

Sinkronisasi Pangkalan Data Rumah Sakit dengan Aplikasi *Open Source SymmetricDS*

Andi Wahyu Rahardjo Emanuel^{#1}, Teresa Liliana Wargasetia^{*2}, Ade Kurnia Surawijaya^{#3}

^{#1}Program Studi Teknik Informatika, ^{*2}Bagian Biologi, Fakultas Kedokteran
Universitas Kristen Maranatha, Jl. Suria Sumantri no. 65, Bandung

¹andi.wre@maranatha.edu

²teresa.lw@med.maranatha.edu

^{#3}Rumah Sakit Jiwa Provinsi Jawa Barat
Jl. Kolonel Masturi Km. 7, Cisarua, Bandung Barat

³dr.ade.kurnia@gmail.com

Abstract —Health is an important supporting factor in the society. One of the curative implementation of the healthcare is hospital. Supporting system based on Information Technology of hospital is called Hospital Information System (SIRS) in which this system is mandatory in every hospital. Unfortunately, this system is developed individually by the hospital so that the integration of the system in higher levels is difficult. This research demonstrates integration of Hospital Information System in small scale using SymmetricDS Open Source database replication software. It can be shown that the integration of Hospital Information System is highly possible to enable higher scale healthcare support system in the future.

Keywords— Hospital Information System, SymmetricDS, Open Source, Medical Record, Database Synchronization

I. PENDAHULUAN

Pelayanan kesehatan merupakan faktor penunjang penting bagi masyarakat yang semakin dibutuhkan di era yang serba sibuk dan serba modern ini, dan salah satu bentuk pelayanan kesehatan yang diperlukan adalah Rumah Sakit. Berbagai Rumah Sakit berlomba untuk menawarkan berbagai layanan bagi pasien dengan menonjolkan berbagai layanan dan fasilitas yang dimiliki. Untuk mendukung layanan Rumah Sakit yang semakin modern diperlukan Sistem Informasi Rumah Sakit (SIRS) yang semakin lengkap dan komprehensif. Salah satu komponen penting dalam Sistem Informasi Rumah Sakit yang harus dimiliki adalah Rekam Medis.

Rekam Medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan, dan pelayanan lain yang telah diberikan kepada pasien yang diatur dalam Permenkes 269 th 2008. Rekam Medis tersebut akan didukung oleh sebuah Sistem Informasi Rumah Sakit (SIRS) yang juga wajib dimiliki yang diatur dalam Permenkes No. 1171/MENKES/PER/VI/2011. Disamping itu, berbagai Perguruan Tinggi juga menawarkan program studi yang berhubungan dengan Rekam Medis

seperti Program D3 (Vokasi) Rekam Medis di Universitas Gadjah Mada, dan di berbagai Politeknik Kesehatan seperti di Poltek Negeri Jember, Poltekkes Malang, Poltekkes Tasikmalaya, Poltekkes Semarang dan lain – lainnya.

Meskipun Rumah Sakit sekarang ini sudah semakin modern dan canggih, namun dirasakan Sistem Informasi Rumah Sakit dan juga Rekam Medis yang dimiliki masih memiliki beberapa kelemahan khususnya dalam hal integrasi ke tingkatan yang lebih tinggi. Sistem Informasi Rumah Sakit yang dimiliki oleh berbagai rumah sakit masih berdiri sendiri – sendiri dan tidak dapat terintegrasi satu sama lain. Hal ini menimbulkan potensi kurang maksimalnya layanan kesehatan secara keseluruhan dalam melayani pasien atau calon pasien. Penelitian ini menawarkan konsep integrasi SIRS pada umumnya dan Rekam Medis pada khususnya dalam bentuk sinkronisasi dan integrasi pangkalan data untuk SIRS dan Rekam Medis.

II. TINJAUAN PUSTAKA

Sebelumnya telah terdapat berbagai penelitian yang berkaitan dengan Sistem Informasi Rumah Sakit dan Rekam Medis. Seperti Pembuatan berbagai jenis Sistem Informasi Rumah Sakit untuk beberapa daerah seperti RSUD Pacitan [1], RSUD Batam [2] dan RS Bratanata Jambi [3]. Ada pula tentang penerapan berbagai teknik seperti *datawarehouse* Rumah Sakit [4], logistik obat dengan algoritma *Apriori* [5], *e-CRM* [6], Sistem Pakar berbasis *Case-Based Reasoning* (CBR) untuk pasien kejiwaan [7], basis data terdistribusi dengan CORBA [8] dan SIRS RS Kanker Dharmasi menggunakan *Total Architecture Syntesis* [9].

Bentuk penelitian lainnya adalah tentang efektifitas dan efisiensi SIRS dan Rekam Medis dalam berbagai kondisi dan lokasi. Seperti Sistem Informasi Geografis Pelayanan Kesehatan di Kotamadya Yogyakarta [10], faktor - faktor keberhasilan implementasi SIRS menggunakan pendekatan HOT-Fit [11], penggunaan *Open Source Software* pada SIRS [12] dan beberapa penelitian lainnya. Salah satu yang

penelitian penting yang dijadikan acuan argumentasi pada penelitian ini adalah

investigasi penggunaan SIRS di DIY [13] yang menunjukkan bahwa mayoritas rumah sakit di DIY (60,42%) belum mampu melakukan pertukaran data elektronik dan hanya sebagian kecil (18,75%) yang dapat melakukan pertukaran data elektronik secara aktif. Meskipun data diatas hanya terjadi di DIY, namun tim peneliti meyakini bahwa secara nasional temuan – temuan ini tidak terlalu berbeda jauh.

