun.

4. getPenderitaW1

Yaitu untuk memberikan data penderita penyakit yang ada di Puskesmas ke Dinas Kesehatan yang sesuai dengan laporan W1.

5. getPenderitaW2

Yaitu untuk memberikan data penderita penyakit yang ada di Puskesmas ke Dinas Kesehatan sesuai dengan laporan W2.

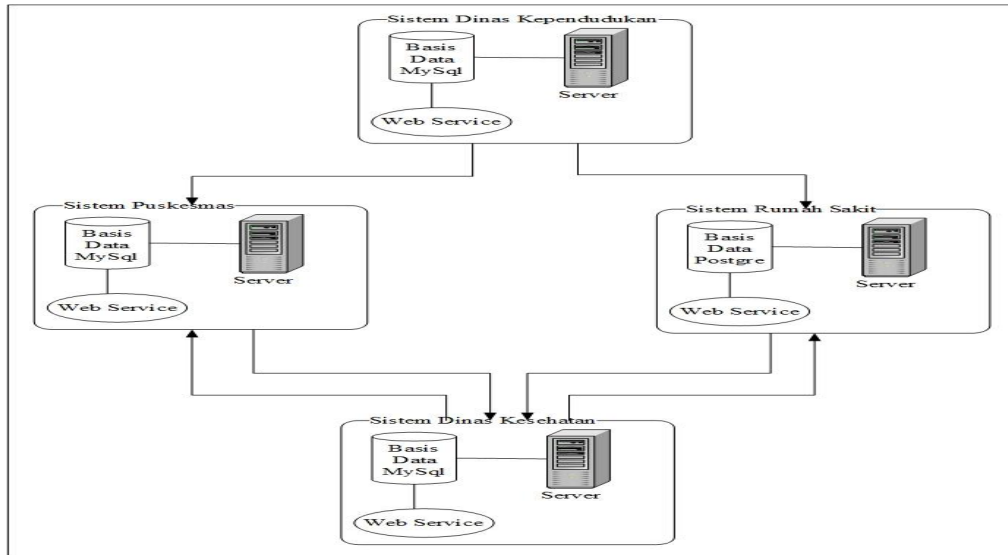
6. getPenderitaLB1

Yaitu untuk memberikan data penderita penyakit yang ada di Puskesmas ke Dinas Kesehatan sesuai dengan laporan LB1.

- c. Integrasi dan sinkronisasi antara sistem Rumah sakit dengan sistem Dinas Kesehatan dalam proses monitoring data laporan penyakit dan laporan penderita penyakit menggunakan *service* menggunakan web service SOAP dengan komunikasi data XML. Berikut adalah web service sistem Rumah sakit untuk memerikan data laporan penyakit dan laporan penderita penyakit.
1. getLaporanW1RS
Yaitu service yang diberikan sistem Rumah sakit agar bisa melihat data penyakit harian.
 2. getLaporanW2RS
Yaitu service yang diberikan sistem Rumah sakit agar bisa mengetahui data penyakit mingguan.
 3. getLaporanLB1RS
Yaitu service yang diberikan sistem Rumah Sakit untuk mengetahui data penyakit selama sebulan.
 4. LaporanPenderitaPenyakitW1
Yaitu service untuk mengetahui penderita penyakit yang sesuai dengan laporan W1 atau harian.
 5. LaporanPenderitaPenyakitW2
Yaitu service untuk mengetahui penderita penyakit yang sesuai dengan laporan W2 mingguan.
 6. LaporanPenderitaPenyakitLB1
Yaitu service untuk mengetahui penderita penyakit yang sesuai dengan laporan LB1 atau laporan penyakit bulanan.
- d. Integrasi data ICD penyakit antara sistem Puskesmas dan sistem Rumah Sakit dengan sistem Dinas Kesehatan dibuat *service* untuk membagi data ICD penyakit ke sistem lain menggunakan *web service* SOAP dengan komunikasi data menggunakan XML. Berikut adalah web service yang diberikan oleh sistem Dinas Kesehatan.
1. getDataICD
Yaitu service untuk mendapatkan data master ICD penyakit yang sudah distandarkan oleh Dinkes.

5.3. Hasil Rancangan Arsitektur Integrasi Antar Host Unit Surveilans Dengan Dinkes.

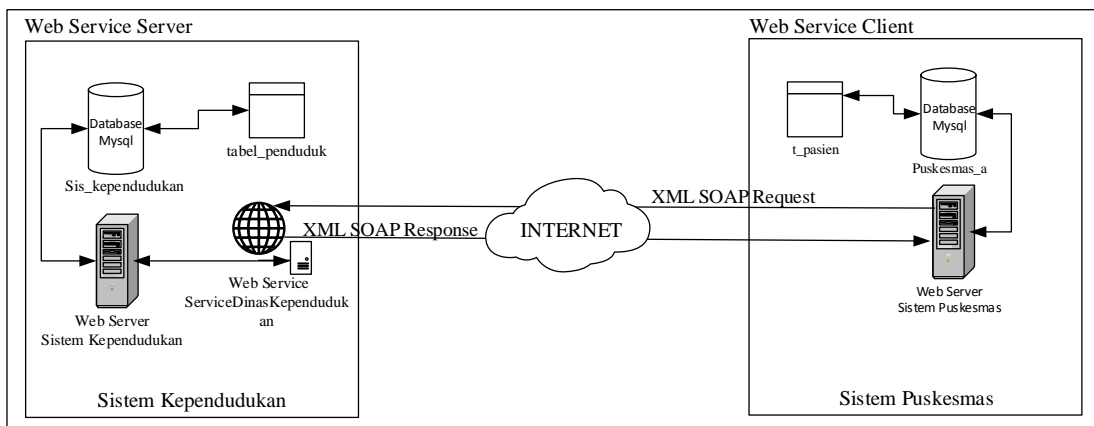
Berikut adalah hasil desain sistem integrasi dari sistem informasi yang heterogen, dengan membuat prototipe empat sistem informasi yang berbeda. Seperti pada gambar berikut ini.



Gambar 17: Desain Integrasi Antar Unit Surveilans Denga Web Service antar Sistem
 Penjelasan Hasil Integrasi dengan Interaksi Web Services adalah sebagai berikut :

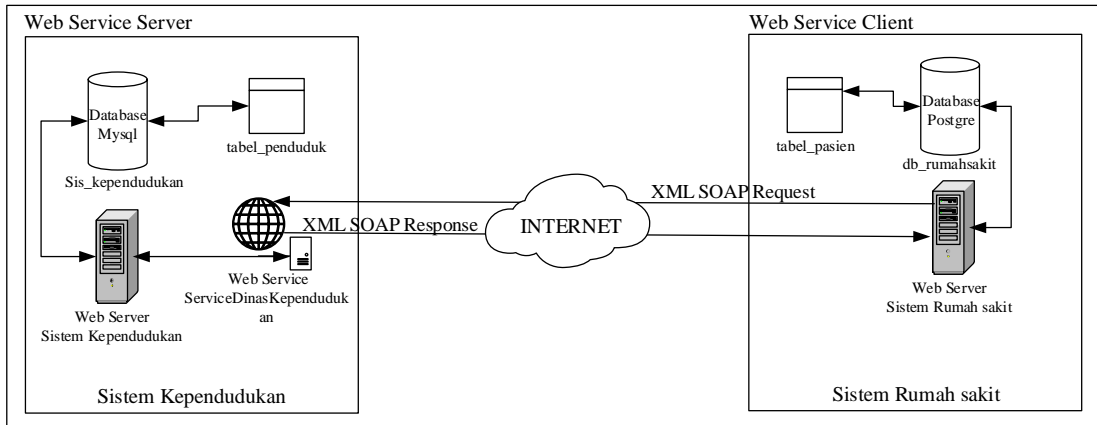
a. Sistem Informasi Dinas Kependudukan

Sistem informasi dari Dinas Kependudukan akan memberikan service ke Puskesmas dan Rumah Sakit, Berikut ini adalah arsitektur integrasi antara sistem Kependudukan dengan sistem Puskesmas.



Gambar 18: Integrasi antara Sistem Kependudukan dengan Sistem Puskesmas

Sistem Kependudukan juga memberikan service atau terintegrasi dengan sistem Rumah sakit, berikut ini adalah arsitektur integrasi antara sistem Kependudukan dengan sistem Rumah sakit.



Gambar 19: Integrasi antara Sistem Kependudukan dengan Sistem Puskesmas

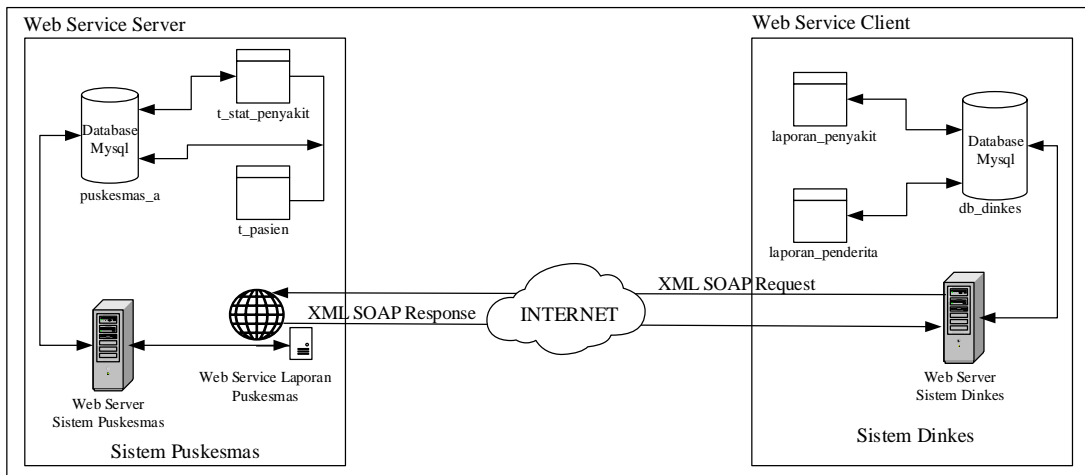
Web service ServiceDataKependudukan yang akan membagi data ke host lain yaitu sebagai berikut:

Tabel 2. Perancangan web service sistem Dinas Kesehatan

Method Web Service	Parameter	Url WSDL
getDataPenduduk	-	localhost/fik/disduk/ws/kependudukan.php?wsdl

b. Sistem Informasi Puskesmas

Sistem informasi Puskesmas akan memberikan service kepada dinas agar bisa melihat data laporan pencatatan penyakit. Berikut ini adalah arsitektur sistem integrasi.



Gambar 20 : Arsitektur integrasi Sistem Puskesmas dengan Sistem Dinkes

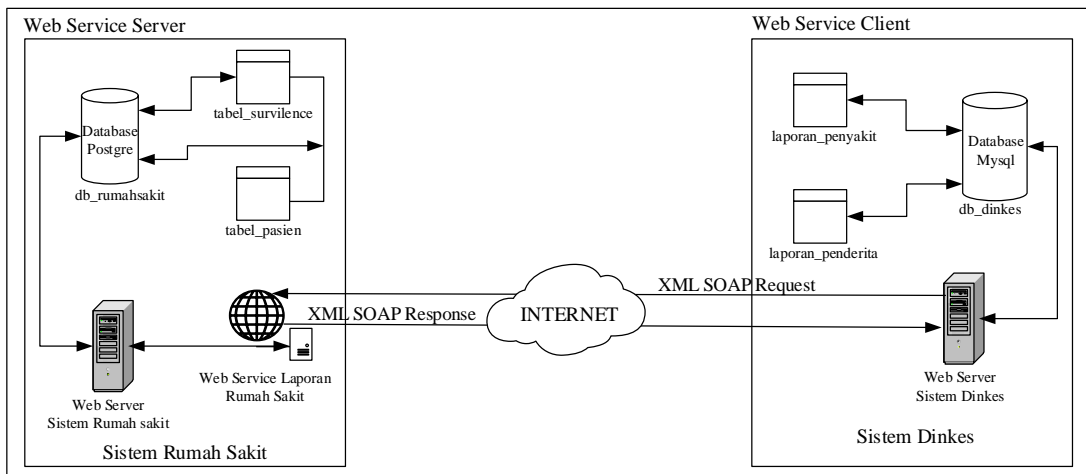
Data yang akan di integrasikan adalah data-data laporan, dengan metode web service berikut method yang membagi data ke sistem lain yaitu sebagai berikut:

Tabel 3. Perancangan web service sistem Puskesmas

Method Web Service	Parameter	Url WSDL
getLaporanKDPA	Tanggal (yyyy-mm-dd)	http://localhost/fik/puskesmasA/ws/puskesmas_service.php?wsdl
getLaporanW2PA	Tanggal (yyyy-mm-dd)	
getLaporanLB1PA	TahunBulan (yyyy-mm)	
getPenderitaW1	Kodeicd dan tgl (yyyy-mm-dd)	
getPenderitaW2	Kodeicd dan tgl (yyyy-mm-dd)	
getPenderitaLB1	Kodeicd, bulan, tahun	

c. Sistem Informasi Rumah Sakit

Integrasi antara sistem informasi Rumah sakit dengan Sistem informasi Dinkes, berikut adalah arsitektur integrasi dari kedua sistem diatas.



Gambar 2: Arsitektur integrasi sistem Rumah sakit dengan sistem Dinkes

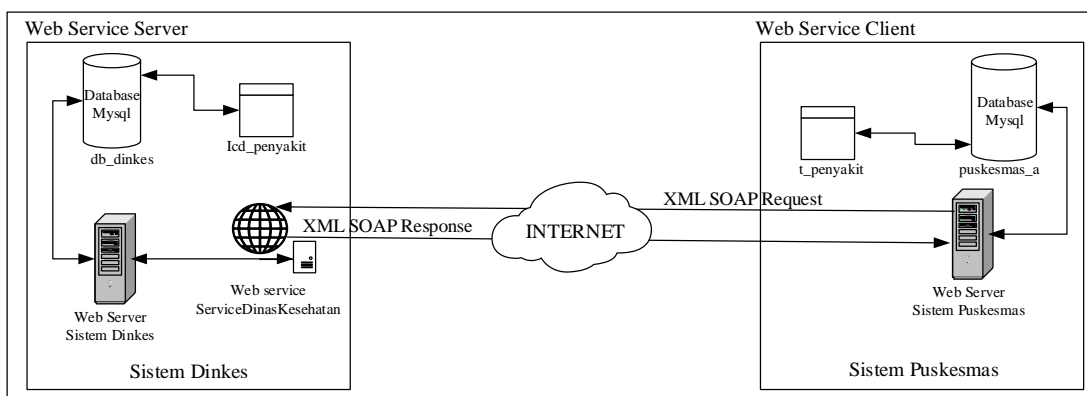
Berikut adalah service yang diberikan Sistem Rumah Sakit kepada Dinas Kesehatan:

Tabel 4. Perancangan web service sistem Rumah sakit

Method Web Service	Parameter	Url WSDL
getLaporanW1RS	Tanggal (yyyy-mm-dd)	http://localhost/fik/rs/ws/laporan_rs.php?wsdl
getLaporanW2RS	Tanggal (yyyy-mm-dd)	
getLaporanLB1RS	TahunBulan (yyyy-mm)	
LaporanPenderitaPenyakitW1	Kodeicdpenyakit, Tanggal (yyyy-mm-dd)	
LaporanPenderitaPenyakitW2	Kodeicdpenyakit, Tanggal (yyyy-mm-dd)	
LaporanPenderitaPenyakitLB1	Kodeicdpenyakit, tahun, bulan	

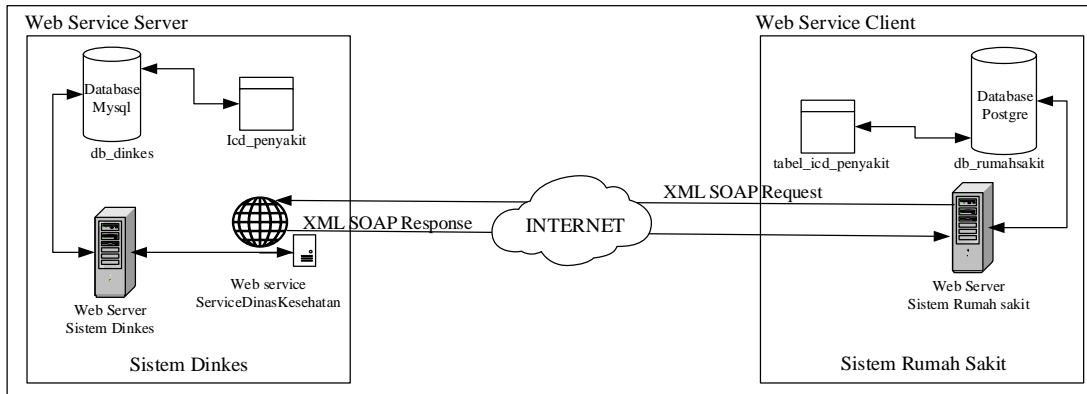
d. Sistem Informasi Dinas Kesehatan

Sistem informasi Dinas Kesehatan akan memberikan service ke Puskesmas dan Rumah Sakit, Berikut ini adalah arsitektur integrasi antara sistem Dinas Kesehatan dengan sistem Puskesmas.



Gambar 22: Arsitektur Integrasi Sistem Dinas Kesehatan dengan Sistem Puskesmas

Berikut ini adalah arsitektur integrasi antara sistem Dinas Kesehatan dengan sistem Puskesmas.



Gambar 23 : Arsitektur Integrasi Sistem Dinas Kesehatan dengan Sistem Rumah sakit

Berikut adalah service yang diberikan Sistem Rumah Sakit kepada Dinas Kesehatan.

Tabel 5. Perancangan web service sistem Dinas Kesehatan

Method Web Service	Parameter	Url WSDL
getDataICD	-	http://localhost/fik/dinkes/ws/dinkes_service.php?wsdl

5.4. Hasil Rancangan Web Services

Dalam penelitian ini kami buat suatu prototype data center (datawarehouse epidemiologi) yang mendapatkan sumber data epidemiologi dari puskesmas dan rumah sakit. dengan dbms mysql yang . Data center tersebut akan mendapatkan sumber data epidemiologi dari Dalam proses pembuatan XML Web Services terdapat beberapa services atau fungsi-fungsi yang dibuat untuk mengakses database. Services-services tersebut yang nantinya di panggil dan digunakan untuk membangun sistem integrasi laporan epidemiologi puskesmas dan rumah sakit menuju data center epidemiologi dinas kesehatan sebagai basis data terpadu sistem surveilans terpadu (SST). Proses integrasi pertama adalah mengintegrasikan laporan wabah LB1 (format epidemiologi harian puskesmas) dari puskesmas ke data center epidemiologi dinas kesehatan. Pengiriman LB 1 selama ini dilakukan secara manual, sehingga bagian data center bekerja ekstra untuk memasukan data untuk selanjutnya melakukan pengolahan data epidemiologi. Proses pengiriman data ini dilakukan secara harian. Dengan membuat service yang dapat mengintegrasikan data epidemiologi secara otomatis ke data center dinas kesehatan akan memiliki dampak efisiensi dalam pengolahan dan menganalisis data epidemiologi serta terintegrasi dengan unit surveilans yang lain. Desain web services yang di embeded pada sisi web puskesmas dengan fungsi utama send data transaksi LB1 ke data center dinas kesehatan adalah seperti gambar dibawah ini.

```

$server->configureWSDL("Web Service Laporan Puskesmas","urn:DataLaporanPuskesmas");
//----- register SOAP getLaporanLB1PA-----
$server->register('getLaporanLB1PA', // method name
array('TahunBulan'=>'xsd:string'), // input parameters
array('output' => 'xsd:Array'), // output parameters
'urn:LaporanLB1Puskesmas', // namespace
'urn:LaporanLB1Puskesmas#getLaporanLB1PA', // soapaction
'rpc', // style
'encoded', // use
'Service Laporan LB1 dari Puskesmas' // documentation
);
function getLaporanLB1PA($TahunBulan){
try {
include_once 'conn.php';
$preparedStatement=$koneksi->prepare("SELECT t_stat_penyakit.kode_penyakit,nama_penyakit,COUNT(*) AS jml_penyakit
FROM t_stat_penyakit JOIN t_penyakit WHERE t_penyakit.kode_penyakit=t_stat_penyakit.kode_penyakit
AND tgl_kejadian LIKE '%$TahunBulan%' GROUP BY kode_penyakit ORDER BY jml_penyakit DESC");
$preparedStatement->execute();
$res=$preparedStatement->fetchAll();
foreach ($res as $data) {
$Lap[]=array('kodepenyakit'=>$data['kode_penyakit'],
'nama_penyakit'=>$data['nama_penyakit'],
'jmlpenyakit'=>$data['jml_penyakit']
);
}
$koneksi=null;
return $Lap;
catch (PDOException $e) {
echo $e->getMessage();
}
}
//----- getPenderitaLB1 -----
$server->register('getPenderitaLB1', // method name
array('kodeicd'=>'xsd:string','bulan'=>'xsd:int','tahun'=>'xsd:int'), // input parameters
array('output' => 'xsd:Array'), // output parameters
'urn:DataPenderitaPenyakitLB1', // namespace
'urn:DataPenderitaPenyakitLB1#getPenderitaLB1', // soapaction
'rpc', // style
'encoded', // use
'Service Penderita Penyakit hasil diagnosa, parameter input kode icd, bulan (int), tahun (int) ' // documentation
);
function getPenderitaLB1($kodeicd,$bulan,$tahun){
try {
include_once 'conn.php';
$preparedStatement=$koneksi->prepare("SELECT t_stat_penyakit.kode_penyakit,nama_penyakit,COUNT(*) AS jml_penyakit
FROM t_stat_penyakit JOIN t_penyakit WHERE t_penyakit.kode_penyakit=t_stat_penyakit.kode_penyakit
AND tgl_kejadian LIKE '%$tahun%' AND tgl_kejadian >= '$bulan-01-2014' AND tgl_kejadian < '$tahun-12-31-2014'
GROUP BY kode_penyakit ORDER BY jml_penyakit DESC");
$preparedStatement->execute();
$res=$preparedStatement->fetchAll();
foreach ($res as $data) {
$Lap[]=array('kodepenyakit'=>$data['kode_penyakit'],
'nama_penyakit'=>$data['nama_penyakit'],
'jmlpenyakit'=>$data['jml_penyakit']
);
}
$koneksi=null;
return $Lap;
catch (PDOException $e) {
echo $e->getMessage();
}
}
}

```

Gambar 24. Dokumen WSDL Services Mengambil Data LB1 Puskesmas Untuk Di Kirim Ke Dinkes
Desain web services pengiriman data laporan epidemiologi harian dari rumah sakit ke data center epidemiologi dinas kesehatan pada prinsipnya adalah sama dengan yang didesain pada sisi web puskesmas. Perbedaannya adalah pada format pelaporan, format yang dipakai diberi nama W1 harian. Desain web services tersebut adalah seperti gambar dibawah ini :

```

$server->configureWSDL("Web Service Laporan Rumah Sakit","urn:DataLaporanRumahSakit");
//register SOAP getLaporanKDPA
$server->register('getLaporanW1RS', // method name
array('tanggal'=>'xsd:string'), // input parameters
array('output' => 'xsd:Array'), // output parameters
'urn:LaporanW1RumahSakit', // namespace
'urn:LaporanW1RumahSakit#getLaporanW1RS', // soapaction
'rpc', // style
'encoded', // use
'Service Laporan W1 dari Rumah Sakit' // documentation
);
function getLaporanW1RS($tanggal){
try {
include_once 'conn.php';
$preparedStatement=$koneksi->prepare("SELECT tabel_surveillance.kodeicdpenyakit, tabel_icd_penyakit.namaicdpenyakit, COUNT(*) AS
jml_penyakit
FROM tabel_surveillance
JOIN tabel_icd_penyakit
ON tabel_icd_penyakit.kodeicdpenyakit=tabel_surveillance.kodeicdpenyakit
WHERE tanggalkejadian = '$tanggal'");
$preparedStatement->execute();
$res=$preparedStatement->fetchAll();
foreach ($res as $data) {
$Lap[]=array('kodeicd'=>$data['kodeicdpenyakit'],
'namaicd'=>$data['namaicdpenyakit'],
'jmlpenyakit'=>$data['jml_penyakit']
);
}
$koneksi=null;
return $Lap;
catch (PDOException $e) {
echo $e->getMessage();
}
}
}

```

Gambar 25: Dokumen WSDL Services Mengambil Data W1 Rumah Sakit Untuk Dikirim Ke Dinkes

Sedangkan services penerimaan data LB1 dan W1 pada sisi webservice di dinas kesehatan atas respon pengiriman data atas permintaan services tersebut adalah sebagai berikut :

```

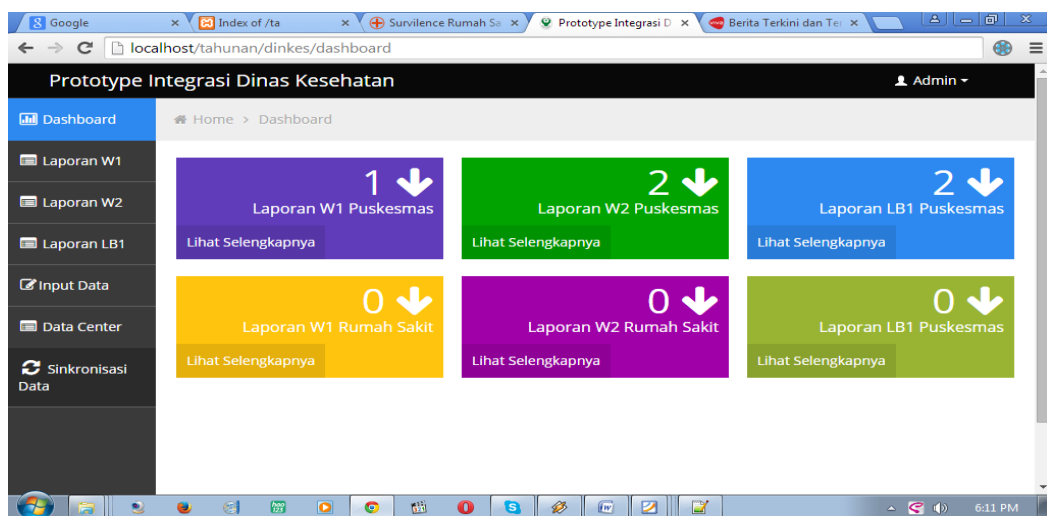
class Lihatsoap extends CI_Controller{
public function __construct(){
    parent::__construct();
    $this->load->library("nuSoap_lib");
    $this->load->model("tanggal_model");
    $this->load->model("validation_model");
}
//method untuk melihat xml soap service w1 dari Puskesmas
public function SoapW1Puskesmas($tanggal){
// service data penduduk service laporan w1 dari puskesmas
    $this->nusoap_client = new nusoap_client("http://localhost/taunan/puskesmasA/ws/puskesmas_service.php?wsdl");
    if($this->nusoap_client->fault)
    {
        $text = 'Error: '.$this->nusoap_client->fault;
    }
    else
    {
        if ($this->nusoap_client->getError())
        {
            $text = 'Error: '.$this->nusoap_client->getError();
        }
        else
        {
            $row['data'] = $this->nusoap_client->call('getLaporanKDPA',array('tanggal'=>$tanggal));
            $row['request']=$this->nusoap_client->request;
            $row['respon']=$this->nusoap_client->response;
        }
    }
    $row['tanggal']=$this->tanggal_model->konversiTanggal($tanggal);
    $row['tgl']=$tanggal;
        $this->load->view('menu_view');
        $this->load->view('lihatsoap/soap_w1_puskesmas_view',$row);
        $this->load->view('footer_view');
    }
//method untuk melihat xml soap service w1 dari Rumah Sakit
public function SoapW1RumahSakit($tanggal){
// service data penduduk service laporan w1 dari puskesmas
    $this->nusoap_client = new nusoap_client("http://localhost/taunan/rs/ws/laporan_rs.php?wsdl");
    if($this->nusoap_client->fault)
    {
        $text = 'Error: '.$this->nusoap_client->fault;
    }
    else
    {
        if ($this->nusoap_client->getError())
        {
            $text = 'Error: '.$this->nusoap_client->getError();
        }
        else
        {
            $row['dataw1'] = $this->nusoap_client->call('getLaporanWIRS',array('tanggal'=>$tanggal));
            $row['request']=$this->nusoap_client->request;
            $row['respon']=$this->nusoap_client->response;
            // $this->load->view('aritmatika_client_view',$row);
        }
    }
    $row['tanggal']=$this->tanggal_model->konversiTanggal($tanggal);
    $row['tgl']=$tanggal;
    $this->load->view('menu_view');
    $this->load->view('lihatsoap/soap_w1_rumahsakit_view',$row);
    $this->load->view('footer_view');
}
}

```

Gambar 26. Dokumen WSDL Services Penerimaan Data Dari Respon Web Services Puskesmas Dan RS

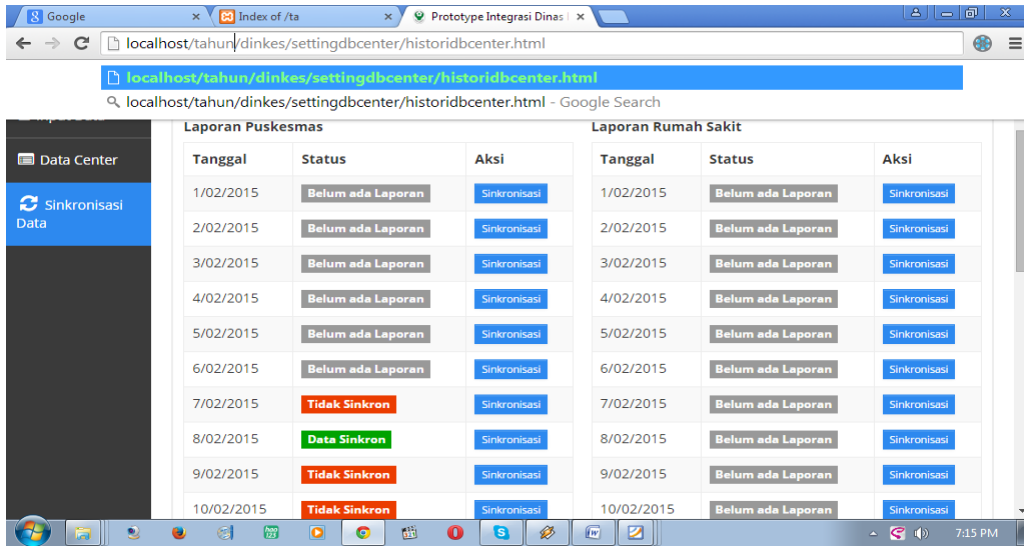
5.5. Hasil Perancangan Interface Integrasi Unit Surveilans Dengan Data Center

Prototype integrasi data center yang di kembangkan pada dinas kesehatan merupakan integrasi dua unit surveilans yaitu puskesmas dan rumah sakit. Dengan mendesain XML Web Services data dari dua unit surveilans dapat dapat disatukan menjadi sistem yang terintegrasi walaupun bersumber dari aplikasi dan dbms yang berbeda (multi platform). Gambar dibawah ini menunjukkan informasi jumlah data laporan epidemiologi yang bersumber dari puskesmas dan rumah sakit.



Gambar 27: Desain Halaman Informasi Data Center Epidemiologi Dinkes

Untuk memonitoring pengiriman data epidemiologi dari unit surveilan dapat dilakukan sinkronisasi data kiriman dari masing-masing unit surveilan setiap hari atau per tanggal. Dengan sistem monitoring harian ini akan dapat memastikan unit surveilans mana saja yang tidak melakukan pengiriman data epidemiologi sehingga bisa dilakukan tindakan tertentu, hal ini sangat penting untuk melakukan monitoring wabah terutama untuk wabah penyakit menular.



Gambar 28. Halaman Monitoring Pelaporan Data Epidemiologi

BAB 6

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Berasarkan tahapan pelaksanaan penelitian yang sudah peneliti lakukan untuk mencapai hasil dalam bentuk *prototype* data warehouse dari integrasi antar sistem yang heterogen yaitu integrasi dari sistem Kependudukan, sistem Rumah sakit, sistem Puskesmas dan sistem Dinkes untuk pengiriman data penduduk, data penyakit, data penderita penyakit dan data ICD penyakit, maka peneliti dapat menyimpulkan diantaranya :

- a. Dengan memanfaatkan metode *web service* SOAP, maka integrasi data diantara sistem yang heterogen tidak perlu mengubah sistem yang sudah ada meskipun diantara sistem memiliki konflik pada penggunaan DBMS yang berbeda, penamaan tabel yang berbeda dan struktur atribut tabel yang berbeda.
- b. Penggunaan *web service* sebagai solusi antar sistem yang heterogen lebih mudah karena tidak perlu mengubah sistem yang sudah ada meskipun ada konflik heterogenitas.
- c. Integrasi antar system yang memiliki platform heterogen akan dapat menyatukan system yang terfragmentasi menjadi system yang terintegrasi

6.2. Saran

Namun demikian untuk meningkatkan kinerja serta menyempurnakan penelitian sistem integrasi yang akan peneliti dibuat, peneliti memberikan saran sebagai berikut:

- a. Untuk pengembangan perlu dibuat *service by demand* yaitu penggunaan *service* ketika diperlukan saja, sehingga tidak membutuhkan koneksi internet yang terus menerus (kebutuhan koneksi internet diperlukan sesuai dengan kebutuhan saja)
- b. Untuk integrasi dan sinkronisasi dilakukan secara *asynchronous* dengan program yang berjalan pada *background process* pada sistem operasi, sehingga integrasi tidak satu arah tetapi memiliki kestaraan integrasi.
- c. Untuk perlu diadakan uji standar keamanan *web service* SOAP untuk pengamanan dokumen XML hasil komunikasi antara *server* dan *client*.

DAFTAR PUSTAKA

- [1] Osvalds, G (2001). *Definition of Enterprise Architecture – Centric Models for The System Engineers*, TASC Inc.
- [2] Almeida, J. P. dkk.. (2009). Model-Driven Service-Oriented Architectures. *International Journal Business Process Integration and Management*, Vol.4, No.1, 2-4.
- [3]. Sarno, R. and Herdiyanti A (2010). “ A Service Portofolio for an Enterprise Resource Planning” ; *International Jurnal of Computer Science and Network Scurity*, ISSN -1738 – 7906, Vol. 10, No. 3, app. 144-156.
- [4] Larrucea, Xabier dkk.. (2007). MDSOA for Achieving Interoperability. *Sixth International IEEE Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems (ICCBSS'07)*, 247.
- [5] ATHENA. (2005). Deliverable DA1.3.1 – Report on Methodology description and guidelines definition. *Integrating and Strengthening the European Research in Advance Technologies for Interoperability of Heterogeneous Enterprise Networks and their Application*.
http://interop-lab.eu/ei_public_deliverables/athena-deliverables/A1/d-a1-3.1, diakses 20 Maret 2012.
- [6] Benguria, Gorka dkk.. (2007). A Platform Independent Model for Service Oriented Architecture. *International Conference on Interoperability of Enterprise Software and Applications 2006*, 23-32. Bordeaux: Springer.
- [7] IEEE. (1998). *IEEE Recommended Practice for Software Requirements Specifications*. NJ: IEEE.
- [8] Mahfudhi, Muhammad Ghufron. (2012). *Kajian dan Evaluasi Pengembangan Pedoman Model Driven Service Oriented Architecture, Studi Kasus: Rekam Medis Rawat Jalan*.
- [9] Davis, F. D. , Perceived usefulness, perceived ease of use, and user acceptance of information technology, *MIS Quarterly*, Vol.13 (3), pp. 319-340.
- [10] DepKes RI. *Bentuk Pokok Penyelenggaraan Sistem Kesehatan, Nasional*. Jakarta, 2002.
- [11] DeLone W. & E.R. Mclean, *The DeLone and McLean Model of Information Systems*, 2003.
- [12] Departemen Kesehatan RI. *Kumpulan Indikator Kesehatan Arti dan Manfaatnya*. Jakarta, 2008.
- [13] LPIU, MMRS. *Makalah Seminar Sehari “Menuju Komputerisasi Rekam Medis Rumah Sakit”*, Universitas Gajah Mada, Yogyakarta, 1999.
- [14] Marsuli. *Mutu Pelayanan Pasien Rawat Jalan*. *Jurnal Manajemen Pelayanan Kesehatan* vol 08/Nomor 01/ Maret/ 2005, Fakultas Kedokteran Universitas Gajah Mada, Yogyakarta.

LAMPIRAN - LAMPIRAN

Lampiran 1. Instrumen

No	Jenis Luaran		Indikator Capaian		
			TS	TS+1	TS+2
1.	Publikasi Ilmiah	Internasional	Draft	Submitted	
		Nasional Terakreditasi			
2.	Pemakalah dalam temu ilmiah	Internasional			
		Nasional	Sudah dilaksanakan	Sudah dilaksanakan	
3.	<i>Invited Speaker</i> dalam temu ilmiah	Internasional			
		Nasional			
4.	<i>Visisting Lecturer</i>	Internasional			
5.	Hak Kekayaan Intelektual	Paten			
		Paten Sederhana			
		Hak Cipta	Terdaftar	Granted	
		Merek Dagang			
		Rahasia Dagang			
		Desain Produk Industri			
		Indikasi Geografis			
		Perlindungan Varietas Tanaman			
		Perlindungan Sirkuit Terpadu			
6.	Teknologi Tepat Guna		Desain Produk	Prototype produk	
7.	Model/Purwarupa/Desain/Karya Seni/Rekayasa Sosial		Desain Model	Purwarupa	
8.	Buku Ajar (ISBN)				
9.	Tingkat Kesiapan Teknologi (TKT)		5	6	

Lampiran 2. Personalia Tenaga Pelaksana Beserta Kualifikasinya

No	Nama/NIDN	Instansi Asal	Bidang Ilmu	Alokasi Waktu (jam / minggu)	Uraian Tugas
1.	Fikri Budimn M.Kom/0604047201	Fak. Ilmu Komputer UDINUS Semarang	Data Mining	8 Jam	<p>Ketua Peneliti</p> <ol style="list-style-type: none"> 1. Mengkoordinir tugas anggota peneliti 2. Mengkoordinasikan dan mengkomunikasikan dengan obyek penelitian dan pihak eksternal yang terkait 3. Menentukan metode penelitian 4. Melakukan Analisis dan Desain Konseptual Datawarehouse. 4. Melakukan Identifikasi dan Spesifikasi Kebutuhan Sistem. integrasi data Puskesmas-RS 5. Merumuskan Hipotesa 6. Menguji, Memantau dan mengendalikan pelaksanaan penelitian 7. Melaporkan hasil penelitian 8. Mempublikasikan
2.	Muslih, M.kom /0604057501	Fak. Ilmu Komputer UDINUS Semarang	Data Base	6 Jam	<p>Anggota 1</p> <ol style="list-style-type: none"> 1. Bendahara dan administrasi 1. Menyusun Kerangka Sinkronisasi Puskesmas, RS dan Dinkes 2. Menyusun Kebutuhan Integrasi Epidemiologi. 3. Melakukan Evaluasi Penyebaran Penyakit. 4. Menyusun Skenario Implementasi Data Center
3.	Tim Teknis M.Zainu		Tim Teknis Programing dan Jaringan Komputer	5 Jam	<p>Tim Teknis</p> <ol style="list-style-type: none"> 1. Menyusun Database Data Center 2. Programming 3. Hitung dan analisa pola sinkronisasi dan integrasi antar unit surveilans 4. Evaluasi dan Refisi Fungsi modul Epidemiologi 5. Menyusun requirement spesifikasi fungsi dan sinkronisasi database atar unit surveilans.

DESAIN MODEL INTEGRASI DAN SINKRONISASI ANTAR UNIT SURVEILANS UNTUK Mendukung DATA WAREHOUSE EPIDEMIOLOGI

Fikri Budiman¹, Slamet Sudaryanto N², Muslih³

¹ Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Semarang - Jawa Tengah

Email: fikribudiman01@gmail.com¹, slametalica301@dsn.dinus.ac.id², muslih@gmail.com³

ABSTRACT

Monitoring data wabah penyakit pada wilayah dan populasi tertentu (epidemiologi) memerlukan dukungan data yang terintegrasi dari setiap unit surveilans yang terkait. Masalah monitoring antar unit surveilans saat ini masih dilakukan secara parsial, hal ini dikarenakan tidak ada keseragaman aplikasi dan database. Namun dengan perkembangan teknologi web services dan XML ketidak seragaman aplikasi dan database (heterogen) bisa diatasi dengan melakukan desain interoperabilitas dan inter koneksi antar aplikasi melalui jaringan computer. Integrasi data antar unit surveilans (puskesmas, poliklinik, rumah sakit) harus dikelola dan didesain dengan baik sehingga memungkinkan pimpinan dan para analis kesehatan untuk memperoleh, mengintegrasikan, menganalisis dan memonitoring data (kasus penyakit) dari sumber data yang berbeda. Sumber data tersebut bersumber dari system yang heterogen, dalam hal ini sumber data tersebut adalah unit surveilans. Untuk memudahkan penegelolaan data surveilans tersebut akan di desain data center dalam model data warehouse epidemiologi sehingga membentuk system surveilans terpadu (SST). Permasalahan yang dihadapi adalah berkaitan dengan interoperabilitas, yaitu kemampuan untuk mengintegrasikan dan mensinkronisasi data yang bersumber dari system yang berbeda platform (heterogen). Dengan demikian diperlukan suatu metodologi pengintegrasian data dalam model XML kedalam data warehouse epidemiologi tersebut. Sejak XML menjadi standar untuk pertukaran data melalui internet, terutama didalam komunikasi B2B dan B2C maka membutuhkan integrasi data XML kedalam system data warehouse. Didalam penelitian ini kami menjelaskan desain model integrasi dan sinkronisasi antara unit surveilans berdasarkan services oriented architecture (SOA) berbasis arsitektur web services. Metode yang digunakan adalah XML Web Services, yaitu metode yang dapat mengintegrasikan aplikasi dan pertukaran data dalam format XML (Extensible Markup Language). Pertukaran data dalam format XML menggunakan teknologi SOAP (Simple Object Acces Protocol) dan WSDL (Web Services Description Language) serta menggunakan library NuSOAP. Kelebihan dari metode ini adalah kemampuan mengintegrasikan dan mengintegrasikan data yang bersumber dari berbagai aplikasi yang tidak seragam, sehingga tidak memerlukan restrukturisasi data existing. Hasil uji penelitian dengan metode ini adalah dapat terbentuknya data center epidemiologi secara real time dari berbagai aplikasi yang berbeda dariberbagai unit surveilans yang heterogen melaluidesain webservices.

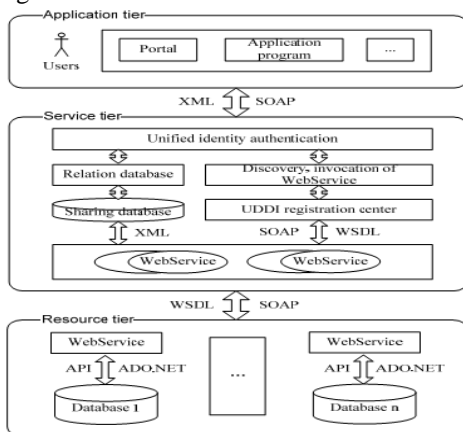
Keywords : Epidemiologi, Data Warehouse, Integrasi, Sinkronisasi, XML, SOA, Web Services, WSDL.