Penelitian ini berbeda apabila dibandingkan dengan penelitian – penelitian sebelumnya yang berkaitan dengan SIRS dan Rekam Medis. Penelitian ini mendemonstrasikan bahwa pengintegrasian Rekam Medis dari SIRS dapat dilakukan dengan bantuan perangkat lunak pendukung sinkronisasi basis data *SymmetricDS* yang berbasis *Open Source*. Upaya ini diharapkan dapat membuka mata berbagai pihak yang berkepentingan bahwa integrasi dari Rekam Medis dari berbagai SIRS dari berbagai Rumah Sakit sangat memungkinkan untuk dilakukan dan diharapkan akan dapat mendukung layanan kesehatan masyarakat yang lebih komprehensif dan holistik bagi pemangku kepentingan di tingkatan yang lebih tinggi dari Rumah Sakit itu sendiri.

III. LANDASAN TEORI

A. Rekam Medis

Peraturan pemerintah yang mengatur tentang Rekam Medis adalah Peraturan Menteri Kesehatan nomor 269/MENKES/PER/III/2008 tentang Rekam Medis [14]. Dalam peraturan tersebut dinyatakan bahwa Rekam Medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan, dan pelayanan lain yang telah diberikan kepada pasien. Terdapat tiga jenis rekam medis yang diatur yaitu Rekam Medis untuk pasien rawat jalan, pasien rawat inap, dan pasien gawat darurat.

Rekam medis untuk pasien rawat jalan sekurang – kurangnya memuat [14]:

- Identitas pasien
- Tanggal dan waktu
- Hasil anamnesis, mencakup sekurang – kurangnya keluhan dan riwayat penyakit
- Hasil pemeriksaan fisik dan penunjang medik
- Diagnosis
- Rencana penatalaksanaan
- Pengobatan dan / atau tindakan
- Pelayanan lain yang telah diberikan kepada pasien
- Untuk pasien kasus gigi dilengkapi dengan odontogram klinik, dan
- Persetujuan tindakan bila diperlukan

Sedangkan untuk rekam medis pasien rawat inap dan perawatan satu hari sekurang – kurangnya memuat [14]:

- Identitas pasien
- Tanggal dan waktu
- Hasil anamnesis, mencakup sekurang – kurangnya keluhan dan riwayat penyakit
- Hasil pemeriksaan fisik dan penunjang medik
- Diagnosis
- Rencana penatalaksanaan
- Pengobatan dan / atau tindakan
- Persetujuan tindakan bila diperlukan
- Catatan observais klinis dan hasil pengobatan
- Ringkasan pulang (discharge summary)
- Nama dan tanda tangan dokter, dokter gigi, atau tenaga kesehatan tertentu yang memberikan pelayanan kesehatan
- Pelayanan lain yang yang dilakukan oleh tenaga kesehatan tertentu, dan
- Untuk pasien kasus gigi dilengkapi dengan odontogram klinik

Isi rekam medis untuk pasien gawat darurat, sekurang – kurangnya memuat [14]:

- Identitas pasien
- Kondisi saat pasien tiba di sarana pelayanan kesehatan
- Identitas pengantar pasien
- Tanggal dan waktu
- Hasil anamnesis, mencakup sekurang – kurangnya keluhan dan riwayat penyakit
- Hasil pemeriksaan fisik dan penunjang medik
- Diagnosis
- Pengobatan dan / atau tindakan
- Ringkasan kondisi pasien sebelum meninggalkan pelayanan unit gawat darurat dan rencana tindak lanjut
- Nama dan tanda tangan dokter, dokter gigi, atau tenaga kesehatan tertentu yang memberikan pelayanan kesehatan
- Sarana transportasi yang digunakan bagi pasien yang akan dipindahkan ke sarana pelayanan kesehatan lain, dan
- Pelayanan lain yang telah diberikan kepada pasien.

B. Sistem Informasi Rumah Sakit

Sistem Informasi Rumah Sakit (SIRS) adalah proses pengumpulan, pengolahan, dan penyajian data Rumah Sakit yang berbentuk aplikasi. Sistem Informasi Rumah Sakit (SIRS) diatur dalam Peraturan Menteri Kesehatan nomor 1171/MENKES/PER/VI/2011 tentang Sistem Informasi Rumah Sakit. Peraturan tersebut menyatakan bahwa Rumah Sakit wajib memiliki SIRS yang meliputi [15]:

- Data identitas Rumah Sakit
- Data ketenagaan yang bekerja di Rumah Sakit
- Data rekapitulasi kegiatan pelayanan
- Data kompilasi penyakit / morbiditas pasien rawat inap, dan

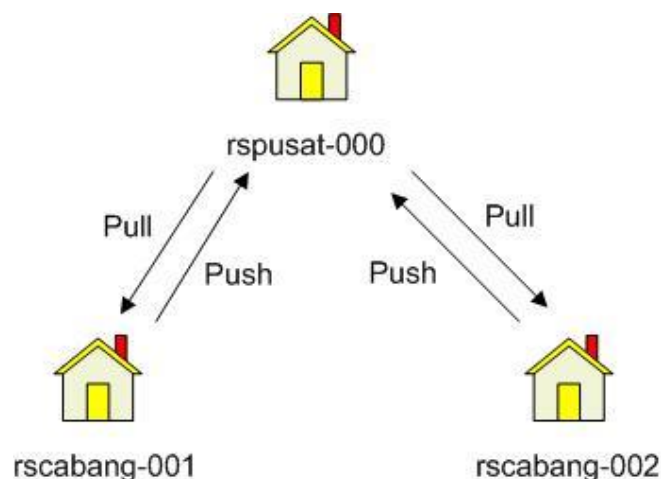
- Data kompilasi penyakit / morbiditas pasien rawat jalan.