1 PENDAHULUAN

Desain Data Warehouse Epidemiologi merupakan strategi yang memiliki kontribusi sangat penting dalam meningkatkan kualitas dan mutu monitoring data kesehatan. Surveilans epidemiologi adalah rangkaian proses yang sistematis dan berkelanjutan dalam pengumpulan, analisa dan interpretasi data kesehatan. Program pencegahan dan pemberantasan penyakit akan sangat efektif jika didukung oleh Data Warehouse Epidemiologi, karena salah satu fungsi data warehouse ini adalah ketersediaan data

epidemiologis yang bersumber dari semua unit surveilans secara terintegrasi dan up to date. Dengan Sistem Data Warehouse maka mutu surveilans epidemiologi akan selalu terjaga. Sistem data warehousing epidemiologi adalah seperangkat teknologi dan alat yang memungkinkan sumber data epidemiologi dari unit surveilans (puskesmas, poliklinik, rumah sakit) yang tersebar dan bersifat heterogen dapat melakukan interoperabilitas (komunikasi data) khususnya dalam hal integrasi dan sinkronisasi data. Bagian terpenting dari system data

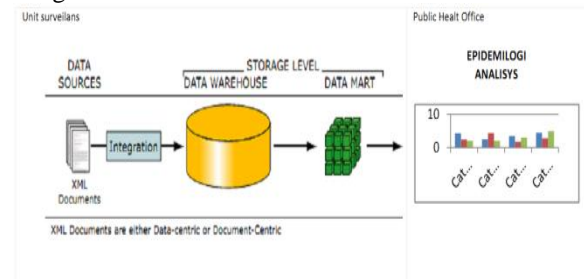
warehouse ini adalah data mart, data mart merupakan koleksi histori data sumber (local schema) yang berorientasi pada kebutuhan analisis epidemiologi khusus. Proses desain system data warehouse terdiri dari beberapa tahapan seperti analisis data heterogen dari sumber data unit surveilans, desain model data warehouse, definisi transformasi dan proses integrasi, desain sinkronisasi dan implementasi peralatan yang dibutuhkan oleh user untuk mengakses dan merepresentasikan data warehouse. WebServices merupakan metode yang paling populer dari SOA saat ini [1]. Memiliki seri kroteria dan protocol yang menjamin fungsi yang haasrus dilakukan, seperti WebServices Description Language (WSDL), UDDI, Simple Object Access Protocol (SOAP) dan sebagainya[2]. Web services merupakan platform independen dan cocok untuk lingkungan yang heterogen.



Gambar 1 SOA-based WebServices Architecture

SOA merupakan metode integrasi terkemuka dan framework arsitektur dilingkungan system heterogen [3]. Ide dasar dari SOA adalah services cored dan mengintegrasikan sumberdaya informasi pada layanan standar, memungkinkan rekonfigurasi dan pemakaian kembali sumberdaya informasi [5]. SOA adalah kelonggaran services yang dapat di gabungan dengan arsitektur layanan [6,7]. Layanan komunikasi dengan sederhana dan ketepatan pendefinisian interface yang tidak melibatkan interface pemrograman dan model komunikasi. Ada tiga bagian SOA berbasis arsitektur web services : services provider, services registry, services consumer [8]. Penyedia jasa dan layanan konsumen masing-masing mengacu kepada pada system eksternal dan system internal korporasi. Sebagai solusi dapat dibagi menjadi tiga lapis arsitektur, yaitu application tier, services tier and resource tier. Resource tier contains all sorts of database of the MIS wich need to be integrated. Services tier is the core of the solution, including web services encapsulation, composition, registration, discovery, invocation and data mapping module. Application tier consists of the application client of MIS ind information portal. XML is an emerging internet application oriented markup language for documents

contains both content and structure. Xml is extensible, platform independent and easy to transfer in network, so it is very suitable for information describing and data synchronaization and database integration.



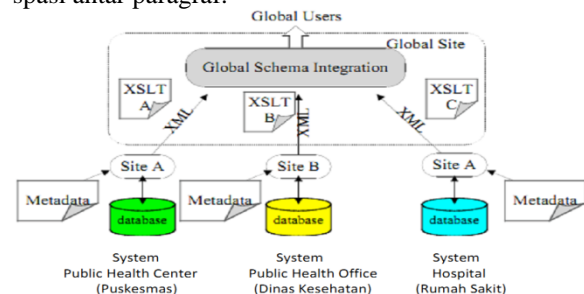
Gambar 2 Integration of XML document within a datawarehouse epidemiologi

XML Web Services adalah sebuah system perangkat lunak yang dirancang untuk mendukung interoperabilitas dalam interaksi mesin ke mesin dalam sebuah jaringan [9]. Interaksi dilakukan melalui protocol tertentu. Dengan demikian kemampuan web services dapat menambah kemampuan web untuk saling berkomunikasi dan bertukar informasi dan data dengan pola program-to-program. Dengan desain XML web services akan dapat mengintegrasikan system, bahasa pemrograman, basis data dan platform system operasi yang berbeda dengan protocol http (Hypertext Transfer Protocol). Dengan demikian aplikasi dan dbms yang heterogen antar unit surveilans dapat berkomunikasi dengan system data warehouse epidemiologi.

Tujuan dari penelitian ini adalah untuk melakukan analisis, perancangan, integrasi dan sinkronisasi antar system unit surveilans dengan system dinas kesehatan dalam model prototype data warehouse epidemiologi.

2 MODEL, ANALISIS, DESAIN, DAN IMPLEMENTASI

Bab 2 menggambarkan tentang model, teori, hipotesis, analisis, atau desain dan implementasi. Makalah harus ditulis dengan satu spasi dan tanpa spasi antar paragraf.



Gambar 3 Model arsitektur Integrasi Database Heterogen

2.1 Problem Integrasi Data Heterogen

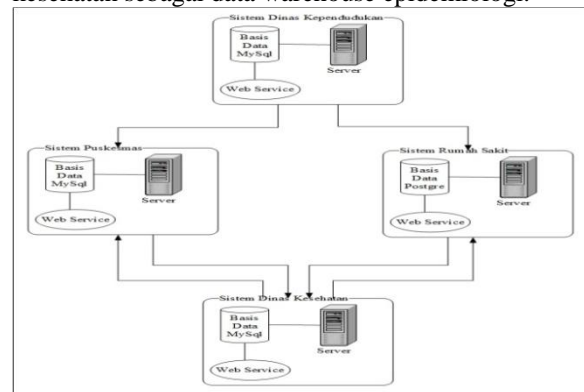
Integrasi adalah proses menggabungkan data yang berbeda di sumber data yang berbeda, dan menyediakan pengguna dengan pandangan yang seragam terhadap data tersebut[11]. Pada dasarnya proses ini melalui pembangunan system yang terintegrasi. Lazerini [11] karakteristik system integrasi data sebagai arsitektur berdasarkan skema global dan sekumpulan data sumber. Sumber data berisi data riel, sementara skema global menyediakan rekonsiliasi, integrasi, dan virtual view dari dasar sumber data. Dengan global virtual global view, pengguna system terintegrasi dapat mendapatkan akses data yang seragam dari sumber data yang berbeda. Dua pendekatan dasar yang digunakan diantara model skema global dan sumber data (skema local). Yang pertama Global-as-View (GaV) dimana global skema sebagai sumber data [12]. Yang kedua adalah Local-as-View (LaV) dimana setiap sumber data didefinisikan sebagai view diatas global schema. Beberapa permasalahan integrasi yang kaitan dengan heterogenitas data antar sumber data sebagai skema local dengan skema global adalah *schema mapping, data cleansing, data transformation, data reconciliation problem*.

2.2 Problem Sinkronisasi Data Heterogen

Sinkronisasi data heterogen berkaitan langsung dengan proses replikasi data antar database yang heterogen. Replikasi data adalah proses menghasilkan, memproduksi, dan memelihara sekumpulan data dalam satu atau lebih lokasi[13]. Replikasi data antar database sangat penting terhadap ketersediaan data yang up-to-date kapanpun dan dimanapun data dibutuhkan. Dalam hal komunikasi dan ketepatan waktu proses replikasi data dapat menjadi sinkron atau asinkron. Proses sinkronisasi data adalah mentransformasikan status data saat ini dari system aplikasi ke system heterogen lain dan memperbarui data untuk menjaga konsistensi data. Sinkronisasi data sebagai “a technical method that can achieve data consistency from multi-database in heterogenous platform”. Sinkronisasi juga merupakan kebutuhan krusial integrasi data untuk menyediakan real-time consistency antara data source local dan global melalui mekanisme update. Dengan demikian, sinkronisasi data berarti jika ada perubahan data pada satu sumber data local akan diberitahu atau disebar ke datasource yang sesuai dengan pengelola domain informasi. Oleh karena itu perubahan lingkungan yang terintegrasi akan otomatis mengupdate anatar system. Sinkronisasi dapat diklasifikasikan menjadi uni-directional data synchronization and bi-directional data synchronization. Beberapa permasalahan yang berkaitan dengan sinkronisasi data adalah komunikasi antar proses terhadap pemakaian data dan sumberdaya yang bersamaan.

3 SKENARIO UJI COBA

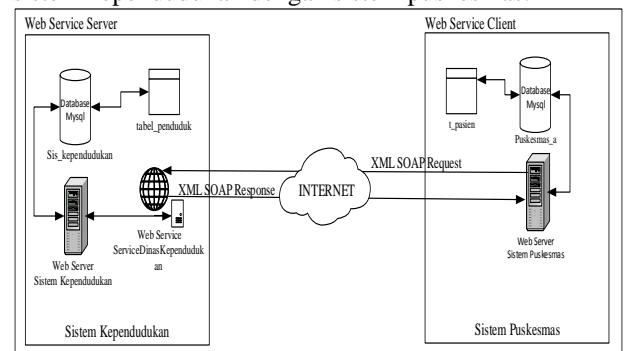
Dalam rancangan untuk model integrasi antar aplikasi yang akan dibangun dengan webs services ini merupakan integrasi data epidemiologi dari unit surveilans puskesmas dan rumah sakit. Dimana unit surveilans yang menggunakan dbms dan aplikasi yang berbeda (My Sql dan PostgreSql). Akses yang akan dibangun dalam komunikasi anatar aplikasi terjadi secara dua arah, dimana melalui web services masing masing unit surveilans akan mengambil data epidemiologi dari database local dirubah kedalam bentuk format dokumen dalam parameter tertentu (SOAP Services). Kemudian web service dengan akses send and request antar aplikasi akan meneruskan ke database epidemiologi dinas kesehatan sebagai data warehouse epidemiologi.



Gambar 4 Desain model integrasi data antar unit surveilans dalam data warehouse epidemiologi

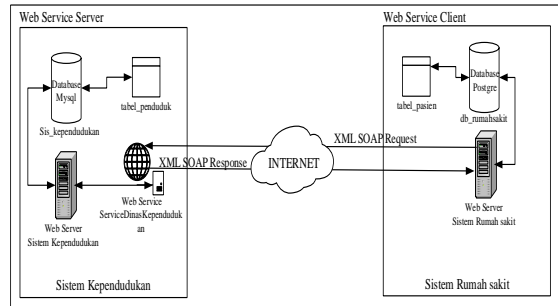
- Integrasi antara system dinas kependudukan dengan system puskesmas.

Sistem informasi dari dinas kependudukan akan memberikan service ke puskesmas dan rumah sakit, Berikut ini adalah arsitektur integrasi antara sistem kependudukan dengan sistem puskesmas.



Gambar 5 Arsitektur integrasi web services antara sistem kependudukan dengan sistem puskesmas

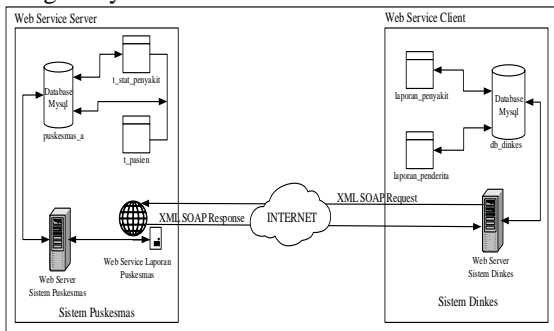
Sistem Kependudukan juga memberikan service terhadap system rumah sakit agar terintegrasi atar keduanya, berikut ini adalah arsitektur integrasi antara sistem Kependudukan dengan sistem Rumah sakit.



Gambar 6 Arsitektur integrasi web service antara sistem kependudukan dengan sistem rumah sakit

b. Integrasi Antara Sistem Puskesmas Dengan Sistem Dinas Kesehatan

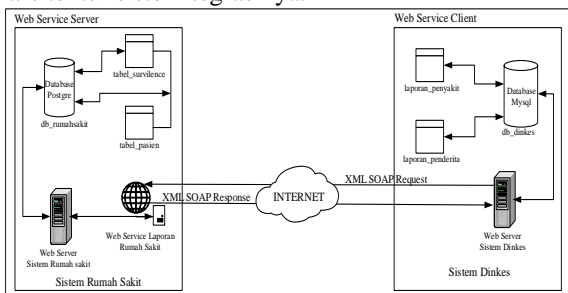
Sistem informasi Puskesmas akan memberikan service kepada dinas kesehatan, agar dinas kesehatan bisamendapat data setiap pencatatan kasus penyakit. Berikut ini adalah arsitektur sistem integrasinya.



Gambar 7 Arsitektur integrasi web services sistem puskesmas dengan sistem dinas kesehatan

c. Integrasi antara sistem rumah sakit dengan system dinas kesehatan.

Integrasi antara sistem informasi rumah sakit dengan sistem dinas kesehatan memungkinkan dinas kesehatan mematau ddata epidemiologi yang bersumber dari rumah sakit, berikut ini adalah arsitektur siste integrasinya.

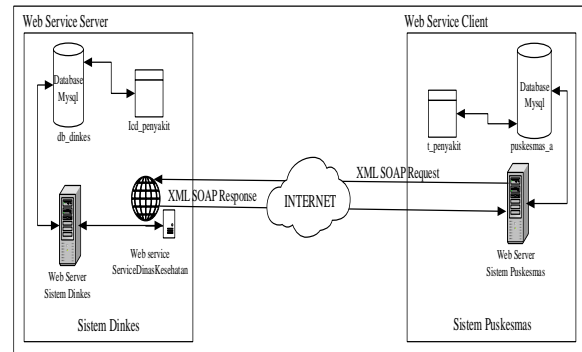


Gambar 8 Arsitektur integrasi web service sistem rumah sakit dengan sistem dinas kesehatan

d. Integrasi antara sistem puskesmas dengan system dinas kesehatan.

Integrasi sistem dinas kesehatan akan memberikan service ke system puskesmas, sehingga data surveilan epidemiologi setiap kasus penyakit akan dapat di monitor oleh dinas kesehatan. Berikut

ini adalah arsitektur integrasi antara sistem dinas kesehatan dengan sistem puskesmas.



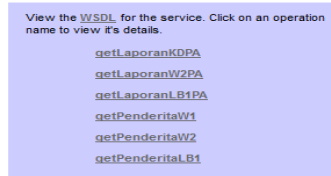
Gambar 9 Arsitektur integrasi web services sistem dinas kesehatan dengan sistem puskesmas

4 HASIL UJI COBA

Setelah tahap desain model integrasi web services antar beberapa unit surveilans, maka tahapan selanjutnya adalah melakukan implemetasi pemrograman XML pada masing masing desain web services yang terkait dengan system integrasi antar unit surveilans yang terkait. Beberapa desain web service s tersebut adalah sebagai berikut :

a. Web services system pelaporan puskesmas berintegrasi dengan system dinas kesehatan.