C. SymmetricDS

SymmetricDS adalah aplikasi berbasis *Open Source* yang berfungsi untuk replikasi basisdata [16]. Aplikasi ini memiliki tiga keunggulan utama yaitu mendukung berbagai platform, skalabilitas, dan konfigurasi yang fleksibel. Proses instalasi *SymmetricDS* ini yaitu dengan mengunduh berkas versi terbaru (saat penelitian ini versi terbaru adalah versi 3.8.18) di situs resmi dan kemudian mengekstraknya di folder */opt* pada komputer berbasis *Ubuntu* yang dipergunakan untuk simulasi penelitian. Untuk kepentingan penelitian ini *SymmetricDS* akan dipasang untuk bekerja pada basisdata *MySQL* dan dikelola dengan mempergunakan aplikasi web *phpMyAdmin*. *SymmetricDS* sendiri membutuhkan *Java Runtime* untuk dapat bekerja dengan semestinya.

IV. DESAIN SIMULASI

Skenario yang dibuat untuk simulasi sinkronisasi SIRS dan Rekam medis adalah sebagai berikut:



Gambar 1. Skenario Simulasi Sinkronisasi SIRS dan Rekam Medis

Pada gambar 1 di atas digambarkan dua buah Rumah Sakit Cabang (dinamai *rscabang-001* dan *rscabang-002*) yang datanya dihubungkan oleh sebuah Rumah Sakit Pusat (dinamai *rspusat-000*). Berbagai data yang berkaitan dengan SIRS dan Rekam Medis akan disinkronisasikan antar ketiga rumah sakit tersebut dengan asumsi skenario sebagai berikut:

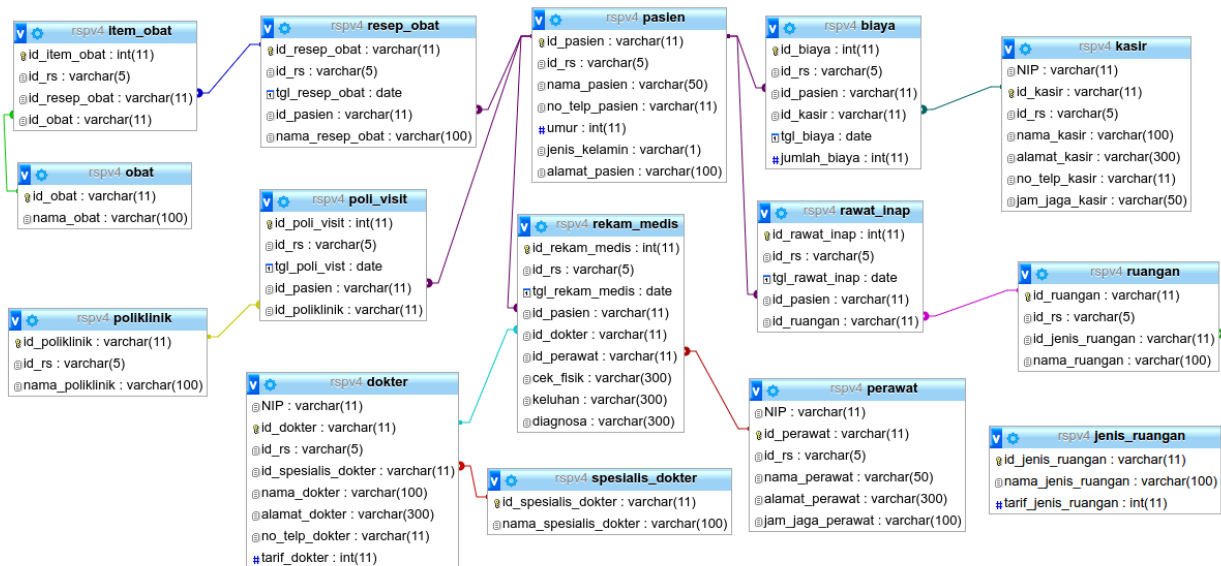
1. Data ketenagaan dari Rumah Sakit dikendalikan oleh Rumah Sakit Pusat yaitu data dokter, perawat, dan pegawai (kasir).
2. Data suplai obat yang tersedia di masing – masing Rumah Sakit Cabang dikendalikan oleh Rumah Sakit Pusat.

3. Data poliklinik yang berisi jenis – jenis klinik yang dilayani oleh setiap Rumah Sakit Cabang dikendalikan oleh Rumah Sakit Pusat.
4. Data pasien diisikan oleh setiap Rumah Sakit Cabang yang kemudian akan terkumpul secara otomatis di Rumah Sakit Pusat.
5. Data Rekam Medis diisikan di setiap Rumah Sakit Cabang yang kemudian akan terkumpul secara otomatis di Rumah Sakit Pusat.
6. Data resep diisikan di setiap Rumah Sakit Cabang yang kemudian akan terkumpul secara otomatis di Rumah Sakit Pusat.
7. Data pembayaran diisikan di setiap Rumah Sakit Cabang yang kemudian akan terkumpul secara otomatis di Rumah Sakit Pusat.

Beberapa asumsi tambahan yang dipergunakan dalam mendesain simulasi ini adalah sebagai berikut:

1. Komputer yang dipakai sebagai Rumah Sakit Pusat, dan setiap Rumah Sakit Cabang kesemuanya menggunakan Sistem Operasi *Ubuntu* versi 16.04 LTS dan *Java RE (Runtime Environment)* versi 8.0.
2. Semua komputer sudah terpasang aplikasi *SymmetricDS* di folder */opt*. Proses instalasi dilakukan dengan mengekstrak berkas instalasi *SymmetricDS* yang berupa berkas zip pada folder tersebut.
3. Asumsi semua komputer sudah terpasang *Apache2*, *PHP7*, dan *MySQL Server 5.7* dan juga *PHPMyAdmin* untuk administrasi basis data *MySQL* melalui tampilan web.
4. Setiap komputer di Rumah Sakit Pusat dan setiap Rumah Sakit Cabang diberikan alamat IP spesifik, dan *hostname* pada setiap komputer tersebut dibuat identik dengan alamat IP-nya. Pada penelitian ini komputer untuk Rumah Sakit Pusat memiliki alamat IP 10.2.2.66, alamat IP untuk Rumah Sakit Cabang 001 adalah 10.2.2.120 dan alamat IP untuk Rumah Sakit Cabang 002 adalah 10.2.2.30.
5. *Hostname* dari setiap komputer dibuat identik dengan alamat IP dari setiap komputer.
6. Komputer di Rumah Sakit Pusat dan setiap Rumah Sakit Cabang terkoneksi via jaringan Ethernet via *hub / switch* dan bisa berkomunikasi satu dengan yang lainnya, misalnya bisa melakukan *ping* antar komputer.