Web Service Laporan Puskesmas



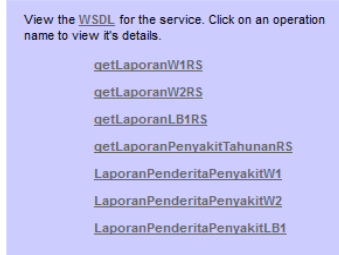
Gambar 10 Web service Sistem Puskesmas

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsdl="http://www.w3.org/2001/XMLSchema"
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:DataLaporanPuskesmas"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="urn:DataLaporanPuskesmas">
<types>
<xsd:schema targetNamespace="urn:DataLaporanPuskesmas">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
</xsd:schema>
</types>
<message name="getLaporanKDPAREquest">
<part name="tanggal" type="xsd:string" /></message>
<message name="getLaporanKDPAResponse">
<part name="output" type="xsd:Array" /></message>
<input><soap:body use="encoded" namespace="urn:DataPenderitaPenyakitLBI"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></input>
<output><soap:body use="encoded"
namespace="urn:DataPenderitaPenyakitLBI"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></output>
</operation>
</binding>
</service name="Web Service Laporan Puskesmas">
<part name="Web Service Laporan PuskesmasPort" binding="tns:Web Service
Laporan PuskesmasBinding">
<soap:address
location="http://localhost/ta/puskesmas/ws/puskesmas_service.php"/>
</port>
</service>
</definitions>
```

Gambar 11 Example WSDL web services sistem puskesmas

b. Web services system pelaporan rumah sakit terintegrasi dengan system dinas kesehatan

Web Service Laporan Rumah Sakit



Gambar 12 Web services sistem rumah sakit

c. Mapping Rules Schema

Merupakan teknik integrasi data antar skema yang heterogen terutama berhubungan dengan masalah transformasi dan pertukaran data. Mapping rules bersifat middle ware sehingga bersifat independent dan tidak terpengaruh dari berbagai platform. Mapping rules schema juga dapat diterapkan secara format schema maupun values hasil query.

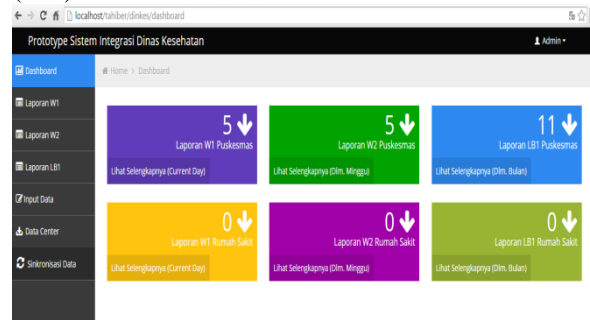
```
// Mapping value schema dari hasil query
// SQLLocRow=>this->db->query("SELECT * FROM laporan_penderita WHERE 'kode_rekam'='&kode_rekam'
// AND 'kode_pasien'='&kode_pasien'
// AND tanggal_kejadian='&tanggal_kejadian'
// AND kode_instansi='&kode_rea'
// AND kode_penyakit='&value[kodepenyakit]'");
$cekDataPasien=$SQLLocRow->num_rows();
if($cekDataPasien==$value['jmlpenyakit']){
    continue;
}
else{
    foreach ($arr['pasienrs'] as $key) {
        $SQLLocRowPasien=>db->query("SELECT * FROM laporan_penderita
        WHERE 'kode_rekam'='&key[kode_rekam]'
        AND 'kode_pasien'='&key[kode_pasien]'
        AND tanggal_kejadian='&tanggal_kejadian'
        AND kode_instansi='&kode_rea'
        AND kode_penyakit='&value[kodepenyakit]'");
        $cekDataPasienDB=$SQLLocRowPasien->num_rows();
        if($cekDataPasienDB==$value['jmlpenyakit']){
            continue;
        }
        else{
            //insert ke tabel laporan data pasien dari hasil query
            $this->db->exec("code_rekam", $key['kode_rekam']);
            $this->db->exec("code_pasien", $key['kode_pasien']);
            $this->db->exec("nama_pasien", $key['nama_pasien']);
            $this->db->exec("alamat_pasien", $key['alamat_pasien']);
            $this->db->exec("dasa_pasien", $key['kelurahan_pasien']);
            $this->db->exec("umur_pasien", $key['umur_pasien']);
            $this->db->exec("status_pasien", $key['statuspasien']);
            $this->db->exec("tanggal_kejadian", $key['tanggal_kejadian']);
            $this->db->exec("kode_instansi", $key['kode_rea']);
            $this->db->exec("kode_penyakit", $value['kodepenyakit']);
            $this->db->insert('laporan_penderita');
            ...
        }
    }
}
```

Gambar 13 Example mapping value schema hasil query

d. Antar Muka Hasil Integrasi Data

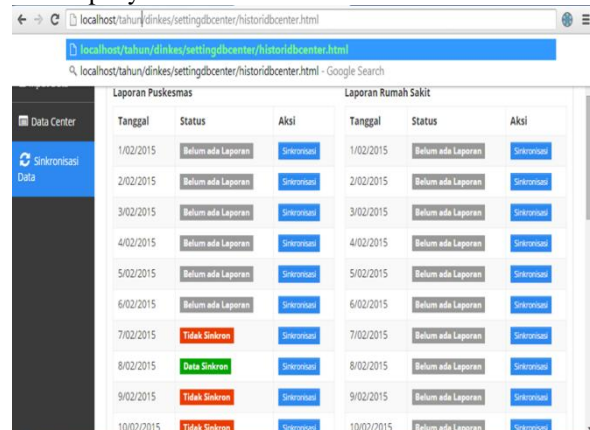
Prototype integrasi data center yang di kembangkan pada dinas kesehatan merupakan integrasi dua unit surveilans yaitu puskesmas dan rumah sakit. Dengan mendesain XML Web Services data dari dua unit surveilans data dapat disatukan menjadi satu sistem yang terintegrasi (dalam model data warehouse) walaupun bersumber dari system aplikasi dan dbms yang berbeda (multi platform). Gambar dibawah ini menunjukkan informasi jumlah data laporan epidemiologis yang bersumber dari unit surveilans (puskesmas dan rumah sakit) baik yang bersifat harian (w1), mingguan (w2) maupun bulanan

(Lb1).

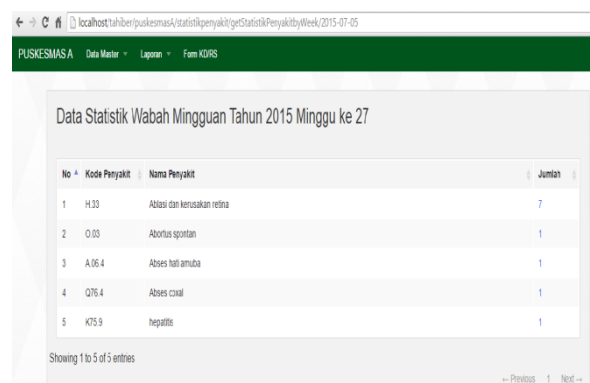


Gambar 15 Monitoring dan analisis hasil dari integrasi data pada data center epidemiologi dinas kesehatan.

Untuk memonitoring pengiriman data epidemiologi dari unit surveilans dapat dilakukan sinkronisasi data kiriman dari masing-masing unit surveilans setiap hari secara terus menerus, sehingga secara otomatis dapat di monitoring. Dengan sistem monitoring harian ini akan dapat memastikan unit surveilans mana saja yang tidak melakukan pengiriman data epidemiologi sehingga bisa dilakukan tindakan tertentu, hal ini sangat penting untuk melakukan monitoring wabah terutama untuk wabah penyakit menular.



Gambar 16 Hasil proses sinkronisasi data



Gambar 17 Monitoring dalam statistik surveilans epidemiologi wabah dari puskesmas tertentu

5 KESIMPULAN

Dalam penelitian ini telah dilakukan

pengembangan desain web services dengan XML dan telah dilakukan pengujian terhadap prototype data center epidemiologi dinas kesehatan yang mana sumber data epidemiologinya berasal dari pelaporan data epidemiologi puskesmas dan rumah sakit dengan format harian (W1), mingguan (W2) dan bulanan (Lb1). Dari hasil pengujian prototype data center epidemiologi tersebut, desain web service berhasil mengintegrasikan sistem pelaporan epidemiologi dari unit surveilans walaupun sumber datanya berasal dari aplikasi dan dbms yang berbeda (MySQL dan PostgreSQL) serta struktur data yang berbeda. Hal ini menunjukkan komunikasi protocol web service yang bersifat middleware mampu melakukan pertukaran data epidemiologi sebagai message dengan memanfaatkan protocol HTTP melalui sebuah jaringan komputer antar aplikasi berbasis web dan database antar unit surveilans dengan database data center epidemiologi dinas kesehatan. Integrasi antar unit surveilans dengan teknologi web services dalam jaringan komputer juga menunjukkan komunikasi antar services provider dan services requester dapat dimanfaatkan untuk monitoring pengiriman data secara real time, sehingga dapat didesain aplikasi monitoring dan sinkronisasi pengiriman data secara efektif. Dengan demikian setiap kejadian wabah dari suatu wilayah tertentu dapat dimonitoring secara terus menerus dalam rangka program pemberantasan dan pengendalian penyakit.

6 DAFTAR PUSTAKA

- [1] Al-Otaibi Noura Meshaan¹, Noaman Amin Yousef, "Biological data integration using SOA", Proceedings of World Academy of Science, Engineering and Technology, WASET, vol.73, pp.920-925, 2011.
- [2] Chung C. Chang, Kou-Chan Hsiao, "A SOA-Based e-Learning System for Teaching Fundamental Information Management Courses", Journal of Convergence Information Technology, Advanced Institute of Convergence Information Technology, vol.6, no.4, pp.298, 2011.
- [3] Karande, Aarti M., "Working of web services using BPEL workflow in SOA", 2nd International Conference on Advances in Computing, Communication and Control, pp.143-149, 2011.
- [4] Gouwei W, Guangming X., "research On Data Synchronization and Integration Platform Based SOA" Journal of Convergence Information Technology JCIT, Vol. 7, pp. 223-224, 2012.
- [5] Zhanmin Xu, Haoran Zhang, "A Solution for Integration of Enterprise Legacy Systems Based on SOA", the Seventh Wuhan International E-Business Conference, pp.777, 2008.
- [6] Chatla S., Kadam S., "Complex networks and SOA: Mathematical modelling of granularity based web service compositions", Sadhana, Springer India, vol.36, no.4, pp. 441-461, 2011.
- [7] A. M. Riad, Q. F. Hassan, "Service-Oriented Architecture - A New Alternative to Traditional Integration Methods in B2B Applications", Journal of Convergence Information Technology, Advanced Institute of Convergence Information Technology, vol.3, no.1, pp.31-32, 2008.
- [8] JIAN Jie, WU Jian-jun, "Design for general interface of community information foundational database based on SOA", Computer Engineering and Design, Computer Engineering and Design, vol.31, no.16, pp.3612, 2010.
- [9] Erl T, "Services Oriented Architecture : Concepts, Technology and Design", Indiana, Prentice Hall PTR, 2005.
- [10] Sciore et al., "Using Semantic Values to Facilitate Interoperability among Heterogenous Information System", ACM Trans. Database System, pp. 254-290, 1994.
- [11] Lenzerini M., "Data Integration : A theoretical perspective", In Symposium on Principles of Database System, 2002.
- [12] Chawathe et al., "The TSIMMIS project : Integration of Heterogenous Information Resources", In Proceedings of 10th Meeting of the Information Processing Society of Japan, 1994.
- [13] Haase P., "Heterogenous Data Replication", In BTW Studierenden-Program, 2001.
- [14] Rajpathak Dnyanesh¹, Chougule Rahul, "A generic ontology development framework for data integration and decision support in a distributed environment", International Journal of Computer Integrated Manufacturing, Taylor and Francis Ltd., vol.24, no.2, pp.154-170, 2011.
- [15] HuDerek Hao, ZhengVincent Wenchen, "Cross-domain activity recognition via transfer learning", Pervasive and Mobile Computing, Elsevier, vol. 7, no.3, pp. 344-358, 2011.

Sebagai Pemakalah Seminar Nasional



Design Model Integration and Synchronization Between Surveillance Units to Support Data Warehouse Epidemiology

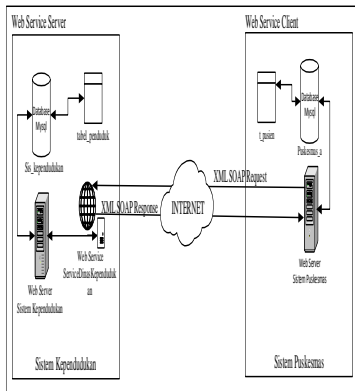
Fikri Budiman^a, Slamet Sudaryanto N^a, Sudaryanto^a

^aFaculty of Computer Science, Dian Nuswantoro University, City of Semarang, Central Java, Indonesia

Article history

Received XXXX
Received in revised form XXXX
Accepted XXXX

Graphical abstract



Abstract

Data monitoring outbreaks of disease in a region and specific populations (epidemiology) requires the support of an integrated data from all related surveillance unit. The integration of data between surveillance units (health centers, polyclinics, hospitals) must be design and good properly managed so as to enable health leaders and analysts to acquire, integrate, analyze and monitor the data (cases of the disease) from different data sources. The source data comes from a heterogeneous system, in this case the data source is a distribution surveillance units. To facilitate the management of surveillance data will be in the design of the data center in the model epidemiological data warehouse so as to form an integrated surveillance system (ISS). The problem faced is related to interoperability, ie the ability to integrate and synchronize data from different systems platforms (heterogeneous). Thus we need a methodology of integrating data in XML model into the epidemiological data warehouse. Since XML becomes a standard for data exchange over internet, especially in the B2B and B2C communication is needed integration system with the model XML data into the data warehouse. In this study we describe the design of a model of integration and synchronization between surveillance unit in the method SOA-based web services architecture. Metode yang digunakan adalah XML Web Services, sebuah metode yang dapat mengintegrasikan aplikasi dan pertukaran data dalam format XML (Extensible Markup Language). Exchange data in XML format using the technology SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language) and using the library NuSOAP.

Keyword : Epidemiology, Data Warehouse, Integration, Synchronization, XML,SOA, WebServices

© 2016 Penerbit UTM Press. All rights reserved.

1.0 INTRODUCTION

Design data warehouse epidemiology is a strategy that has a very important contribution in improving the quality and quantity monitoring health data. Epidemiological surveillance is a series of systematic sustainable process in the collection, analysis and interpretation of health data. Disease prevention and eradication programs are most effective if they are supported by epidemiological data warehouse, because they one of the functions of a data warehouse is the availability of epidemiological data sourced from all units in an integrated surveillance and up to date With System Data Warehouse then the quality of epidemiological surveillance will always be preserved. Epidemiological data warehousing system is a set of technologies and tools that enable data sources epidemiological surveillance

unit (health centers, polyclinics, hospitals) are scattered and heterogeneous can be perform interoperability (communication data), especially in terms of integration and data synchronization. The most important part of this system is a data warehouse data marts, a data mart is a historical collection of data sources (local schema) that is oriented to the needs of particular epidemiological analyzes. Data warehouse system design process consists of several stages such as analysis of data from heterogeneous data sources surveillance unit, the design of data warehouse model, the definition of data transformation and integration process, synchronization design and implementation of equipment needed by the user to access and represent data warehouse. Web services are the most popular method of SOA now [1]. Has a series of criteria and protocols that guarantee the functions to be

performed, such as WebServices Description Language (WSDL), Web services are platform independent and are suitable for heterogeneous environments.

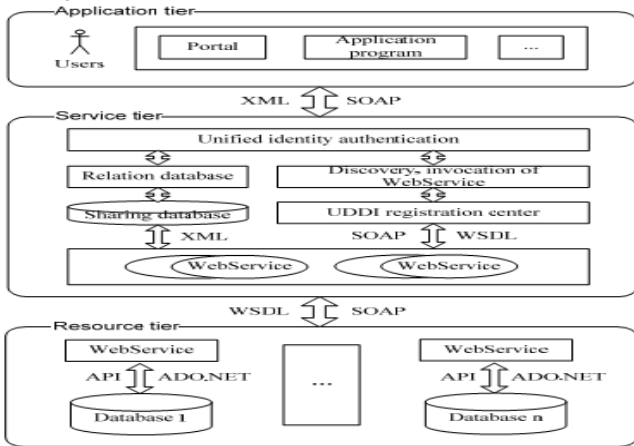


Figure 1 SOA-based web services architecture

SOA is a leading integration method and the heterogeneous system environment architecture framework [3]. The basic idea of SOA is services cored and integrate information resources on service standards, enabling reconfiguration and reuse of information resources [5]. SOA is respite services that can be combined with architecture services [6,7]. Communication services with a simple interface definition and precision that does not involve programming interface and communication models. There are three parts SOA architecture based on web services: services provider, registry services, consumer services [8]. Consumer service providers and each refers to the system of external and internal corporate systems. As a solution can be divided into three tier architecture, the application tier, services tier and resource tier. Resource tier contains all sorts of databases of the MIS wich need to be integrated. Services tier is the core of the solution, Including web services encapsulation, composition, registration, discovery, invocation and the data mapping module. Application tier consists of the application client of MIS ind information portal. XML is an emerging Internet applications oriented markup language for documents contains both content and structure. Xml is extensible, platform independent and easy to transfer in the network, so it is very suitable for describing information and the data synchronaization and database integration.

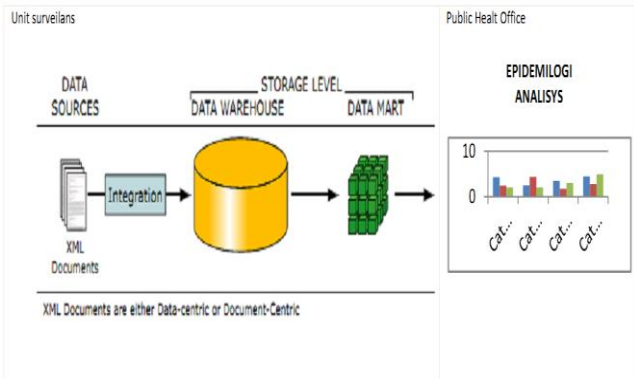


Figure 2 Integration of XML document within a data warehouse epidemiology

UDDI, Simple Object Access Protocol (SOAP) and so on [2].

XML web svcsies is a software system designed to support interoperability in machine to machine interactions in a network [9]. Interaction is done through a specific protocol. Thus the ability of web services can be increase the ability of the web to communicate and share information and data with the pattern of program-to-program. With the design of XML web services will can be integrate the system, programming language support, database and operating system platform that is different from the protocol http (Hypertext Transfer Protocol). Thus dbms application and heterogeneous between surveillance unit can communicate with the system of epidemiological data warehouse. The purpose of this study was to analyze, design, system integration and synchronization between surveillance unit with the health service system in a prototype model epidemiological data warehouse.

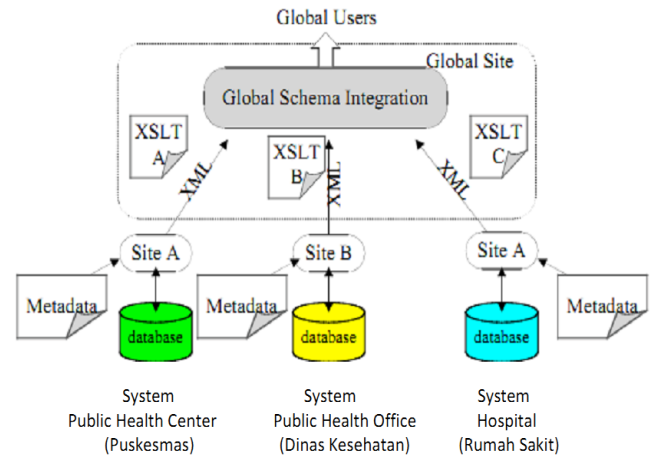


Figure 3 The heterogenous database integration architecture

1.1. Heterogenous Data Integration Problem

Integration is the process of combining different data in different data sources, and provide users with a unified view of the data [11]. Basically this process through the development of an integrated system. Lazerini [11] characteristics of the system as a data integration architecture based on a global scheme and a set of data sources. The data source contains data riel, while global scheme provides a reconciliation, integration, and a virtual view of the basic sources of data. With global virtual global view, users can be gain access system integrated uniform data from different data sources. Two basic approaches are used between the model of global schema and data sources (local scheme). The first Global-as-View (GAV) where the global schema as the data source [12]. The second is the Local-as-View (LAV) in which each data source is defined as a view over the global schema. Some of the problems related to the integration of heterogeneity of data between data sources as a local schema with a global schema is schema mapping, the data cleansing, transformation of data, the data reconciliation problem.

1.2. Heterogenous Data Synchronization Problem

Sinkronisasi heterogenous data directly related to the process of replication of data between heterogeneous databases. Data replication is the process of generating, producing, and maintaining a set of data in one or more locations [13]. Data replication between databases is very important to the availability of the data up-to-date whenever and wherever the data is needed.

In terms of communication and the timeliness of the data replication process can be synchronous or asynchronous. The process of data synchronization is the process of transforming the current data status of the application system to other heterogeneous systems and update the data to maintain data consistency. Synchronization of data as "a technical method that can Achieve consistency of data from databases in heterogeneous multi-platform". Synchronization is also a specially needs data integration to provide real-time consistency between local and global data sources through the update mechanism. Thus, data synchronization means that if there are changes to the data on the local data source will be notified, or distributed to a datasource that corresponds to the domain administrator information. Therefore, an integrated environmental changes will automatically perform inter-system update. Synchronization can be classified into uni-directional data synchronization and bi-directional data synchronization. Some of the problems related to the data synchronization is inter-process communication on the use of data and resources together.

2.0 EXPERIMENTAL

2.1. Architecture Model Integration Between Surveillance Units
 In the draft for a model of integration between applications to be built with webs services is the integration of epidemiological data from surveillance units health centers and hospitals. Where the surveillance unit that uses dbms and different applications (My Sql and PostgreSQL). Access will be built in advance of the application communication occurs in two directions, which through web services each surveillance unit will take the local epidemiological data from the database was changed into the form of a document in a format specific parameters (SOAP Services). Then the web service to send and request access between applications will continue to health office (dinkes) epidemiological database as epidemiological data warehouse.

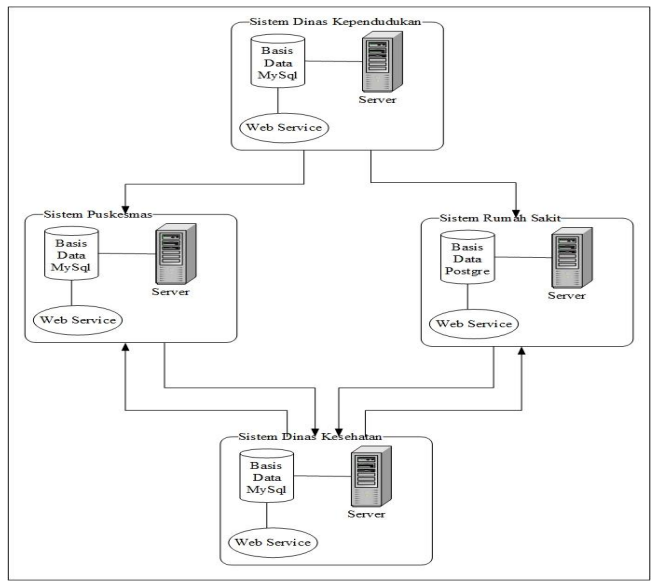


Figure 4 Model design data integration between units surveillance in a data warehouse epidemiological

2.2. Design Model Integration Web Services Between Units Surveillance.

a). The integration between the systems department of population with health center system. The system of official of population

information will provide a service to health centers and hospitals. Here is the architectural integration between the system of population with health center system.

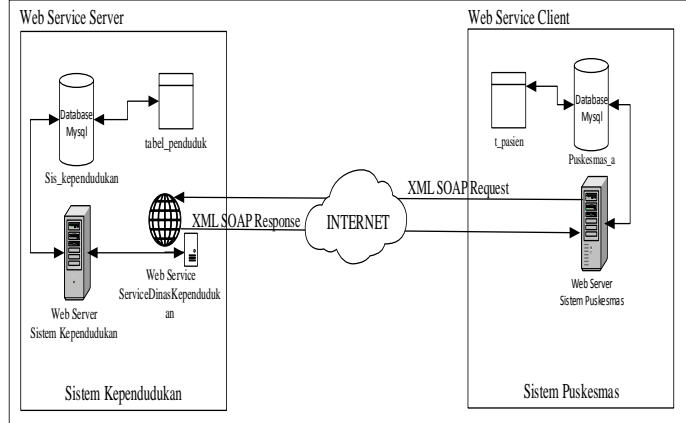


Figure 5 Architecture web services integration between the system of population with health center system

Population system also provides service to the hospital system that integrates both, the following is the architectural integration between the system of population with hospital systems.

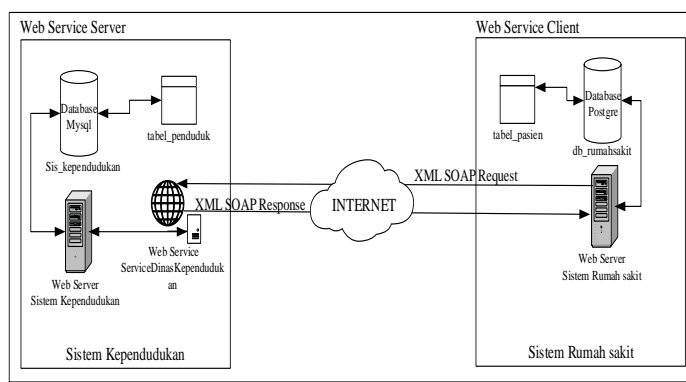


Figure 6 Architecture web service integration between systems of population with hospital systems

b). Integration between health center systems with system health office. Health center information system will provide service to the health department, so the health department can receive data from each recording of cases of the disease. The following is a system architecture integration.

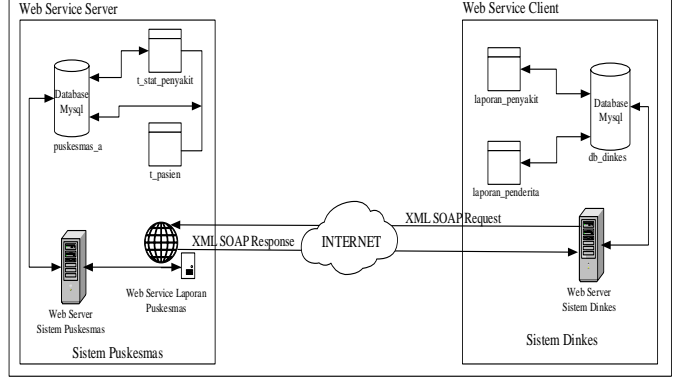


Figure 7 Integration of web services architecture health center system with system health office

c). The integration between the hospital system with system health office. The integration between the hospital information system with a system allowing health office to monitor health

office epidemiology ddata sourced from hospitals, the following is a systematic integration architecture.

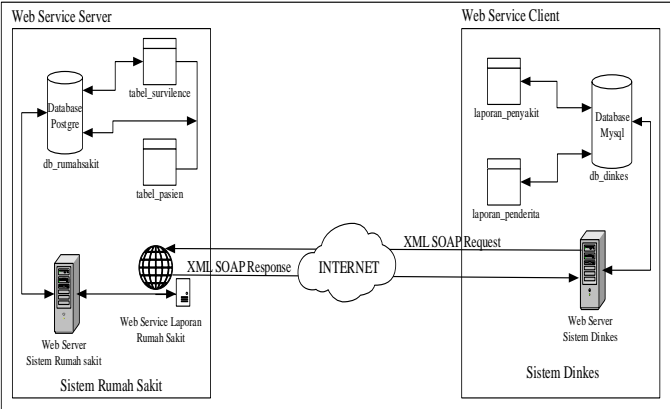


Figure 8 Integration architecture webservices hospital system with system health office

d). Integration between health center systems with system health office. Integration of health services system will provide service to the health centers system, so that epidemiological surveillance data each case the disease will be able to monitor the health office. The following is a the architecture of system integration between health office with health center system.

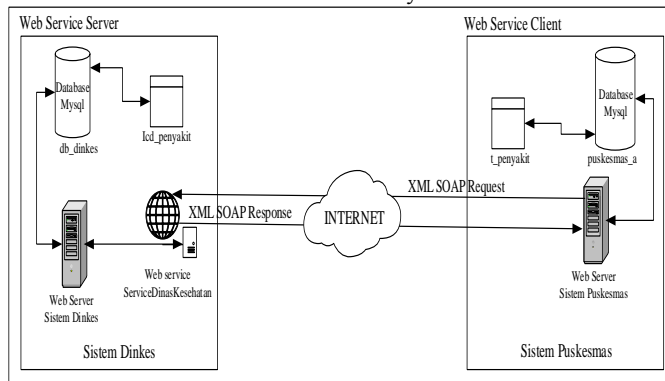


Figure 9 Integration of web services architecture system health office with system health centers

2.3. Web Services Design

After the design phase of web services integration model between several surveillance units, then the next stage is to perform the implementation of the XML programming on each web design services related to system integration. Atar surveilans related units. Some of web design services are as follows:

a). Web services system of reporting health centers to integrate with system health office.

Web Service Laporan Puskesmas

View the [WSDL](#) for the service. Click on an operation name to view it's details.

- [getLaporanKDPA](#)
- [getLaporanW2PA](#)
- [getLaporanLB1PA](#)
- [getPenderitaW1](#)
- [getPenderitaW2](#)
- [getPenderitaLB1](#)

Figure 10 Web services for health centers system

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:DataLaporanPuskesmas"
xmlns:soap="http://schemas.xmlsoap.org/wdl/soap/"
xmlns:wdl="http://schemas.xmlsoap.org/wdl/"
xmlns="http://schemas.xmlsoap.org/wdl/"
targetNamespace="urn:DataLaporanPuskesmas">
<types>
<xsd:schema targetNamespace="urn:DataLaporanPuskesmas">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wdl/" />
</xsd:schema>
</types>
<message name="getLaporanKDPARequest">
<part name="tanggal" type="xsd:string" /></message>
<message name="getLaporanKDPAResponse">
<part name="output" type="xsd:Array" /></message>
.....
<input><soap:body use="encoded" namespace="urn:DataPenderitaPenyakitLB1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></input>
<output><soap:body use="encoded"
namespace="urn:DataPenderitaPenyakitLB1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></output>
</operation>
</binding>
<service name="Web Service Laporan Puskesmas">
<port name="Web Service Laporan PuskesmasPort" binding="tns:Web Service
Laporan PuskesmasBinding">
<soap:address
location="http://localhost/ta/puskesmasA/ws/puskesmas_service.php"/>
</port>
</service>
</definitions>
.....
```

Figure 11 Example WSDL web services health centers ssystem

b). Web services for hospital reporting system integrated with health office system.

Web Service Laporan Rumah Sakit

View the [WSDL](#) for the service. Click on an operation name to view it's details.

- [getLaporanW1RS](#)
- [getLaporanW2RS](#)
- [getLaporanLB1RS](#)
- [getLaporanPenyakitTahunanRS](#)
- [LaporanPenderitaPenyakitW1](#)
- [LaporanPenderitaPenyakitW2](#)
- [LaporanPenderitaPenyakitLB1](#)

Figure 12 Web services for hospital system

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:DataLaporanRumahSakit"
xmlns:soap="http://schemas.xmlsoap.org/wdl/soap/"
xmlns:wdl="http://schemas.xmlsoap.org/wdl/"
xmlns="http://schemas.xmlsoap.org/wdl/"
targetNamespace="urn:DataLaporanRumahSakit">
<types>
<xsd:schema targetNamespace="urn:DataLaporanRumahSakit">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wdl/" />
</xsd:schema>
</types>
<message name="getLaporanW1RSRequest">
<part name="tanggal" type="xsd:string" /></message>
<message name="getLaporanW1RSResponse">
<part name="output" type="xsd:Array" /></message>
...
<message name="getLaporanPenyakitTahunanRSRequest">
<part name="Tahun" type="xsd:int" /></message>
<message name="getLaporanPenyakitTahunanRSResponse">
<part name="output" type="xsd:Array" /></message>
<message name="LaporanPenderitaPenyakitW1Request">
<part name="kodeIdpenyakit" type="xsd:string" />
<message name="LaporanPenderitaPenyakitW2Request">
<part name="kodeIdpenyakit" type="xsd:string" />
.....
</operation>
<operation name="getLaporanLB1RS">
<soap:operation soapAction="urn:LaporanLB1RumahSakit#getLaporanLB1RS"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanLB1RumahSakit"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></input>
<output><soap:body use="encoded" namespace="urn:LaporanLB1RumahSakit"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></output>
</operation>
<operation name="getLaporanPenyakitTahunanRS">
<soap:operation
soapAction="urn:LaporanPenyakitTahunanRumahSakit#getLaporanPenyakitTahunanRS" style="rpc"/>
.....
```

Figure 13 Example WSDL web services hospital system

2.4. Mapping Rule Schema

A data integration techniques, among heterogeneous scheme mainly deal with the problem of the transformation and exchange of data. Mapping rules are middle-ware that is independent and not influenced from various platforms. Mapping rules can also be applied schema in format and values of query results.

```
// Mapping value schema dari hasil query
$SQLcekRow=$this->db->query("SELECT * FROM laporan_penderita WHERE 'kode_rekam'='$kode_rekam'
AND 'kode_pasien'='$kode_pasien'
AND tanggal_kejadian='$tanggal_kejadian'
AND kode_instansi='$kodere'
AND kode_penyakit='$value[kodepenyakit]'");
$cekDataPasien=$SQLcekRow->num_rows();
if($cekDataPasien==$value['jmlpenyakit']){
continue;
}
else{
foreach ($arr['pasienrs'] as $key ) {
$SQLcekRowPasien=$this->db->query("SELECT * FROM laporan_penderita
WHERE 'kode_rekam'='$key[kode_rekam]'
AND 'kode_pasien'='$key[kode_pasien]'
AND tanggal_kejadian='$tanggal_kejadian'
AND kode_instansi='$kodere'
AND kode_penyakit='$value[kodepenyakit]'");
$cekDataPasienDB=$SQLcekRowPasien->num_rows();
if($cekDataPasienDB==1){
continue;
}
else{
//insert ke tabel laporan data pasien dari hasil query
$this->db->set('kode_rekam', $key['kode_rekam']);
$this->db->set('kode_pasien', $key['kode_pasien']);
$this->db->set('nama_pasien', $key['nama_pasien']);
$this->db->set('alamat_pasien', $key['alamat_pasien']);
$this->db->set('deasa_pasien', $key['kelurahan_pasien']);
$this->db->set('umur_pasien', $key['umur_pasien']);
$this->db->set('status_pasien', $key['statuspasien']);
$this->db->set('tanggal_kejadian', $tanggal_kejadian);
$this->db->set('kode_instansi', $kodere);
$this->db->set('kode_penyakit', $value['kodepenyakit']);
$this->db->insert('laporan_penderita');
}
}
}
.....
```

Figure 14 Example mapping value schema from query

2.5. Design Gav (Global as View) as a mediator schema synchronization.

Is a global schema as mediator that serve to help synchronize data items in the form of unit surveilas local schema LAV (Local as View). The mediator scheme is temporary shelter data reception with local schema defines techniques (reformulated) to match the structure of the global scheme. Scheme mediator also helps the process of reformulating the query with the data request through the local schema global scheme.