Adapun yang menjadi skema pangkalan data SIRS merupakan pengembangan dari situs Aina-tunk [17] dengan pengembangan dan penambahan sesuai kebutuhan penelitian. Gambar 2 dibawah ini menunjukkan hubungan antar tabel dari SIRS yang dipergunakan dalam simulasi.



Gambar 2. Pangkalan Data SIRS

Gambar 2 diatas menunjukkan sebuah simulasi SIRS yang terdiri dari 14 tabel. Ketigabelas tabel tersebut adalah:

1. Tabel perawat
2. Tabel dokter
3. Tabel pasien
4. Tabel rekam medis
5. Tabel jenis ruangan
6. Tabel ruangan
7. Tabel rawat_inap
8. Tabel obat
9. Tabel resep_obat
10. Tabel item_obat
11. Tabel kasir
12. Tabel biaya
13. Tabel poliklinik
14. Tabel poli_visit

Langkah pengembangan komputer yang menjadi *rspusat-000* adalah sebagai berikut:

1. Membuat basis data kosong dengan nama *rsp* di MySQL yang terpasang di komputer Rumah Sakit Pusat..
2. Mengisi basis data kosong tersebut dengan skema basis data simulasi SIRS seperti yang ditunjukkan pada gambar 3 dibawah ini:

```
-- Database untuk simulasi sinkronisasi data RS BP2D Jawa Barat
-- creator: Andi WRE
CREATE TABLE `perawat` (
  `NIP` varchar(11) NOT NULL,
```

```
`id_perawat` varchar(11) NOT NULL,
`id_rs` varchar(5) NOT NULL,
`nama_perawat` varchar(50) NOT NULL,
`alamat_perawat` varchar(300) NOT NULL,
`jam_jaga_perawat` varchar(100) NOT NULL,
PRIMARY KEY (`id_perawat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `spesialis_dokter` (
  `id_spesialis` varchar(11) NOT NULL,
  `nama_spesialis` varchar(100) NOT NULL,
PRIMARY KEY (`id_spesialis`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `dokter` (
  `NIP` varchar(11) NOT NULL,
  `id_dokter` varchar(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `id_spesialis` varchar(11) NOT NULL,
  `nama_dokter` varchar(100) NOT NULL,
  `alamat_dokter` varchar(300) NOT NULL,
  `no_telp_dokter` varchar(11) NOT NULL,
  `tarif_dokter` int(11) NOT NULL,
PRIMARY KEY (`id_dokter`),
FOREIGN KEY(`id_spesialis`) REFERENCES
`spesialis_dokter` (`id_spesialis`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `pasien` (
  `id_pasien` varchar(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `nama_pasien` varchar(50) NOT NULL,
  `no_telp_pasien` varchar(11) NOT NULL,
  `umur` int(11) NOT NULL,
  `jenis_kelamin` varchar(1) NOT NULL,
  `alamat_pasien` varchar(100) NOT NULL,
  `id_dokter` varchar(11) NOT NULL,
  `id_perawat` varchar(11) NOT NULL,
PRIMARY KEY (`id_pasien`),
FOREIGN KEY(`id_dokter`) REFERENCES `dokter` (`id_dokter`),
FOREIGN KEY(`id_perawat`) REFERENCES `perawat` (`id_perawat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `rekam_medis` (
  `id_rekam_medis` int(11) NOT NULL AUTO_INCREMENT,
  `id_rs` varchar(5) NOT NULL,
  `tgl_rekam_medis` date NOT NULL,
  `id_pasien` varchar(11) NOT NULL,
  `id_dokter` varchar(11) NOT NULL,
  `id_perawat` varchar(11) NOT NULL,
  `cek_fisik` varchar(300) NOT NULL,
  `keluhan` varchar(300) NOT NULL,
  `diagnosa` varchar(300) NOT NULL,
PRIMARY KEY(`id_rekam_medis`),
FOREIGN KEY(`id_dokter`) REFERENCES `dokter` (`id_dokter`),
FOREIGN KEY(`id_perawat`) REFERENCES `perawat` (`id_perawat`),
```

```

FOREIGN KEY(`id_pasien`) REFERENCES `pasien`(`id_pasien`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;;

CREATE TABLE `jenis_ruangan` (
  `id_jenis_ruangan` varchar(11) NOT NULL,
  `nama_ruangan` varchar(100) NOT NULL,
  `tarif_ruangan` int(11) NOT NULL,
  PRIMARY KEY (`id_jenis_ruangan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `ruangan` (
  `id_ruangan` varchar(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `id_jenis_ruangan` varchar(11) NOT NULL,
  `nama_ruangan` varchar(100) NOT NULL,
  PRIMARY KEY (`id_ruangan`),
  FOREIGN KEY(`id_jenis_ruangan`) REFERENCES
`jenis_ruangan`(`id_jenis_ruangan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `rawat_inap` (
  `id_rawat_inap` int(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `id_pasien` varchar(11) NOT NULL,
  `id_ruangan` varchar(11) NOT NULL,
  PRIMARY KEY (`id_rawat_inap`),
  FOREIGN KEY(`id_pasien`) REFERENCES `pasien`(`id_pasien`),
  FOREIGN KEY(`id_ruangan`) REFERENCES `ruangan`(`id_ruangan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `obat` (
  `id_obat` varchar(11) NOT NULL,
  `nama_obat` varchar(100) NOT NULL,
  PRIMARY KEY (`id_obat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `resep_obat` (
  `id_resep` varchar(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `id_pasien` varchar(11) NOT NULL,
  `nama_resep` varchar(100) NOT NULL,
  PRIMARY KEY (`id_resep`),
  FOREIGN KEY(`id_pasien`) REFERENCES `pasien`(`id_pasien`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `item_obat` (
  `id_item_obat` int(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `id_resep` varchar(11) NOT NULL,
  `id_obat` varchar(11) NOT NULL,
  PRIMARY KEY (`id_item_obat`),
  FOREIGN KEY(`id_obat`) REFERENCES `obat`(`id_obat`),
  FOREIGN KEY(`id_resep`) REFERENCES `resep_obat`(`id_resep`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `kasir` (
  `NIP` varchar(11) NOT NULL,
  `id_kasir` varchar(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `nama_kasir` varchar(100) NOT NULL,
  `alamat_kasir` varchar(300) NOT NULL,
  `no_telp_kasir` varchar(11) NOT NULL,
  `jam_jaga_kasir` varchar(50) NOT NULL,
  PRIMARY KEY (`id_kasir`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `biaya` (
  `id_bayar` varchar(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `id_pasien` varchar(11) NOT NULL,
  `id_kasir` varchar(11) NOT NULL,
  `tgl_bayar` date NOT NULL,
  `jumlah_bayar` int(11) NOT NULL,
  PRIMARY KEY (`id_bayar`),
  FOREIGN KEY(`id_pasien`) REFERENCES `pasien`(`id_pasien`),
  FOREIGN KEY(`id_kasir`) REFERENCES `kasir`(`id_kasir`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `poliklinik` (
  `id_poli` varchar(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `nama_poli` varchar(100) NOT NULL,
  PRIMARY KEY (`id_poli`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

CREATE TABLE `poli_visit` (
  `id_poli_visit` int(11) NOT NULL,
  `id_rs` varchar(5) NOT NULL,
  `id_pasien` varchar(11) NOT NULL,
  `id_poli` varchar(11) NOT NULL,
  PRIMARY KEY (`id_poli_visit`),
  FOREIGN KEY(`id_pasien`) REFERENCES `pasien`(`id_pasien`),
  FOREIGN KEY(`id_poli`) REFERENCES `poliklinik`(`id_poli`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Gambar 3. Isi skrip sql untuk basisdata SIRS

- Memasukkan file *rspusat.properties* ke folder *engines* di lokasi folder dari *SymmetricDS*. Pada berkas tersebut perlu disesuaikan pada bagian *username* dan *password* dari MySQL serta alamat IP dari Rumah Sakit Pusat. Isinya dari berkas *rspusat.properties* adalah sebagai seperti pada Gambar 4 berikut ini (hanya ditampilkan yang relevan saja):