```
class="break"></span>Sinkronisasi Data Center</h2>
<div class="box-look">
...
function sinkronisasi($tgl,$bulan,$tahun){
$tanggal=$tahun."-".$bulan."-".$tgl;
$url= $SERVER['DOCUMENT_ROOT']."/cas/dinkee/application/libraries/nusoap/lib/nusoap.php";
require_once($url);
$client=new nusoap_client("http://localhost/cas/puskesmas/ws/puskesmas_service.php?wsdl");
$result=$client->call('getLaporanKODPA',array('tanggal'=>$tanggal));
if($is_array($result)){
$jmlLappuskesmas=count($result);
...
}
/* -- proses sinkronisasi apakah jml.data penyakit di dbcenter sama dengan jml penyakit dari RS
if($jmlRec==$value['jmlpenyakit']){
// echo " | no action ".$.$.", ";
continue;
}
}
/* cek apakah jml record yang ada di dbcenter masih 0, maka akan insert
elseif($jmlRec==0){
// echo "insert ke ".$.$.", ";
$this->db->set('kode_id', $value['kodepenyakit']);
$this->db->set('jumlah_penyakit', $value['jmlpenyakit']);
$this->db->set('tanggal_laporan', $row['tgl']);
$this->db->set('kode_instansi', $kodere);
$this->db->insert('laporan_penyakit');
foreach ($arr['pasienrs'] as $key ) {
$this->db->set('kode_rekam', $key['kode_rekam']);
$this->db->set('kode_pasien', $key['kode_pasien']);
$this->db->set('nama_pasien', $key['nama_pasien']);
$this->db->set('alamat_pasien', $key['alamat_pasien']);
$this->db->set('umur_pasien', $key['umur_pasien']);
$this->db->set('status_pasien', $key['statuspasien']);
$this->db->set('tanggal_kejadian', $tanggal_kejadian);
$this->db->set('kode_instansi', $kodere);
$this->db->set('kode_penyakit', $value['kodepenyakit']);
$this->db->insert('laporan_penderita');
}
}
}
/* cek jumlah laporan yang sudah masuk ke dbcenter
$kon=new PDO("mysql:host=localhost;dbname=db_dinkee','root','timur2000");
$pr=$kon->prepare("SELECT COUNT(*) AS jmlpenyakit FROM 'laporan_penyakit'
WHERE 'tanggal_laporan'=:tanggal AND 'kode_instansi':kodeinstansi");
$pr->bindParam(":tanggal",$tanggal);
$pr->bindParam(":kodeinstansi",$kodeinstansi);
$pr->execute();
$cek=$pr->fetchAll();
foreach ($cek as $key) {
$jmlr=$key['jmlpenyakit'];
}
}
.....
```

Figure 15 Example GAV as mediator schema synchronization between local and global schemes

3.0 RESULTS AND DISCUSSION

Prototype integration of the data center that was developed at health office surveillance is an integration of two units, namely health centers and hospitals. By designing XML Web Services data from two units of surveillance data can be unified into a single integrated system (data warehouse model) although derived from the system and dbms different applications (multi-platform). The figure below shows the number of data reports information derived from epidemiological surveillance units (health centers and hospitals) both daily (w1), weekly (w2) or monthly (LB1).

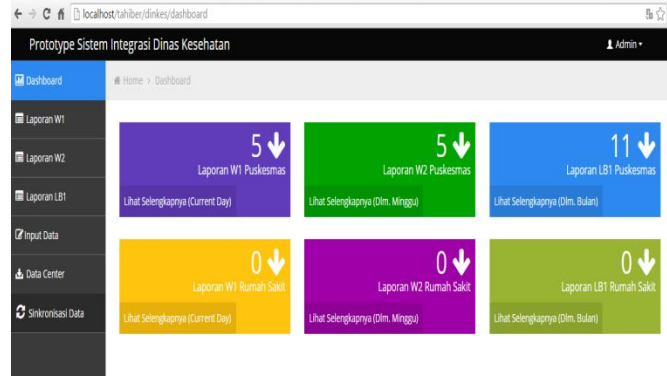


Figure 16 Monitoring and analysis of the results of the integration of data center epidemiological data on health office

For monitoring epidemiological data transmission from the surveillance unit shipment data can be synchronized from each surveillance unit each day continuously, so that it can automatically in the monitoring. With daily monitoring of this system will can be ensure the surveillance unit anywhere that does not perform epidemiological data transmission so that it can be done a certain action, it is very important to monitor the outbreak, particularly for infectious disease outbreaks.

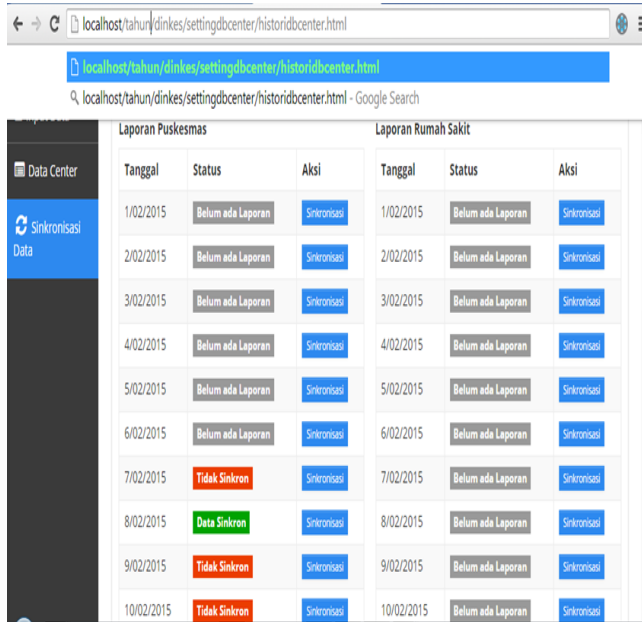


Figure 17 Results of the data synchronization process

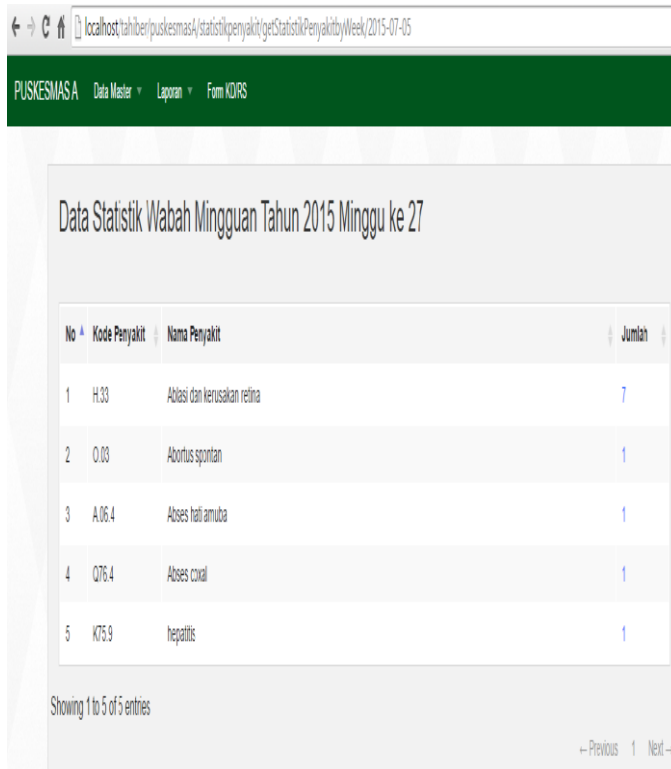


Figure 18 Monitoring statistics epidemiological surveillance in outbreak of certain health centers

4.0 CONCLUSION

In this research has been done web design development servives with XML and has conducted testing of prototype data center where the health agency epidemiological data sources epidemiological derived from epidemiological data reporting health centers and hospitals with a daily format (W1), weekly (W2) and monthly (LB1). From the results of prototype testing of the epidemiological data center, web design service successfully integrated reporting system of units surveilas although epidemiological data sources come from different applications and dbms (MySQL and PostgreSQL) as well as different data structures. This shows the communication protocol middleware web service that is able to exchange epidemiological data as messege by utilizing HTTP protocol via a computer network between web-based and database apikasi between surveillane unit with database data center health office epidemiology. Integration between surveillane unit with web technology servives in computer networks also showed communication between the provider and the services requester servives can be used for monitoring real-time data delivery, so that it can be designed monitoring and synchronization of application data delivery effectively. Thus every event an outbreak of a particular region can be monitored continuously in order to program the eradication and control of diseases.

Acknowledgement. We would like to thank Kemenristek Higher Education, which has support in funding this research (Hiber). Also To the Faculty of Computer Science as well as the leadership of the University of Dian Nuswantoro Semarang, Indonesia

References

- Al-Otaibi Noura Meshaan1, Noaman Amin Yousef, “Biological data integration using SOA”, Proceedings of World Academy of Science, Engineering and Technology, WASET, vol.73, pp.920-925, 2011.
- Chung C. Chang, Kou-Chan Hsiao, “A SOA-Based e-Learning System for Teaching Fundamental Information Management Courses”, Journal of Convergence Information Technology, Advanced Institute of Convergence Information Technology, vol.6, no.4, pp.298-2011.
- Karande, Aarti M., “Working of web services using BPEL workflow in SOA”, 2nd International Conference on Advances in Computing, Communication and Control, pp.143-149, 2011.
- Gouwei W, Guangming X., “research On Data Synchronization and Integration Platform Based SOA” Journal of Convergence Information Technology JCIT, Vol. 7, pp. 223-224, 2012.
- Zhanmin Xu, Haoran Zhang, “A Solution for Integration of Enterprise Legacy Systems Based on SOA”, the Seventh Wuhan International E-Business Conference, pp.777, 2008.
- Chatla S., Kadam S., “Complex networks and SOA: Mathematical modelling of granularity based web service compositions”, Sadhana, Springer India, vol.36, no.4, pp. 441-461, 2011.
- A. M. Riad, Q. F. Hassan, “Service-Oriented Architecture - A New Alternative to Traditional Integration Methods in B2B Applications”, Journal of Convergence Information Technology, Advanced Institute of Convergence Information Technology, vol.3, no.1, pp.31-32, 2008.
- JIAN Jie, WU Jian-jun, “Design for general interface of community information foundational database based on SOA”, Computer Engineering and Design, Computer Engineering and Design, vol.31, no.16, pp.3612, 2010.
- Erl T, “Services Oriented Architecture : Concepts, Technology and Design”, Indiana, Prentice Hall PTR, 2005.
- Sciore et al., “Using Semantic Values to Facilitate Interoperability among Heterogenous Information System”, ACM Trans. Database System, pp. 254-290, 1994.
- Lenzerini M., “Data Integration : A theoretical perspective”, In Symposium on Principles of Database System, 2002.
- Chawathe et al., “The TSIMMIS project : Integration of Heterogenous Information Resources”, In Proceedings of 10th Meeting of the Information Processing Society of Japan, 1994.
- Haase P., “Heterogenous Data Replication”, In BTW Studierend-Program, 2001.
- Rajpathak Dnyanesh1, Chougule Rahul, “A generic ontology development framework for data integration and decision support in a distributed environment”, International Journal of Computer Integrated Manufacturing, Taylor and Francis Ltd., vol.24, no.2, pp.154-170, 2011.
- HuDerek Hao, ZhengVincent Wenchen, “Cross-domain activity recognition via transfer learning”, Pervasive and Mobile Computing, Elsevier, vol. 7, no.3, pp. 344-358, 2011.

Lampiran 4. : Pengajuan HKI Hak Cipta

1


Lampiran I
Peraturan Menteri Kehakiman R.I.
Nomor : M.01-HC.03.01 Tahun 1987

Kepada Yth. :
 Direktur Jenderal HKI
 melalui Direktur Hak Cipta,
 Desain Industri, Desain Tata Letak,
 Sirkuit Terpadu dan Rahasia Dagang
 di
 Jakarta

PERMOHONAN PENDAFTARAN CIPTAAN

I.	Pencipta :	
	1. Nama	: Fikri Budiman, M.Kom ^(a) Slamet Sudaryanto N, ST, M.Kom ^(b) Muslih, M.Kom ^(c)
	2. Kewarganegaraan	: Indonesia ^(a,b,c)
	3. Alamat	: Perum Bukit Diponegoro B.205, Kecamatan Tembalang, Kota Semarang ^(a)
II.	Pemegang Hak Cipta :	
	1. Nama	: Fakultas Ilmu Komputer Universitas Dian Nuswantoro, Kota Semarang
	2. Kewarganegaraan	: Indonesia
	3. Alamat	: Jl. Nakula I No.5-11 Semarang 50131
III.	Kuasa :	
	1. Nama	: -
	2. Kewarganegaraan	: -
	3. Alamat	: -
IV.	Jenis dan judul ciptaan yang dimohonkan	: Perangkat Lunak / "Prototipe Model Data Center Antar Unit Surveilans"
V.	Tanggal dan tempat di- umumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indo- nesia.	: Rabu 22 Agustus 2014 di Hotel Patra Jasa Semarang pada Kegiatan Pemaparan Hasil Seleksi Proposal Penelitian Desentralisasi Kopertis VI Kota Semarang Jawa Tengah
VI.	Uraian ciptaan	: Prototipe Model Data Center Antar Unit Surveilans ini merupakan model sinkronisasi dari database dan aplikasi yang heterogen (berbeda platform) pada masing - masing unit surveilans (puskesmas & Rumah Sakit) sebagai sumber data center. Dengan menggunakan SOA Web Services sumber data yang berbeda platform tersebut dapat di strukturisasi dan integrasi untuk membentuk pusat data (data center) epidemiologi, sehingga dapat member anfaat untuk melakukan analisa dan monitoring kasus dalam pengendalian penyebaran penyakit. Hal ini dikarenakan data center dapat digunakan untuk menerima kiriman data secara on line dari masing-masing unit surveilans. Aplikasi ini berjalan pada lingkungan web base (berbasis web). Merupakan implementasi hasil penelitian hibah bersaing dengan No. Kontrak : 023/A.35-02/UDN.09/V/2016. Semarang, 10 Juli 2016

Dekan
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro
Dr. Abdul Syukur
NPP. 0686.11.1992.017





REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Menteri Hukum dan Hak Asasi Manusia Republik Indonesia, berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta yaitu Undang-Undang tentang perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra (tidak melindungi hak kekayaan intelektual lainnya), dengan ini menerangkan bahwa hal-hal tersebut di bawah ini telah tercatat dalam Daftar Umum Ciptaan:

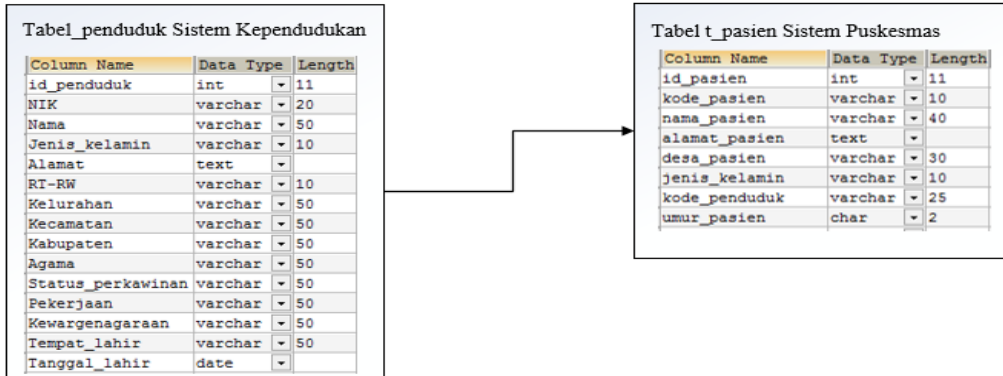
- I. Nomor dan tanggal permohonan : EC00201600754, 9 Agustus 2016
- II. Pencipta
- Nama : **Fikri Budiman, M.Kom**
Alamat : Perum Bukit Diponegoro B.205, rt.02 rw.08 Kelurahan Tembalang, kecamatan tembalang , Semarang, JAWA TENGAH, 50275
- Kewarganegaraan : Indonesia
- Nama : **Slamet Sudaryanto Nurhendratno,ST,M.Kom**
Alamat : Klipang Pesona Asri III Blok B-No.2, Klipang, Sendangmulyo, Tembalang , Semarang, JAWA TENGAH, 50272
- Kewarganegaraan : Indonesia
- Nama : **Muslih,M.Kom**
Alamat : Patemon Rt.01 Rw.03 Kel.Patemon Kec.GunungPati, Semarang, JAWA TENGAH, 50228
- Kewarganegaraan : Indonesia
- III. Pemegang Hak Cipta
- Nama : **Fakultas Ilmu Komputer Universitas Dian Nuswantoro**
Alamat : jalan Imam Bonjol 207, Semarang, JAWA TENGAH, 50131
- Kewarganegaraan : Indonesia
- IV. Jenis Ciptaan : Program Komputer
- V. Judul Ciptaan : **Prototipe Model Data Center Antar Unit Surveilans**
- VI. Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 22 Agustus 2014, di Semarang Jawa Tengah
- VII. Jangka waktu perlindungan : Berlaku selama hidup Pencipta dan terus berlangsung hingga 70 (tujuh puluh) tahun setelah Pencipta meninggal dunia.
- VIII. Nomor pencatatan : 00359

Pencatatan Ciptaan atau produk Hak Terkait dalam Daftar Umum Ciptaan bukan merupakan pengesahan atas isi, arti, maksud, atau bentuk dari Ciptaan atau produk Hak Terkait yang dicatat. Menteri tidak bertanggung jawab atas isi, arti, maksud, atau bentuk dari Ciptaan atau produk Hak Terkait yang terdaftar. (Pasal 72 dan Penjelasan Pasal 72 Undang-undang Nomor 28 Tahun 2014 Tentang Hak Cipta)

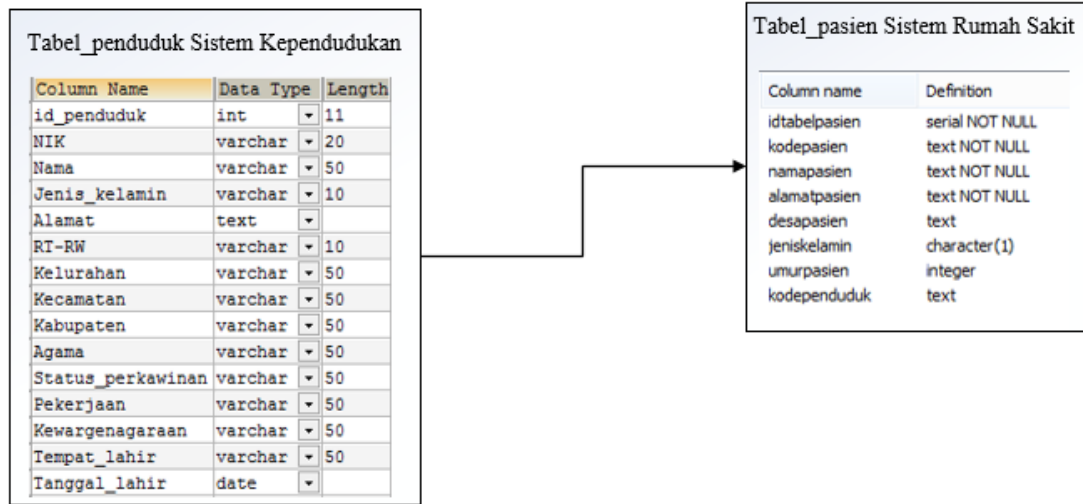
a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
REPUBLIK INDONESIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b.
DIREKTUR HAK CIPTA DAN DESAIN INDUSTRI

Dr. Dra. Erni Widhyastari, Apt., M.Si.
NIP. 196003181991032001

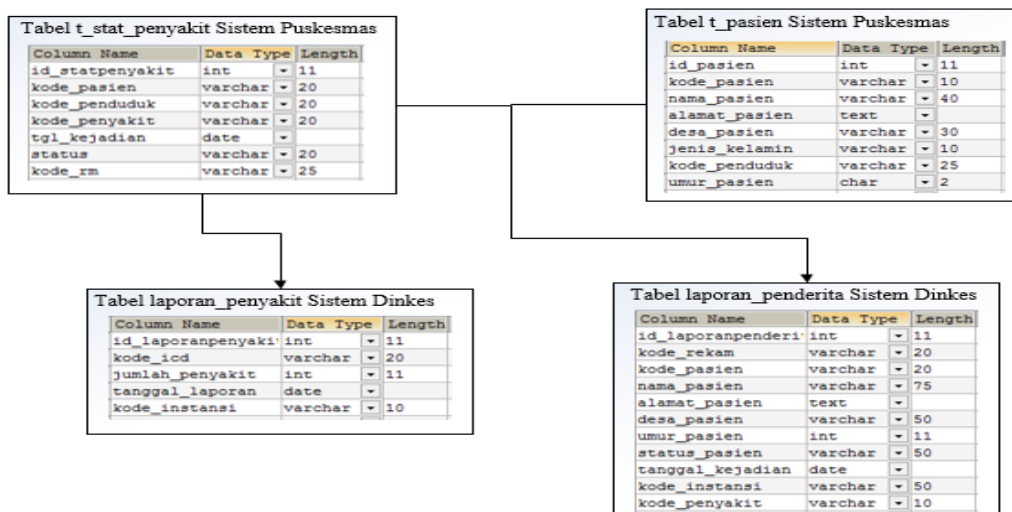
Lampiran 5. : Produk Penelitian



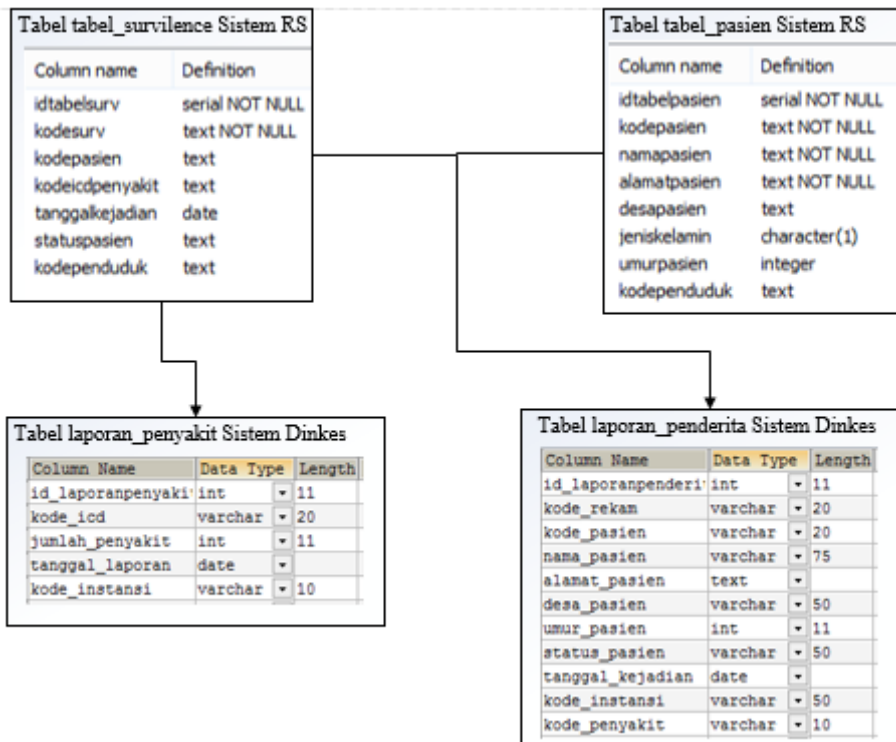
Gambar 3: Perbedaan nama dan struktur atribut tabel tabel_penduduk sistem Kependudukan dengan t_pasien sistem Puskesmas



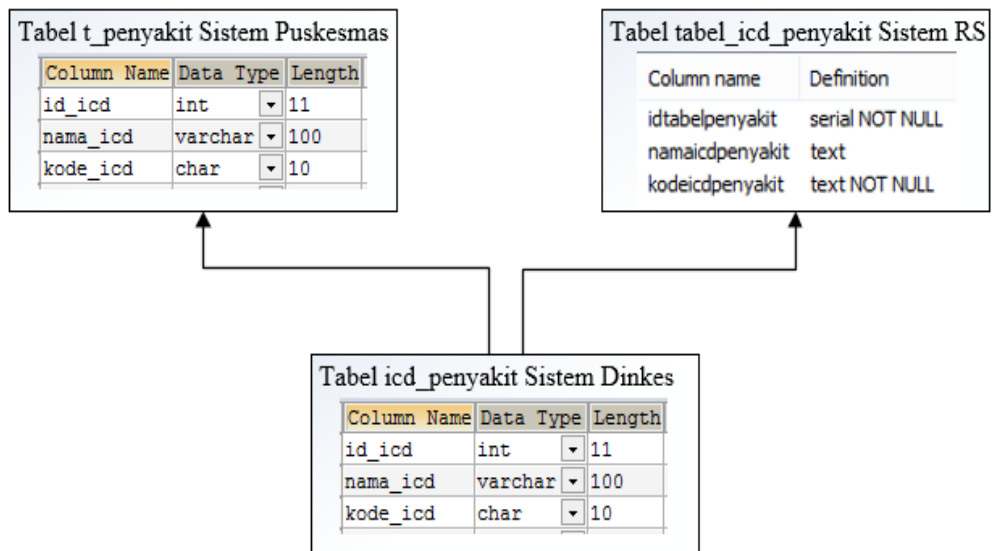
Gambar 4: Perbedaan nama dan struktur atribut tabel_penduduk sistem Kependudukan dengan tabel_pasien sistem Rumah sakit



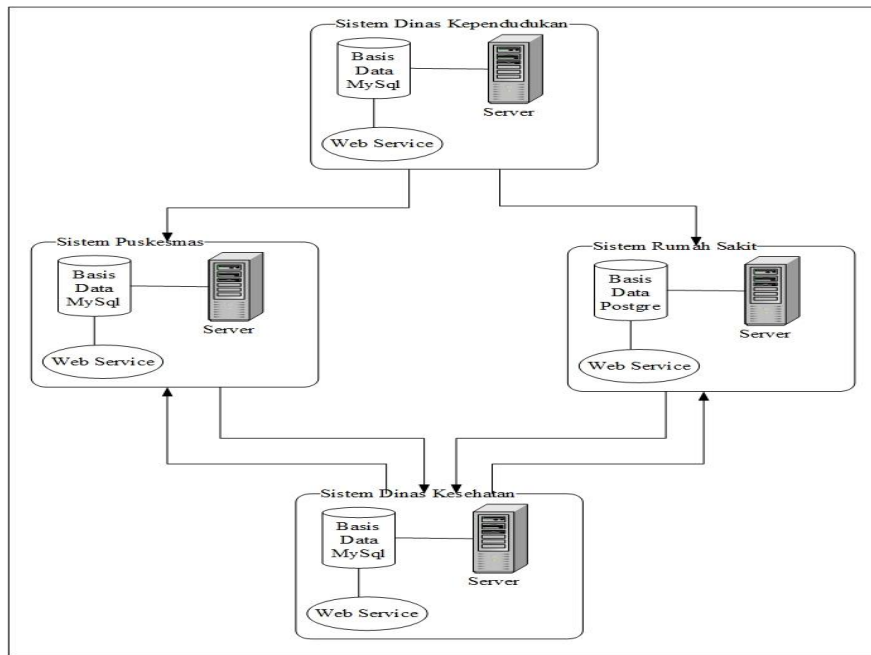
Gambar 5: Perbedaan arsitektur, penamaan tabel, dan struktur atribut antara sistem Puskesmas dengan sistem Dinkes



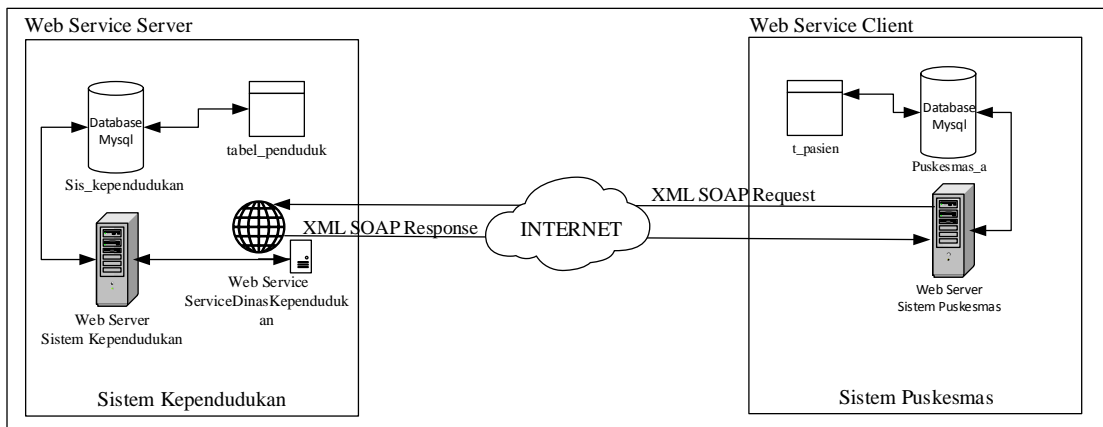
Gambar 6: Perbedaan arsitektur, penamaan tabel, dan struktur atribut antara sistem Rumah sakit dengan sistem Dinkes



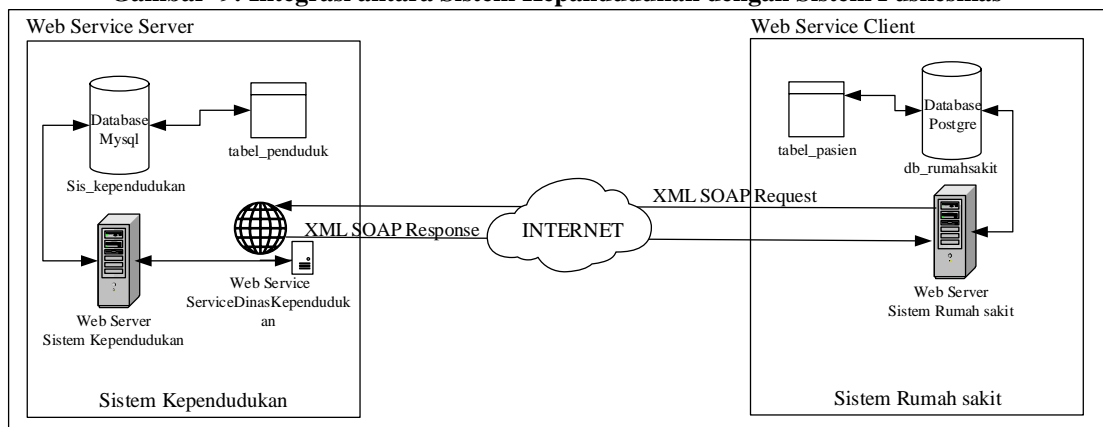
Gambar 7: Perbedaan nama dan struktur atribut tabel dari tiga tabel yang menyimpan ICD penyakit



Gambar 8: Desain Interaksi Web Service antar Sistem



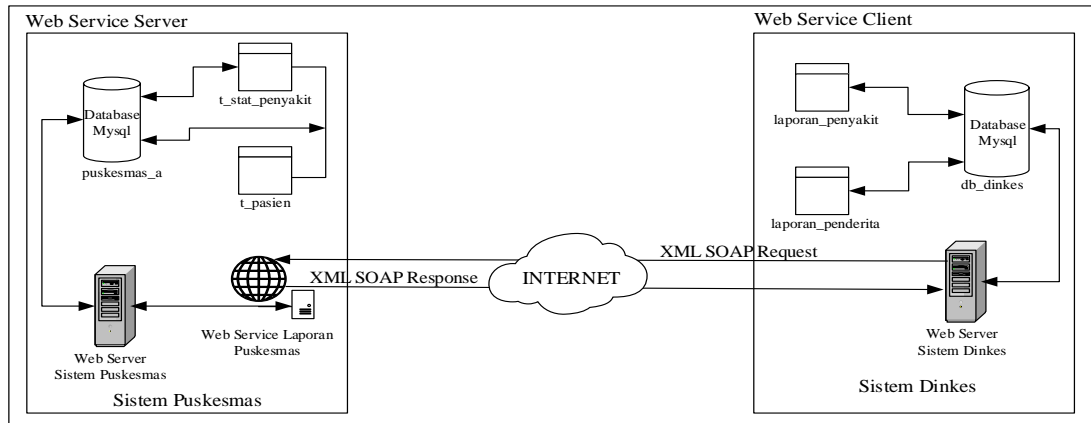