```

engine.name=rspusat-000

# The class name for the JDBC Driver
db.driver=com.mysql.jdbc.Driver

# The JDBC URL used to connect to the database
db.url=jdbc:mysql://localhost/rsp?tinyInttisBit=false

# The user to login as who can create and update tables
db.user=root

# The password for the user to login as
db.password=andi.wre

# Registration
auto.registration=true
auto.reload=true

# The HTTP URL of the root node to contact for registration
# change the IP accordingly
registration.url=http://10.2.2.66:31415/sync/rspusat-000

# Do not change these for running the demo
group.id=rspusat
external.id=000

```

Gambar 4. Isi skrip rspusat.properties

- Menjalankan *SymmetricDS* untuk pertama kali dengan menjalankan perintah *./bin/sym* pada *terminal* dengan akses *root*. Akan ada 46 tabel tambahan yang dibuat oleh aplikasi *SymmetricDS* untuk melancarkan proses sinkronisasi.
- Mematikan *SymmetricDS* setelah proses pembentukan 46 tabel selesai dengan mengetikkan CTRL-C pada *Terminal*. Selanjutnya adalah memasukkan konfigurasi sinkronisasi dengan menjalankan skrip *sql* seperti yang ditunjukkan pada Gambar 5 berikut ini di *phpMyAdmin* pada basisdata *rsp*:

```

-- membersihkan dari data sebelumnya
delete from sym_trigger_router;
delete from sym_trigger;
delete from sym_router;
delete from sym_channel where channel_id in ('channel_pusat',
'dokter');
delete from sym_channel where channel_id in ('channel_pusat',
'spesialis_dokter');
delete from sym_channel where channel_id in ('channel_pusat',
'perawat');
delete from sym_channel where channel_id in ('channel_pusat',
'kasir');
delete from sym_channel where channel_id in ('channel_pusat',
'poliklinik');
delete from sym_channel where channel_id in ('channel_pusat',
'jenis_ruangan');
delete from sym_channel where channel_id in ('channel_pusat',
'ruangan');

```

```

delete from sym_channel where channel_id in ('channel_pusat',
'obat');
delete from sym_channel where channel_id in ('channel_cabang',
'pasien');
delete from sym_channel where channel_id in ('channel_cabang',
'rekam_medis');
delete from sym_channel where channel_id in ('channel_cabang',
'poli_visit');
delete from sym_channel where channel_id in ('channel_cabang',
'resep_obat');
delete from sym_channel where channel_id in ('channel_cabang',
'item_obat');
delete from sym_channel where channel_id in ('channel_cabang',
'biaya');
delete from sym_node_group_link;
delete from sym_node_group;
delete from sym_node_host;
delete from sym_node_identity;
delete from sym_node_security;
delete from sym_node;

-- pembuatan channel: channel_pusat dan channel_cabang
-- membuat channel_pusat untuk update data dari rspusat ke rscabang
(dokter, spesialis dokter, perawat, kasir, poliklinik,
jenis_ruangan, ruangan, obat)
insert into sym_channel
(channel_id, processing_order, max_batch_size, enabled, description)
values('channel_pusat', 1, 100000, 1, 'data yang diupdate dari
rspusat ke rscabang');

-- membuat channel_cabang untuk update data dari rscabang ke
rspusat (pasien, rekam_medis, poli_visit, resep_obat, biaya,
item_obat)
insert into sym_channel
(channel_id, processing_order, max_batch_size, enabled, description)
values('channel_cabang', 1, 100000, 1, 'data yang diupdate dari
rscabang ke rspusat');

-- pembuatan sym_node_group dan sym_node_group_link
-- membuat node rspusat dan node rscabang
insert into sym_node_group (node_group_id) values ('rspusat');
insert into sym_node_group (node_group_id) values ('rscabang');

-- membuat sym_node_group link
insert into sym_node_group_link (source_node_group_id,
target_node_group_id, data_event_action) values ('rspusat',
'rscabang', 'W');
insert into sym_node_group_link (source_node_group_id,
target_node_group_id, data_event_action) values ('rscabang',
'rspusat', 'P');

-- setting trigger untuk rspusat ke rscabang
-- trigger_dokter mengupdate data dokter dari rspusat ke rscabang
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_dokter','dokter','channel_pusat',current_timestam
p,current_timestamp);

-- trigger_spesialis_dokter mengupdate data spesialis_dokter dari
rspusat ke rscabang
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_spesialis_dokter','spesialis_dokter','channel_pusat
',current_timestamp,current_timestamp);

-- trigger_perawat mengupdate data perawat dari rspusat ke rscabang
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_perawat','perawat','channel_pusat',current_timestam
p,current_timestamp);

-- trigger_kasir mengupdate data kasir dari rspusat ke rscabang
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_kasir','kasir','channel_pusat',current_timestamp,cu
rrent_timestamp);

-- trigger_poliklinik mengupdate data kasir dari rspusat ke
rscabang
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_poliklinik','poliklinik','channel_pusat',current_t
imestamp,current_timestamp);

-- trigger_jenis_ruangan mengupdate data jenis_ruangan dari rspusat
ke rscabang
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_jenis_ruangan','jenis_ruangan','channel_pusat',curr
ent_timestamp,current_timestamp);

-- trigger_ruangan mengupdate data ruangan dari rspusat ke rscabang
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_ruangan','ruangan','channel_pusat',current_timestam
p,current_timestamp);

-- trigger_obat mengupdate data obat dari rspusat ke rscabang via
channel_pusat
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_obat','obat','channel_pusat',current_timestamp,curr
ent_timestamp);

-- setting trigger dari rscabang ke rspusat
-- trigger_pasien mengupdate data pasien
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_pasien','pasien','channel_cabang',current_timestam
p,current_timestamp);

-- trigger_rekam_medis mengupdate data rekam_medis
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_rekam_medis','rekam_medis','channel_cabang',current
_timestamp,current_timestamp);

-- trigger_poli_visit mengupdate data poli_visit
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_poli_visit','poli_visit','channel_cabang',current_t
imestamp,current_timestamp);

-- trigger_resep_obat mengupdate data resep_obat
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_resep_obat','resep_obat','channel_cabang',current_t
imestamp,current_timestamp);

-- trigger_item_obat mengupdate data resep_obat
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_item_obat','item_obat','channel_cabang',current_tim
estamp,current_timestamp);

-- trigger_biaya mengupdate data biaya
insert into sym_trigger
(trigger_id,source_table_name,channel_id,last_update_time,create_t
ime)
values('trigger_biaya','biaya','channel_cabang',current_timestamp,c
urrent_timestamp);

-- setting sym_router untuk mengatur rute sinkronisasi antar data
-- rutepusat_ke_cabang arah dari node rspusat ke node rscabang
insert into sym_router
(router_id,source_node_group_id,target_node_group_id,router_type,cr
eate_time,last_update_time)
values('rutepusat_ke_cabang', 'rspusat', 'rscabang',
'default',current_timestamp, current_timestamp);

-- rutecabang_ke_pusat arah dari node rscabang ke node rspusat
insert into sym_router
(router_id,source_node_group_id,target_node_group_id,router_type,cr
eate_time,last_update_time)
values('rutecabang_ke_pusat', 'rscabang', 'rspusat',
'default',current_timestamp, current_timestamp);

-- rutepusat_ke_satu_cabang
insert into sym_router
(router_id,source_node_group_id,target_node_group_id,router_type,ro
uter_expression,create_time,last_update_time)
values('rutepusat_ke_satu_cabang', 'rspusat', 'rscabang',
'column','ID_RS=:EXTERNAL_ID or
OLD_ID_RS=:EXTERNAL_ID',current_timestamp, current_timestamp);

-- rute_trigger_dokter melalui rutepusat_ke_satu_cabang
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,initial_load_select,last_u
pdate_time,create_time)
values('trigger_dokter','rutepusat_ke_satu_cabang',100,'id_rs=''$
(externalId)''',current_timestamp,current_timestamp);

-- rute_trigger_spesialis_dokter melalui rutepusat_ke_cabang
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_t
ime)
values('trigger_spesialis_dokter','rutepusat_ke_cabang', 100,

```

```

current_timestamp, current_timestamp);

-- rute trigger perawat melalui rutepusat_ke_satu_cabang
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,initial_load_select,last_u
pdate_time,create_time)
values('trigger_perawat','rutepusat_ke_satu_cabang',100,'id_rs='$(
externalId)','','current_timestamp,current_timestamp);

-- rute trigger kasir melalui rutepusat_ke_satu_cabang
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,initial_load_select,last_u
pdate_time,create_time)
values('trigger_kasir','rutepusat_ke_satu_cabang',100,'id_rs='$(
ternalId)','','current_timestamp,current_timestamp);

-- rute trigger poliklinik melalui rutepusat_ke_satu_cabang
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,initial_load_select,last_u
pdate_time,create_time)
values('trigger_poliklinik','rutepusat_ke_satu_cabang',100,'id_rs='
'$(externalId)','','current_timestamp,current_timestamp);