Gambar 9: Integrasi antara Sistem Kependudukan dengan Sistem Puskesmas



Gambar 10: Integrasi antara Sistem Kependudukan dengan Sistem Puskesmas

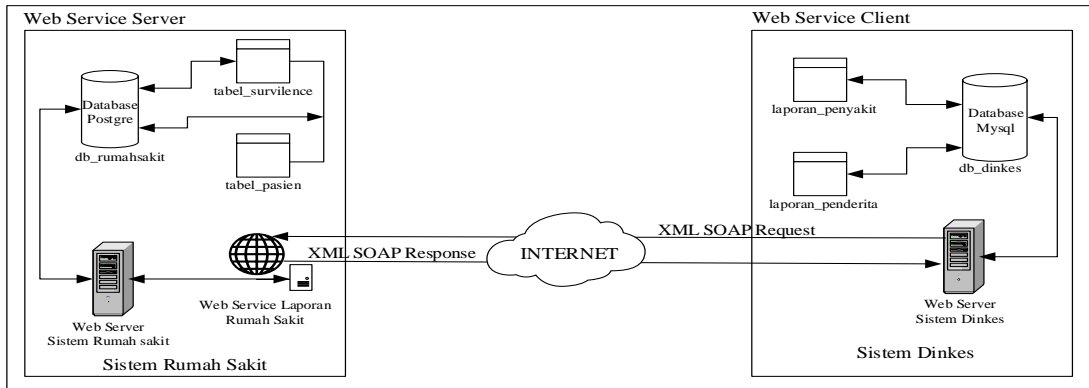
Tabel 1. Perancangan web service sistem Dinas Kesehatan

Method Web Service	Parameter	Url WSDL
getDataPenduduk	-	localhost/ta/disduk/ws/kependudukan.php?wsdl



Gambar 11: Arsitektur integrasi Sistem Puskesmas dengan Sistem Dinkes
Tabel 2. Perancangan web service sistem Puskesmas

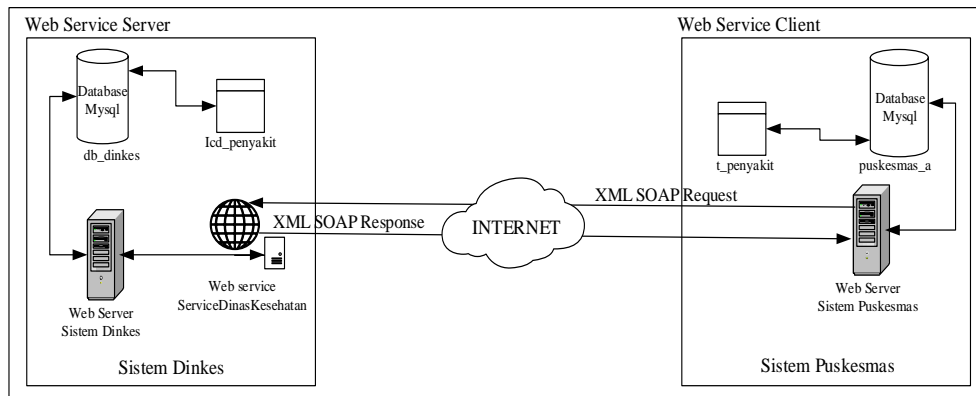
Method Web Service	Parameter	Url WSDL
getLaporanKDPA	Tanggal (yyyy-mm-dd)	http://localhost/tahun/puskesmasA/ws/puskesmas_service.php?wsdl
getLaporanW2PA	Tanggal (yyyy-mm-dd)	
getLaporanLB1PA	TahunBulan (yyyy-mm)	
getPenderitaW1	Kodeicd dan tgl (yyyy-mm-dd)	
getPenderitaW2	Kodeicd dan tgl (yyyy-mm-dd)	
getPenderitaLB1	Kodeicd, bulan, tahun	



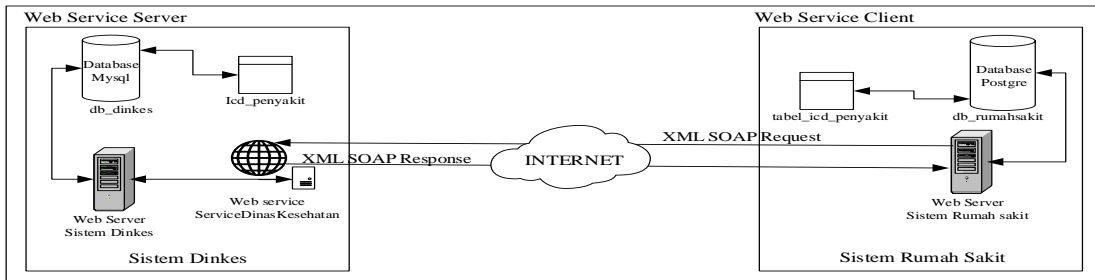
Gambar 12: Arsitektur integrasi sistem Rumah sakit dengan sistem Dinkes

Tabel 3. Perancangan web service sistem Rumah sakit

Method Web Service	Parameter	Url WSDL
getLaporanW1RS	Tanggal (yyyy-mm-dd)	http://localhost/tahun/rs/ws/laporan_rs.php?wsdl
getLaporanW2RS	Tanggal (yyyy-mm-dd)	
getLaporanLB1RS	TahunBulan (yyyy-mm)	
LaporanPenderitaPenyakitW1	Kodeicpenyakit, Tanggal (yyyy-mm-dd)	
LaporanPenderitaPenyakitW2	Kodeicpenyakit, Tanggal (yyyy-mm-dd)	
LaporanPenderitaPenyakitLB1	Kodeicpenyakit, tahun, bulan	



Gambar 13: Arsitektur Integrasi Sistem Dinas Kesehatan dengan Sistem Puskesmas



Gambar 14: Arsitektur Integrasi Sistem Dinas Kesehatan dengan Sistem Rumah sakit

Tabel 4. Perancangan web service sistem Dinas Kesehatan

Method Web Service	Parameter	Url WSDL
getDataICD	-	http://localhost/tahun/dinkes/ws/dinkes_service.php?wsdl

Gambar 15: Diagram use case sistem integrasi Dinkes



Tabel 5. Use case naratif proses melakukan login

Nama Use Case	Login
Aktor	Pegawai Dinkes
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk login ke sistem untuk menggunakan sistem.
Pre-condition	Pengguna belum bisa menggunakan sistem karena belum login.
Post-condition	Pengguna bisa menggunakan sistem dan bisa menggunakan menu-menu yang ada dalam sistem.
Aksi Aktor	Reaksi Sistem
Alur dasar (Basic Flow)	
Mengisi form username dan password pada halaman login	
Menekan tombol login	Mengecek validasi apakah data valid username dan password
	Jika data valid maka akan masuk ke halaman dashboard

Tabel 6. Use case naratif mendapatkan laporan W1 Puskesmas atau Rumah sakit

Nama Use Case	Mendapatkan laporan W1 Puskesmas atau Rumah Sakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk melihat data laporan W1 dari Puskesmas atau Rumah sakit	
Pre-condition	Pengguna belum mengetahui laporan W1 dari Puskesmas atau Rumah sakit dan Pengguna sudah login sistem.	
Post-condition	Pengguna bisa mengetahui data laporan W1 dari Puskesmas atau Rumah sakit.	
Aksi Aktor		Reaksi Sistem
Alur dasar (Basic Flow)		
Pengguna menekan menu Laporan W1 Puskesmas atau Rumah sakit	Sistem menampilkan laporan W1 dari service yang diberikan Puskesmas atau Rumah sakit, jika belum ada data maka ada keterangan.	

Tabel 7. Use case naratif melihat data penderita sesuai dengan laporan W1 Puskesmas atau Rumah sakit

Nama Use Case	Melihat Data Penderita sesuai dengan Laporan W1 Puskesmas atau Rumah sakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk melihat data laporan penderita penyakit sesuai dengan laporan W1 dari Puskesmas atau Rumah sakit	
Pre-condition	Pengguna sudah melihat laporan W1 Puskesmas atau Rumah Sakit	
Post-condition	Pengguna bisa mengetahui data laporan penderita penyakit sesuai laporan W1 dari Puskesmas atau Rumah sakit.	
Aksi Aktor		Reaksi Sistem
Alur dasar (Basic Flow)		
Pengguna menekan tombol detail pada tabel Laporan W1 Puskesmas atau Rumah sakit	Sistem menampilkan laporan penderita penyakit sesuai dengan laporan W1 dari service yang diberikan Puskesmas atau Rumah sakit.	

Tabel 8. Use case naratif mendapatkan laporan W2 Puskesmas atau Rumah sakit

Nama Use Case	Mendapatkan laporan W2 Puskesmas atau Rumah Sakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk melihat data laporan W2 dari Puskesmas atau Rumah sakit	
Pre-condition	Pengguna belum mengetahui laporan W2 dari Puskesmas atau Rumah sakit dan Pengguna sudah login sistem.	
Post-condition	Pengguna bisa mengetahui data laporan W2 dari Puskesmas atau Rumah sakit.	
Aksi Aktor		Reaksi Sistem
Alur dasar (Basic Flow)		
Pengguna menekan menu Laporan W2 Puskesmas	Sistem menampilkan laporan W2 dari service yang	

atau Rumah sakit	diberikan Puskesmas atau Rumah sakit, jika belum ada data maka ada keterangan.
------------------	--

Tabel 9. *Use case* naratif melihat data penderita sesuai dengan Laporan W2 Puskesmas atau Rumah sakit

Nama Use Case	Melihat Data Penderita sesuai dengan Laporan W2 Puskesmas atau Rumah sakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk melihat data laporan penderita penyakit sesuai dengan laporan W2 dari Puskesmas atau Rumah sakit	
Pre-condition	Pengguna sudah melihat laporan W2 Puskesmas atau Rumah Sakit.	
Post-condition	Pengguna bisa mengetahui data laporan penderita penyakit sesuai laporan W2 dari Puskesmas atau Rumah sakit.	
Aksi Aktor	Reaksi Sistem	
Alur dasar (Basic Flow)		
Pengguna menekan tombol detail pada tabel Laporan W2 Puskesmas atau Rumah sakit	Sistem menampilkan laporan penderita penyakit sesuai dengan laporan W2 dari service yang diberikan Puskesmas atau Rumah sakit.	

Tabel 10. *Use case* naratif mendapatkan laporan LB1 Puskesmas atau Rumah sakit

Nama Use Case	Mendapatkan laporan LB1 Puskesmas atau Rumah Sakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk melihat data laporan LB1 dari Puskesmas atau Rumah sakit	
Pre-condition	Pengguna belum mengetahui laporan LB1 dari Puskesmas atau Rumah sakit dan Pengguna sudah login sistem.	
Post-condition	Pengguna bisa mengetahui data laporan LB1 dari Puskesmas atau Rumah sakit.	
Aksi Aktor	Reaksi Sistem	
Alur dasar (Basic Flow)		
Pengguna menekan menu Laporan LB1 Puskesmas atau Rumah sakit	Sistem menampilkan laporan LB1 dari service yang diberikan Puskesmas atau Rumah sakit, jika belum ada data maka ada keterangan.	

Tabel 11. *Use case* naratif melihat data penderita sesuai dengan laporan LB1 Puskesmas atau Rumah sakit

Nama Use Case	Melihat Data Penderita sesuai dengan Laporan LB1 Puskesmas atau Rumah sakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk melihat data laporan penderita penyakit sesuai dengan laporan LB1 dari Puskesmas atau Rumah sakit	
Pre-condition	Pengguna sudah melihat laporan LB1 Puskesmas atau Rumah Sakit.	
Post-condition	Pengguna bisa mengetahui data laporan penderita penyakit sesuai laporan LB1 dari Puskesmas atau Rumah sakit.	
Aksi Aktor	Reaksi Sistem	

Alur dasar (<i>Basic Flow</i>)	
Pengguna menekan tombol detail pada tabel Laporan LB1 Puskesmas atau Rumah sakit	Sistem menampilkan laporan penderita penyakit sesuai dengan laporan LB1 dari service yang diberikan Puskesmas atau Rumah sakit.

Tabel 12. *Use case* naratif menyimpan dan modifikasi data ICD penyakit

Nama <i>Use Case</i>	Menyimpan dan Modifikasi data ICD Penyakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk menambah data ICD baru, menghapus data, atau mengubah data.	
<i>Pre-condition</i>	Pengguna akan mengubah data ICD Penyakit yang sudah ada, menghapus data atau menambah baru.	
<i>Post-condition</i>	Pengguna bisa menambah, mengubah, atau menghapus data ICD penyakit.	
Aksi Aktor		Reaksi Sistem
Alur dasar (<i>Basic Flow</i>)		
Pengguna masuk ke menu Input data, kemudian masuk ke data ICD	Sistem menampilkan halaman form input dan tabel ICD penyakit	
Jika menambah data baru, isi form kode penyakit dan nama penyakit	Sistem akan menyimpan dan mengeluarkan pop notifikasi jika data berhasil masuk	
Pengguna menekan icon edit pada tabel	Sistem akan menampilkan form edit data ICD penyakit	
Jika klik simpan di form edit	Sistem ada notifikasi data berhasil disimpan	
Pengguna menekan icon hapus pada tabel	Sistem menghapus data	

Tabel 13. *Use Case* Naratif melihat XML SOAP dari service laporan Puskesmas dan Rumah sakit

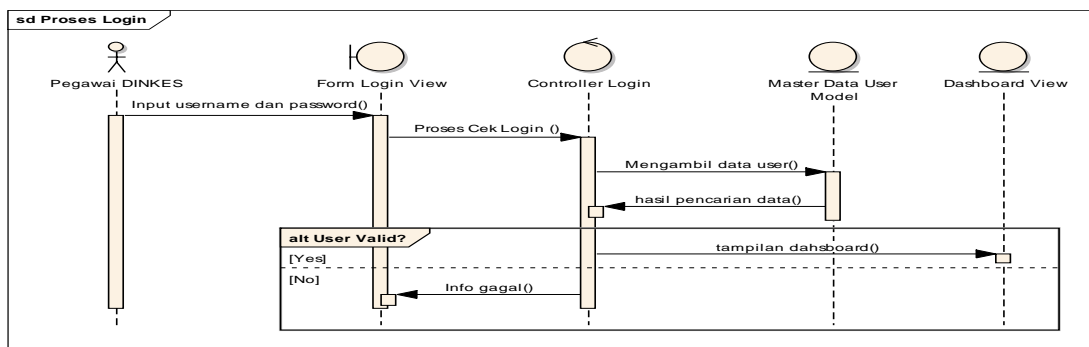
Nama <i>Use Case</i>	Melihat XML SOAP dari service laporan Puskesmas dan Rumah sakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk isi pesan komunikasi antar sistem dengan web service SOAP dalam bentuk XML.	
<i>Pre-condition</i>	Pengguna sudah masuk menu lihat laporan data yang menggunakan service dari sistem lain.	
<i>Post-condition</i>	Pengguna bisa melihat pesan SOAP XML dari service yang diberikan sistem lain.	
Aksi Aktor		Reaksi Sistem
Alur dasar (<i>Basic Flow</i>)		
Pengguna melihat laporan yang memanfaatkan service dari sistem lain		
Pengguna menekan tombol XML service SOAP	Sistem akan menampilkan pesan XML SOAP sesuai dengan service laporan dari sistem lain	

Tabel 14. Use Case Naratif Menyimpan data master Puskesmas dan Rumah sakit

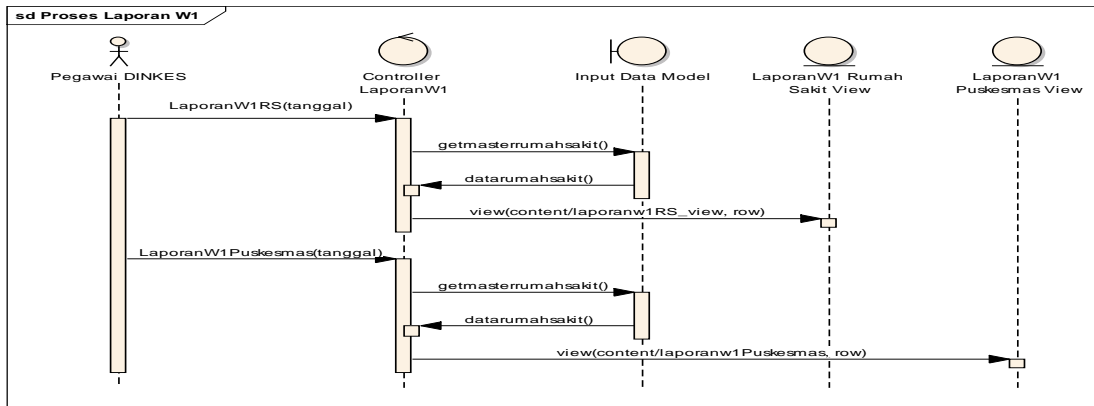
Nama Use Case	Menyimpan data master Puskesmas dan Rumah sakit	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk mengubah data master Puskesmas atau Rumah sakit	
Pre-condition	Pengguna sudah masuk menu input data dan memilih menu RS/Puskesmas	
Post-condition	Pengguna bisa melihat dan mengubah data master Rumah sakit atau Puskesmas	
Aksi Aktor	Reaksi Sistem	
Alur dasar (Basic Flow)		
Pengguna memilih menu input data kemudian memilih menu RS/Puskesmas	Sistem akan menampilkan dua pilihan menu apakah memilih Puskesmas atau Rumah sakit	
Pengguna memilih jenis operasi apakah akan memodifikasi data Puskesmas atau Rumah sakit	Sistem akan menampilkan tabel informasi Puskesmas atau Rumah sakit	
Jika tekan tombol edit	Sistem akan menampilkan data pada form	
Jika tekan tombol save change	Sistem akan menyimpan data.	
Jika tekan tombol delete	Sistem akan menghapus data yang ada di database	

Tabel 15. Use Case naratif melakukan sinkronisasi data-data laporan

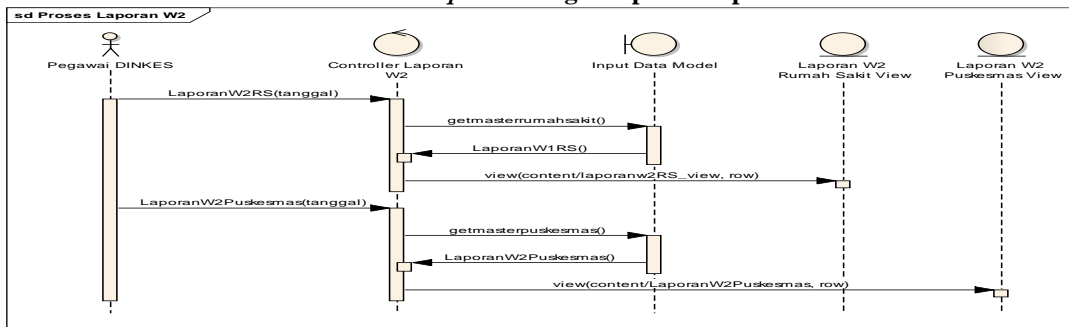
Nama Use Case	Melakukan Sinkronisasi Data-data Laporan	
Aktor	Pegawai Dinkes	
Deskripsi	Proses ini adalah sebuah kegiatan pengguna untuk melihat data-data laporan apakah sudah sinkron dengan laporan yang ada di Puskesmas atau Rumah sakit.	
Pre-condition	Pengguna belum mengetahui data yang sudah sinkron atau belum.	
Post-condition	Pengguna bisa melihat data-data yang sudah sinkron atau belum.	
Aksi Aktor	Reaksi Sistem	
Alur dasar (Basic Flow)		
Pengguna masuk pada menu sinkronisasi data	Sistem akan menampilkan data dari Puskesmas dan Rumah sakit dengan keterangan status data per tanggal tiap bulannya	
Pengguna melakukan sinkronisasi data dengan menekan tombol sinkronisasi	Sistem akan melakukan sinkronisasi data-data yang ada di database Dinkes dengan data-data yang ada di Puskesmas atau Rumah sakit	



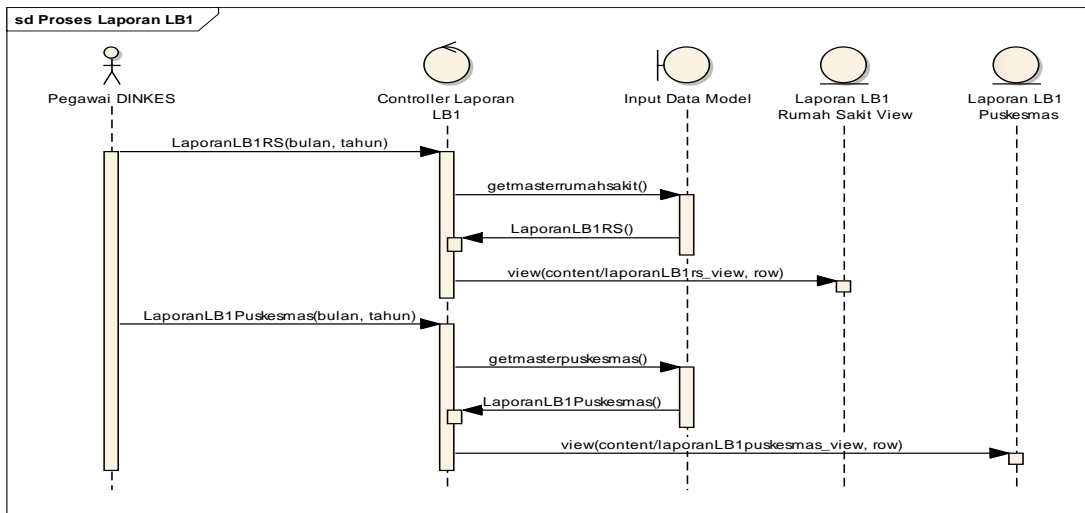
Gambar 16: Sequence diagram proses login



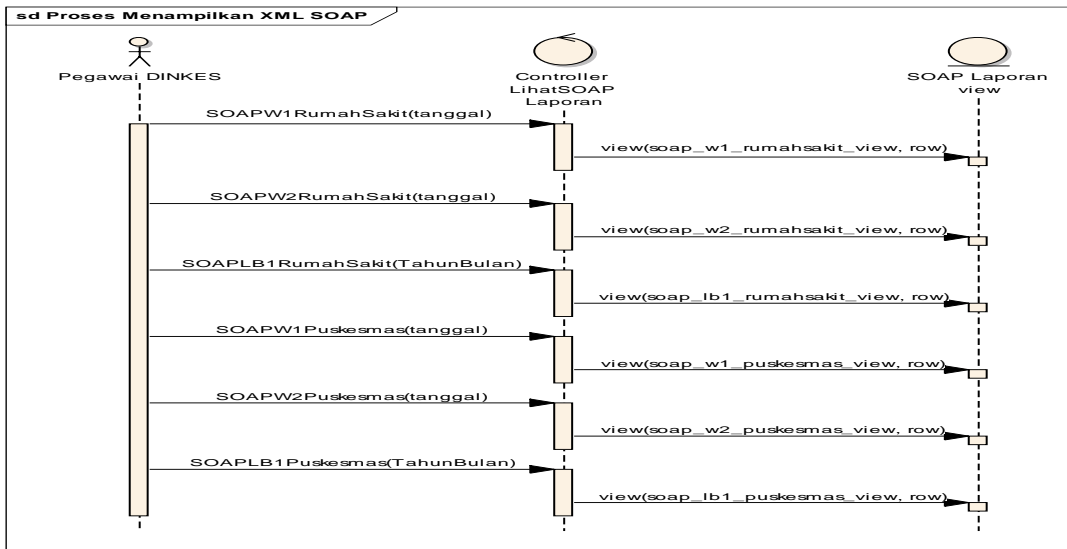
Gambar 17: Sequence diagram proses laporan W1



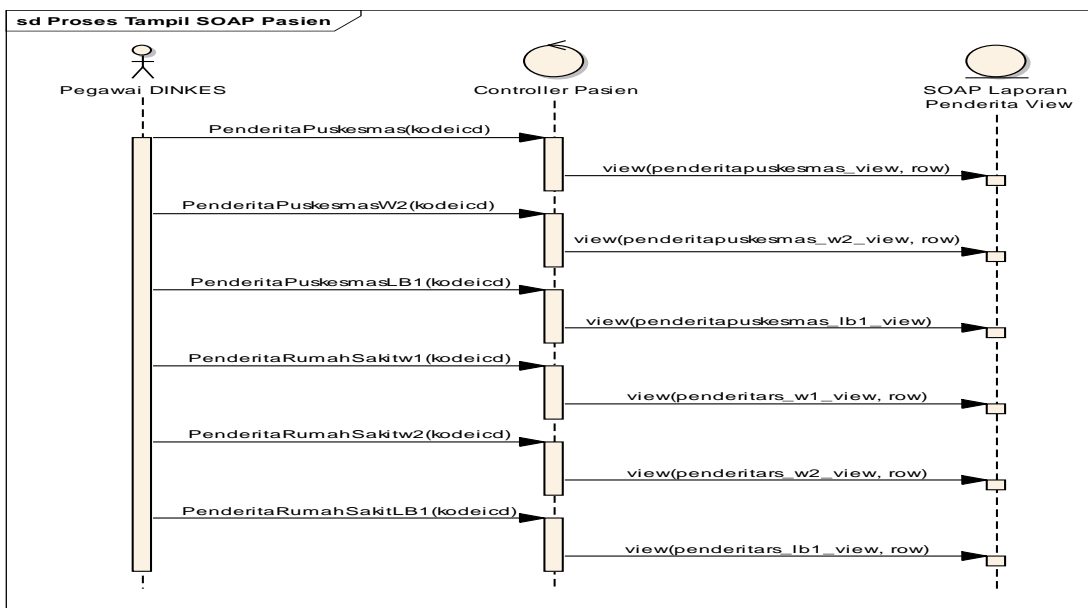
Gambar 18: Sequence diagram proses laporan W2



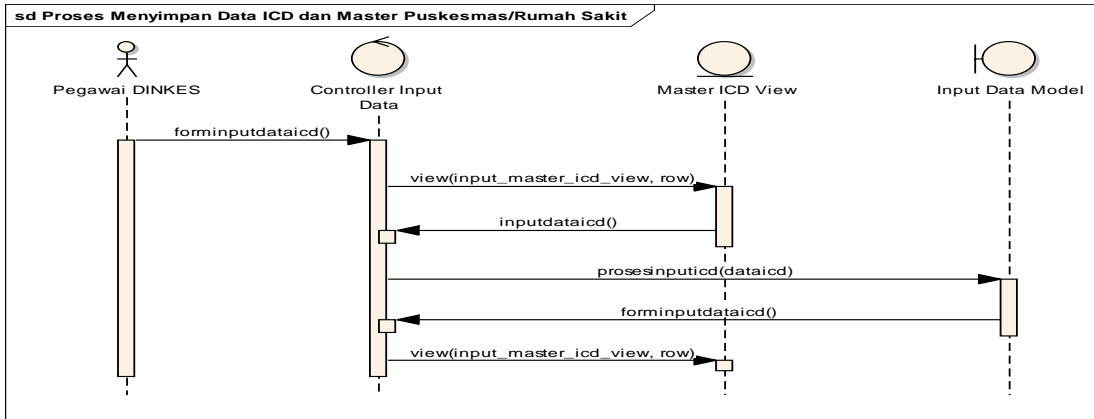
Gambar 19: Sequence diagram proses laporan LB1



Gambar 20: Sequence diagram proses menampilkan SOAP XML laporan

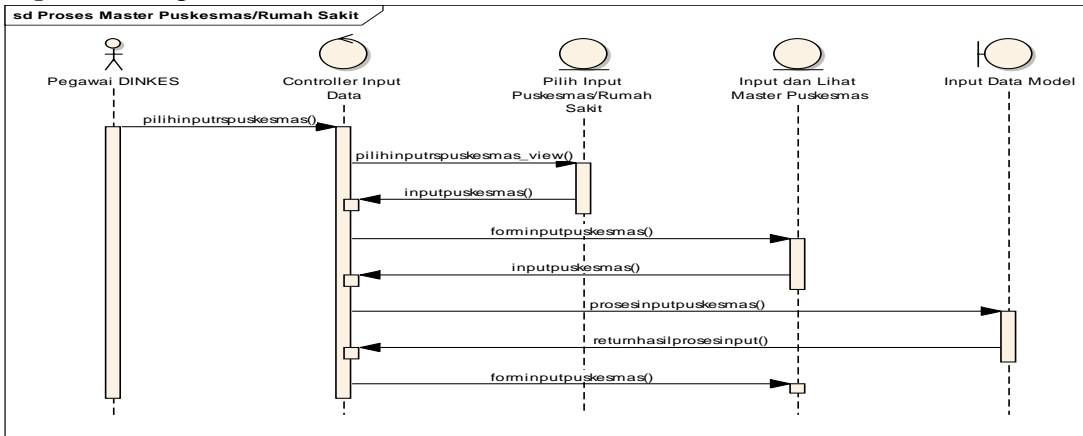


Gambar 21: Sequence proses menampilkan SOAP XML data penderita penyakit

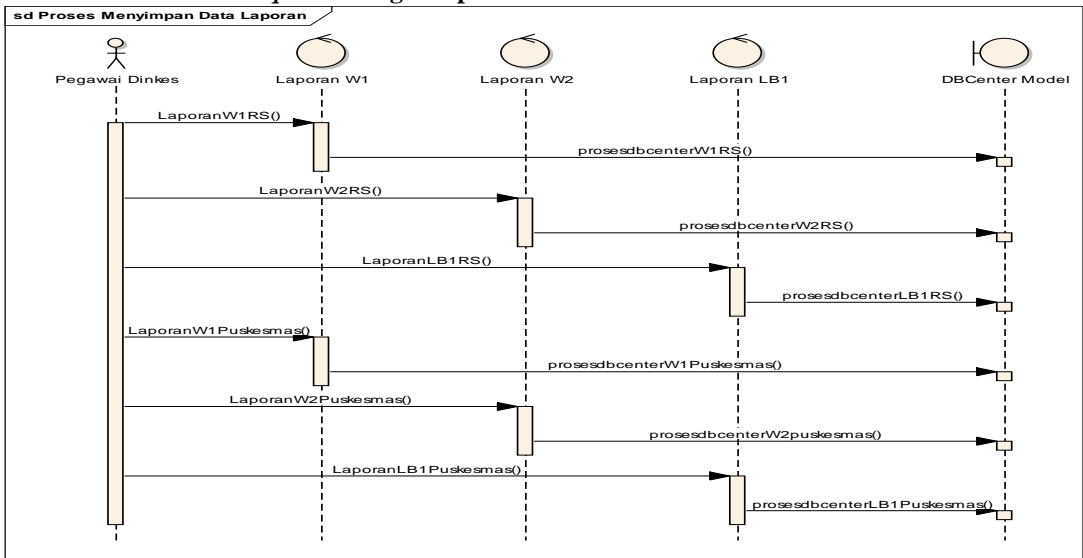


Gambar 22: Sequence diagram proses master data ICD penyakit

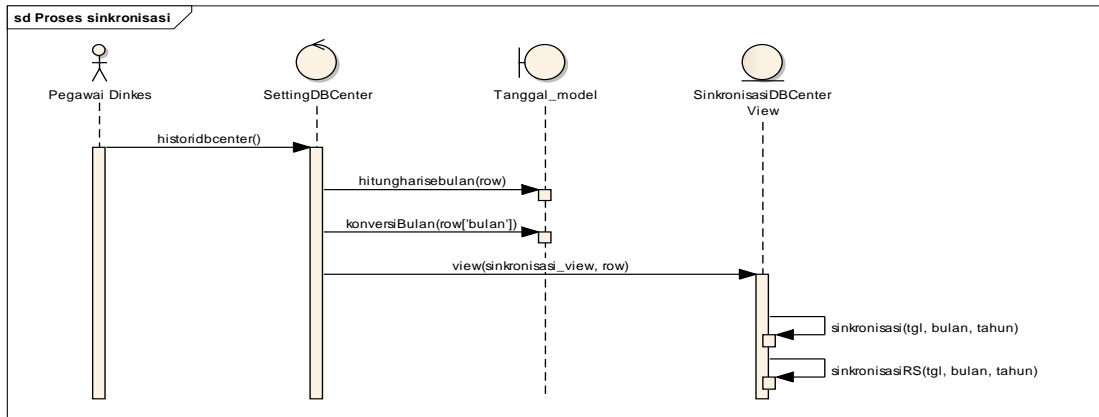
a. Sequence Diagram Proses Master Data Puskesmas atau Rumah Sakit



Gambar 23: Sequence diagram proses master data Puskesmas atau Rumah Sakit



Gambar 24: Sequence diagram proses menyimpan data – data laporan



Gambar 25: Sequence diagram proses sinkronisasi data laporan Dinkes dengan laporan Puskesmas dan Rumah Sakit

ServiceDinasKependudukan

View the [WSDL](#) for the service. Click on an operation name to view it's details.

[getDataPenduduk](#)

Gambar 26: Web service sistem Dinas Kependudukan

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:DataPenduduk"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:DataPenduduk">
<types>
<xsd:schema targetNamespace="urn:DataPenduduk">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
</xsd:schema>
</types>
<message name="getDataPendudukRequest"></message>
<message name="getDataPendudukResponse">
<part name="output" type="xsd:Array" /></message>
<portType name="ServiceDinasKependudukanPortType">
<operation name="getDataPenduduk">
<documentation>WS getDataPenduduk</documentation>
<input message="tns:getDataPendudukRequest"/>
<output message="tns:getDataPendudukResponse"/>
</operation>
</portType>
<binding name="ServiceDinasKependudukanBinding"
type="tns:ServiceDinasKependudukanPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="getDataPenduduk">
<soap:operation soapAction="urn:DataPenduduk#showData" style="rpc"/>

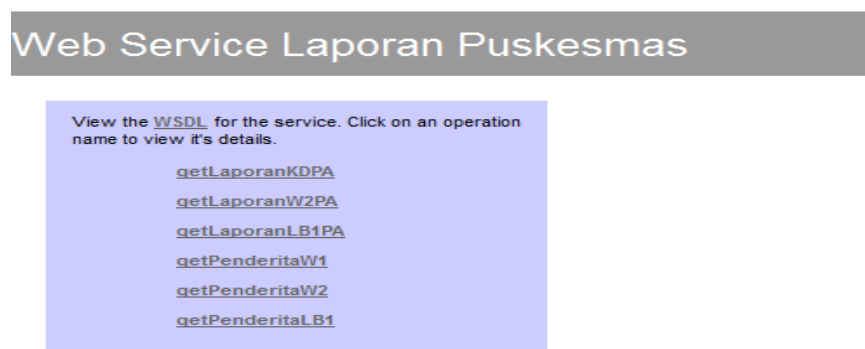
```

```

<input><soap:body use="encoded" namespace="urn:DataPenduduk"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></input>
<output><soap:body use="encoded" namespace="urn:DataPenduduk"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></output>
</operation>
</binding>
<service name="ServiceDinasKependudukan">
<port name="ServiceDinasKependudukanPort"
binding="tns:ServiceDinasKependudukanBinding">
<soap:address location="http://localhost/ta/disduk/ws/kependudukan.php" />
</port>
</service>
</definitions>

```

Gambar 27: WSDL web service sistem Dinas Kependudukan



Gambar 28: Web service Sistem Puskesmas

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:DataLaporanPuskesmas"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="urn:DataLaporanPuskesmas">
<types>
<xsd:schema targetNamespace="urn:DataLaporanPuskesmas">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
</xsd:schema>
</types>
<message name="getLaporanKDPAResponse">
<part name="tanggal" type="xsd:string" /></message>
<message name="getLaporanKDPAResponse">
<part name="output" type="xsd:Array" /></message>
<message name="getLaporanW2PAResponse">
<part name="tanggal" type="xsd:string" /></message>
<message name="getLaporanW2PAResponse">
<part name="output" type="xsd:Array" /></message>
<message name="getLaporanLB1PAResponse">
<part name="TanggalBulan" type="xsd:string" /></message>
<message name="getLaporanLB1PAResponse">
<part name="output" type="xsd:Array" /></message>
<message name="getPenderitaW1Request">
<part name="kodeicd" type="xsd:string" />
<part name="tgl" type="xsd:string" /></message>
<message name="getPenderitaW1Response">

```

```

<part name="output" type="xsd:Array" /></message>
<message name="getPenderitaW2Request">
<part name="kodeicd" type="xsd:string" />
<part name="tanggal" type="xsd:string" /></message>
<message name="getPenderitaW2Response">
<part name="output" type="xsd:Array" /></message>
<message name="getPenderitaLB1Request">
<part name="kodeicd" type="xsd:string" />
<part name="bulan" type="xsd:int" />
<part name="tahun" type="xsd:int" /></message>
<message name="getPenderitaLB1Response">
<part name="output" type="xsd:Array" /></message>
<portType name="Web Service Laporan PuskesmasPortType">
<operation name="getLaporanKDPA">
<documentation>Service Laporan KD/W1 dari Puskesmas</documentation>
<input message="tns:getLaporanKDPARequest"/>
<output message="tns:getLaporanKDPAResponse"/>
</operation>
<operation name="getLaporanW2PA">
<documentation>Service Laporan W2 dari Puskesmas</documentation>
<input message="tns:getLaporanW2PARequest"/>
<output message="tns:getLaporanW2PAResponse"/>
</operation>
<operation name="getLaporanLB1PA">
<documentation>Service Laporan LB1 dari Puskesmas</documentation>
<input message="tns:getLaporanLB1PARequest"/>
<output message="tns:getLaporanLB1PAResponse"/>
</operation>
<operation name="getPenderitaW1">
<documentation>Service Penderita Penyakit hasil diagnosa, parameter input
kode icd dan tanggal (yyyy-mm-dd) </documentation>
<input message="tns:getPenderitaW1Request"/>
<output message="tns:getPenderitaW1Response"/>
</operation>
<operation name="getPenderitaW2">
<documentation>Service Penderita Penyakit hasil diagnosa, parameter input
kode icd dan tanggal (yyyy-mm-dd) </documentation>
<input message="tns:getPenderitaW2Request"/>
<output message="tns:getPenderitaW2Response"/>
</operation>
<operation name="getPenderitaLB1">
<documentation>Service Penderita Penyakit hasil diagnosa, parameter input
kode icd, bulan (int), tahun (int) </documentation>
<input message="tns:getPenderitaLB1Request"/>
<output message="tns:getPenderitaLB1Response"/>
</operation>
</portType>
<binding name="Web Service Laporan PuskesmasBinding" type="tns:Web Service
Laporan PuskesmasPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="getLaporanKDPA">
<soap:operation soapAction="urn:LaporanKDPuskesmas#getLaporanKDPA"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanKDPuskesmas"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/></input>
<output><soap:body use="encoded" namespace="urn:LaporanKDPuskesmas"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/></output>
</operation>
<operation name="getLaporanW2PA">
<soap:operation soapAction="urn:LaporanW2Puskesmas#getLaporanW2PA"
style="rpc"/>

```



```

<input><soap:body use="encoded" namespace="urn:LaporanW2Puskesmas"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:LaporanW2Puskesmas"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="getLaporanLB1PA">
<soap:operation soapAction="urn:LaporanLB1Puskesmas#getLaporanLB1PA"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanLB1Puskesmas"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:LaporanLB1Puskesmas"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="getPenderitaW1">
<soap:operation soapAction="urn:DataPenderitaPenyakitW1#getPenderitaW1"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:DataPenderitaPenyakitW1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:DataPenderitaPenyakitW1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="getPenderitaW2">
<soap:operation soapAction="urn:DataPenderitaPenyakitW2#getPenderitaW2"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:DataPenderitaPenyakitW2"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:DataPenderitaPenyakitW2"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="getPenderitaLB1">
<soap:operation soapAction="urn:DataPenderitaPenyakitLB1#getPenderitaLB1"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:DataPenderitaPenyakitLB1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:DataPenderitaPenyakitLB1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
</binding>
<service name="Web Service Laporan Puskesmas">
<port name="Web Service Laporan PuskesmasPort" binding="tns:Web Service
Laporan PuskesmasBinding">
<soap:address
location="http://localhost/ta/puskesmasA/ws/puskesmas_service.php"/>
</port>
</service>
</definitions>

```

Gambar 29: WSDL web service Sistem Puskesmas

Web Service Laporan Rumah Sakit

View the [WSDL](#) for the service. Click on an operation name to view it's details.

[getLaporanW1RS](#)
[getLaporanW2RS](#)
[getLaporanLB1RS](#)
[getLaporanPenyakitTahunanRS](#)
[LaporanPenderitaPenyakitW1](#)
[LaporanPenderitaPenyakitW2](#)
[LaporanPenderitaPenyakitLB1](#)

Gambar 30: Web Service Sistem Rumah Sakit

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:DataLaporanRumahSakit"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="urn:DataLaporanRumahSakit">
<types>
<xsd:schema targetNamespace="urn:DataLaporanRumahSakit">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
</xsd:schema>
</types>
<message name="getLaporanW1RSRequest">
<part name="tanggal" type="xsd:string" /></message>
<message name="getLaporanW1RSResponse">
<part name="output" type="xsd:Array" /></message>
<message name="getLaporanW2RSRequest">
<part name="tanggal" type="xsd:string" /></message>
<message name="getLaporanW2RSResponse">
<part name="output" type="xsd:Array" /></message>
<message name="getLaporanLB1RSRequest">
<part name="TahunBulan" type="xsd:string" /></message>
<message name="getLaporanLB1RSResponse">
<part name="output" type="xsd:Array" /></message>
<message name="getLaporanPenyakitTahunanRSRequest">
<part name="Tahun" type="xsd:int" /></message>
<message name="getLaporanPenyakitTahunanRSResponse">
<part name="output" type="xsd:Array" /></message>
<message name="LaporanPenderitaPenyakitW1Request">
<part name="kodeicdpenyakit" type="xsd:string" />
<part name="tanggal" type="xsd:string" /></message>
<message name="LaporanPenderitaPenyakitW1Response">
<part name="output" type="xsd:Array" /></message>
<message name="LaporanPenderitaPenyakitW2Request">
<part name="kodeicdpenyakit" type="xsd:string" />
<part name="tanggal" type="xsd:string" /></message>
<message name="LaporanPenderitaPenyakitW2Response">
<part name="output" type="xsd:Array" /></message>
<message name="LaporanPenderitaPenyakitLB1Request">
<part name="kodeicdpenyakit" type="xsd:string" />
<part name="tahun" type="xsd:int" />
<part name="bulan" type="xsd:int" /></message>
<message name="LaporanPenderitaPenyakitLB1Response">
<part name="output" type="xsd:Array" /></message>
<portType name="Web Service Laporan Rumah SakitPortType">
<operation name="getLaporanW1RS">
<documentation>Service Laporan W1 dari Rumah Sakit</documentation>
<input message="tns:getLaporanW1RSRequest" />
<output message="tns:getLaporanW1RSResponse" />
</operation>
<operation name="getLaporanW2RS">
<documentation>Service Laporan W2 dari Rumah Sakit</documentation>
<input message="tns:getLaporanW2RSRequest" />
<output message="tns:getLaporanW2RSResponse" />
</operation>
<operation name="getLaporanLB1RS">
```

```

<documentation>Service Laporan LB1 dari Rumah Sakit</documentation>
<input message="tns:getLaporanLB1RSRequest"/>
<output message="tns:getLaporanLB1RSResponse"/>
</operation>
<operation name="getLaporanPenyakitTahunanRS">
<documentation>Service Laporan Penyakit Tahunan da
    ri Rumah Sakit</documentation>
<input message="tns:getLaporanPenyakitTahunanRSRequest"/>
<output message="tns:getLaporanPenyakitTahunanRSResponse"/>
</operation>
<operation name="LaporanPenderitaPenyakitW1">
<documentation>Service Laporan Penderita Penyakit W1 dengan parameter
    kodeicdpenyakit dan tanggal (yyyy-mm-dd)</documentation>
<input message="tns:LaporanPenderitaPenyakitW1Request"/>
<output message="tns:LaporanPenderitaPenyakitW1Response"/>
</operation>
<operation name="LaporanPenderitaPenyakitW2">
<documentation>Service Laporan Penderita Penyakit W2 dengan parameter
    kodeicdpenyakit dan tanggal (yyyy-mm-dd)</documentation>
<input message="tns:LaporanPenderitaPenyakitW2Request"/>
<output message="tns:LaporanPenderitaPenyakitW2Response"/>
</operation>
<operation name="LaporanPenderitaPenyakitLB1">
<documentation>Service Laporan Penderita Penyakit W2 dengan parameter
    kodeicdpenyakit, tahun (integer), bulan (integer, 1 - 12)</documentation>
<input message="tns:LaporanPenderitaPenyakitLB1Request"/>
<output message="tns:LaporanPenderitaPenyakitLB1Response"/>
</operation>
</portType>
<binding name="Web Service Laporan Rumah SakitBinding" type="tns:Web Service
    Laporan Rumah SakitPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="getLaporanW1RS">
<soap:operation soapAction="urn:LaporanW1RumahSakit#getLaporanW1RS"
    style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanW1RumahSakit"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:LaporanW1RumahSakit"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="getLaporanW2RS">
<soap:operation soapAction="urn:LaporanW2RumahSakit#getLaporanW2RS"
    style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanW2RumahSakit"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:LaporanW2RumahSakit"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="getLaporanLB1RS">
<soap:operation soapAction="urn:LaporanLB1RumahSakit#getLaporanLB1RS"
    style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanLB1RumahSakit"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:LaporanLB1RumahSakit"
    encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="getLaporanPenyakitTahunanRS">
<soap:operation
    soapAction="urn:LaporanPenyakitTahunanRumahSakit#getLaporanPenyakitTahunanRS"
    style="rpc"/>

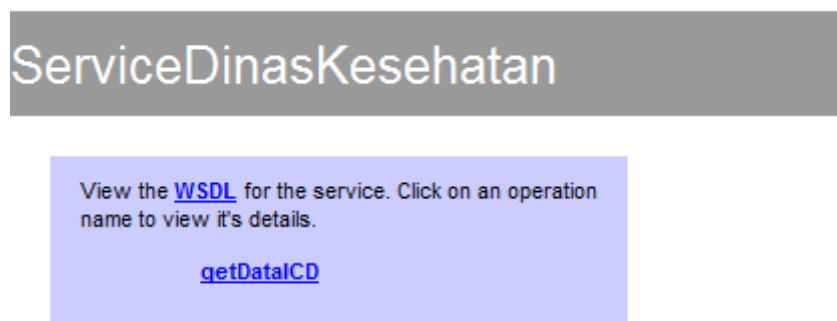
```

```

<input><soap:body use="encoded"
namespace="urn:LaporanPenyakitTahunanRumahSakit"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded"
namespace="urn:LaporanPenyakitTahunanRumahSakit"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="LaporanPenderitaPenyakitW1">
<soap:operation
soapAction="urn:LaporanPenderitaPenyakitW1#LaporanPenderitaPenyakitW1"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanPenderitaPenyakitW1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:LaporanPenderitaPenyakitW1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="LaporanPenderitaPenyakitW2">
<soap:operation
soapAction="urn:LaporanPenderitaPenyakitW2#LaporanPenderitaPenyakitW2"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanPenderitaPenyakitW2"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:LaporanPenderitaPenyakitW2"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
<operation name="LaporanPenderitaPenyakitLB1">
<soap:operation
soapAction="urn:LaporanPenderitaPenyakitLB1#LaporanPenderitaPenyakitLB1"
style="rpc"/>
<input><soap:body use="encoded" namespace="urn:LaporanPenderitaPenyakitLB1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></input>
<output><soap:body use="encoded" namespace="urn:LaporanPenderitaPenyakitLB1"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"></output>
</operation>
</binding>
<service name="Web Service Laporan Rumah Sakit">
<port name="Web Service Laporan Rumah SakitPort" binding="tns:Web Service
Laporan Rumah SakitBinding">
<soap:address location="http://localhost/ta/rs/ws/laporan_rs.php"/>
</port>
</service>
</definitions>

```

Gambar 31: WSDL web service sistem Rumah Sakit



Gambar 32: Web service sistem Dinas Kesehatan

```

http://localhost/ta/dinkes/ws/dinkes_service.php?wsdl
<?xml version="1.0" encoding="ISO-8859-1"?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tns="urn:DataICD10Penyakit"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="urn:DataICD10Penyakit">
<types>
<xsd:schema targetNamespace="urn:DataICD10Penyakit">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
</xsd:schema>
</types>
<message name="getDataICDRequest"></message>
<message name="getDataICDResponse">
<part name="output" type="xsd:Array" /></message>
<portType name="ServiceDinasKesehatanPortType">
<operation name="getDataICD">
<documentation>Web Servcie untuk mendapatkan data penyakit
ICD10</documentation>
<input message="tns:getDataICDRequest"/>
<output message="tns:getDataICDResponse"/>
</operation>
</portType>
<binding name="ServiceDinasKesehatanBinding"
type="tns:ServiceDinasKesehatanPortType">
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
<operation name="getDataICD">
<soap:operation soapAction="urn:DataICD#getDataICD" style="rpc"/>
<input><soap:body use="encoded" namespace="urn:DataICD"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></input>
<output><soap:body use="encoded" namespace="urn:DataICD"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" /></output>
</operation>
</binding>
<service name="ServiceDinasKesehatan">
<port name="ServiceDinasKesehatanPort"
binding="tns:ServiceDinasKesehatanBinding">
<soap:address location="http://localhost/ta/dinkes/ws/dinkes_service.php"/>
</port>
</service>
</definitions>

```

Gambar 33: WSDL web service sistem Dinas Kesehatan