-- rute trigger jenis ruangan melalui rutepusat_ke_cabang
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_ti
me)
values('trigger_jenis_ruangan','rutepusat_ke_cabang', 100,
current_timestamp, current_timestamp);

-- rute trigger ruangan melalui rutepusat_ke_cabang
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,initial_load_select,last_u
pdate_time,create_time)
values('trigger_ruangan','rutepusat_ke_satu_cabang',100,'id_rs='$(
externalId)','','current_timestamp,current_timestamp);

-- rute trigger obat melalui rutepusat_ke_cabang
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_ti
me)
values('trigger_obat','rutepusat_ke_cabang', 100, current_timestamp,
current_timestamp);

-- rute trigger pasien melalui rutecabang_ke_pusat
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_ti
me)
values('trigger_pasien','rutecabang_ke_pusat', 200,
current_timestamp, current_timestamp);

-- rute trigger rekam medis melalui rutecabang_ke_pusat
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_ti
me)
values('trigger_rekam_medis','rutecabang_ke_pusat', 200,
current_timestamp, current_timestamp);

-- rute trigger poli_visit melalui rutecabang_ke_pusat
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_ti
me)
values('trigger_poli_visit','rutecabang_ke_pusat', 200,
current_timestamp, current_timestamp);

-- rute trigger resep_obat melalui rutecabang_ke_pusat
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_ti
me)
values('trigger_resep_obat','rutecabang_ke_pusat', 200,
current_timestamp, current_timestamp);

-- rute trigger item_obat melalui rutecabang_ke_pusat
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_ti
me)
values('trigger_item_obat','rutecabang_ke_pusat', 200,
current_timestamp, current_timestamp);

-- rute trigger biaya melalui rutecabang_ke_pusat
insert into sym_trigger_router
(trigger_id,router_id,initial_load_order,last_update_time,create_ti
me)
values('trigger_biaya','rutecabang_ke_pusat', 200,
current_timestamp, current_timestamp);

-- membuat sym_node
insert into sym_node
(node_id,node_group_id,external_id,sync_enabled,sync_url,schema_ver
sion,symmetric_version,database_type,database_version,heartbeat_tim
e,timezone_offset,batch_to_send_count,batch_in_error_count,created_
at_node_id)
values
('000','rspusat','000',1,null,null,null,null,null,current_timestam
p,null,0,0,'000');

```

```

insert into sym_node
(node_id,node_group_id,external_id,sync_enabled,sync_url,schema_ver
sion,symmetric_version,database_type,database_version,heartbeat_tim
e,timezone_offset,batch_to_send_count,batch_in_error_count,created_
at_node_id)
values
('001','rscabang','001',1,null,null,null,null,null,current_timestam
p,null,0,0,'000');
insert into sym_node
(node_id,node_group_id,external_id,sync_enabled,sync_url,schema_ver
sion,symmetric_version,database_type,database_version,heartbeat_tim
e,timezone_offset,batch_to_send_count,batch_in_error_count,created_
at_node_id)
values
('002','rscabang','002',1,null,null,null,null,null,current_timestam
p,null,0,0,'000');

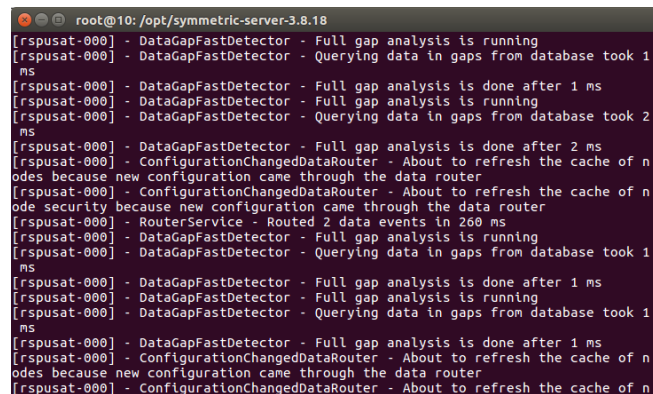
-- informasi sekuriti untuk setiap node
insert into sym_node_security
(node_id,node_password,registration_enabled,registration_time,initi
al_load_enabled,initial_load_time,created_at_node_id)
values
('000','5d1c92bbacbe2edb9e1ca5dbb0e481',0,current_timestamp,0,curre
nt_timestamp,'000');
insert into sym_node_security
(node_id,node_password,registration_enabled,registration_time,initi
al_load_enabled,initial_load_time,created_at_node_id)
values
('001','5d1c92bbacbe2edb9e1ca5dbb0e481',1,null,1,null,'000');
insert into sym_node_security
(node_id,node_password,registration_enabled,registration_time,initi
al_load_enabled,initial_load_time,created_at_node_id)
values
('002','5d1c92bbacbe2edb9e1ca5dbb0e481',1,null,1,null,'000');

-- identitas node, untuk rspusat 000, untuk rscabang mulai 001 dst
insert into sym_node_identity values ('000');

```

Gambar 5. Isi skrip sinkronisasi di RS Pusat

- Menjalankan kembali *SymmetricDS* dengan perintah `./bin/sym`. Setelah beberapa saat maka *rspusat-000* sudah dalam keadaan siap untuk dikoneksikan dengan setiap *rscabang-001* dan *rscabang-002* seperti yang ditunjukkan pada Gambar 6 berikut ini.



Gambar 6. *rspusat-000* Siap Terkoneksi dengan Rumah Sakit Cabang

Langkah membangun dan mengkoneksikan setiap Rumah Sakit Cabang (dalam penelitian ini adalah *rscabang-001* dan *rscabang-002*) adalah sebagai berikut:

- Membuat basis data kosong yang dinamakan *rscabang* di *MySQL* di setiap komputer Rumah Sakit Cabang.
- Mengisi basis data *rscabang* dengan skema seperti pada Gambar 7 dibawah ini. Untuk menghindari bentrokan data *primary key* saat terintegrasi di *rspusat-*

000 maka beberapa tabel yang di *rscabang-001* yang bersifat *AUTO_INCREMENT* dimulai dengan angka yang berbeda. Dibawah ini adalah skema basis data untuk *rscabang-001* yang pengaturan *AUTO_INCREMENT*-nya dimulai dari 1000, sedangkan untuk *rscabang-002* sama namun pengaturan *AUTO_INCREMENT* dimulai dari 2000 (skrip dibawah ini hanya menunjukkan penambahan dari skrip sql pada Gambar 3).

```
ALTER TABLE `rekam_medis` AUTO_INCREMENT = 1000;
ALTER TABLE `rawat_inap` AUTO_INCREMENT=1000;
ALTER TABLE `item_obat` AUTO_INCREMENT = 1000;
ALTER TABLE `biaya` AUTO_INCREMENT=1000;
ALTER TABLE `poli_visit` AUTO_INCREMENT=1000;
```

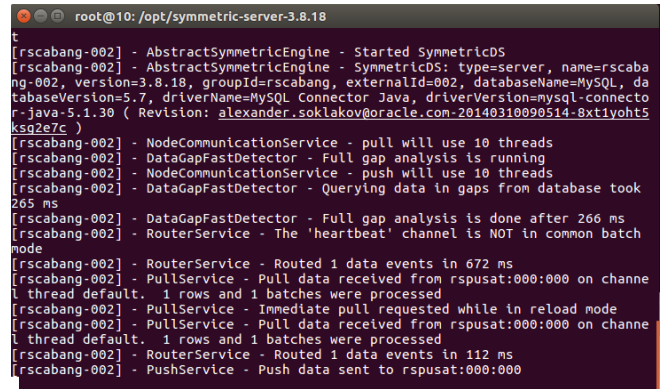
Gambar 7. Penambahan skrip sql untuk basisdata rscabang 001

- Memasukkan berkas *rscabang-001.properties* folder *engines* pada *SymmetricDS* di Rumah Sakit Cabang 001 dan *rscabang-002.properties* di folder yang sama di Rumah Sakit Cabang 002. Gambar 8 adalah skrip *rscabang-002.properties*, untuk *rscabang-001.properties* perlu disesuaikan .

```
engine.name=rscabang-002
# The class name for the JDBC Driver
db.driver=com.mysql.jdbc.Driver
# The JDBC URL used to connect to the database
db.url=jdbc:mysql://localhost/rscabang?tinyInt1isBit=false
# The user to login as who can create and update tables
db.user=root
# The password for the user to login as
db.password=andi.wre
# The HTTP URL of the root node to contact for registration
registration.url=http://10.2.2.66:31415/sync/rspusat-000
# Do not change these for running the demo
group.id=rscabang
external.id=002
```

Gambar 8. Isi skrip rscabang-002.properties untuk RS Cabang 002

- Menjalankan *SymmetricDS* untuk pertama kali dengan mengeksekusi perintah `./bin/sym` di *Terminal* . Terdapat 46 tabel baru ditambahkan oleh aplikasi *SymmetricDS* dan isian akan otomatis terisi berdasarkan isian di tabel - tabel sinkronisasi di *rspusat-000*. Pada kondisi ini proses sinkronisasi telah siap, dengan skema basis data yang baru seperti yang ditunjukkan pada Gambar 9 berikut ini.



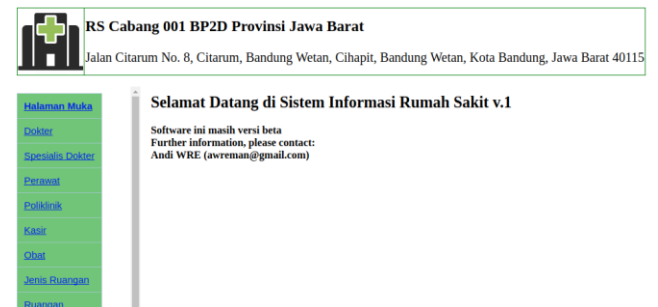
Gambar 9. rscabang-002 Terkoneksi dengan rspusat-000

V. IMPLEMENTASI

Untuk ujicoba implementasi dari simulasi ini maka dibuatlah situs web sederhana dari Rumah Sakit Pusat dan kedua Rumah Sakit Cabang (*rscabang-001* dan *rscabang-002*). Gambar 10 menunjukkan tampilan awal dari Rumah Sakit Pusat, sedangkan Gambar 11 menunjukkan tampilan salah satu Rumah Sakit Cabang yaitu *rscabang-001*. Data yang dipakai untuk simulasi SIRS ini adalah data *dummy* yang dibuat mirip dengan kondisi Rumah Sakit yang sesungguhnya.



Gambar 10. Tampilan Awal RS Pusat



Gambar 11. Tampilan Awal RS Cabang 001

Pada Rumah Sakit Pusat, apabila terjadi penambahan data dokter, perawat, kasir, ruangan, poliklinik, maka otomatis akan terduplikasi di Rumah Sakit Cabang yang dituju. Gambar 12 menunjukkan data dokter dari semua cabang yang terlihat di Rumah Sakit Pusat. Gambar 13 menunjukkan proses penambahan data dokter di Rumah Sakit Pusat untuk ditempatkan di Rumah Sakit Cabang 002.

id_rekam_medis	id_rs	id_pasien	id_dokter	id_perawat	cek_fisik	keluhan	diagnosa	
19602001	DKT001	001	SPS000	Irysad Muhammad, dr	Taman Golf Arcamanik	081788987	25000	
19620202	DKT002	001	SPS007	Pranata Audy, dr.SpB	Setra Duta	0899776276	100000	
19650203	DKT003	002	SPS001	Jaidi, dr.SpA	Margahayu Raya	088888767	80000	
19750204	DKT004	002	SPS003	Anugrah Pratama, dr.SPD	Duta Asri Cihanjuang	0876543456	90000	
19700205	DKT005	002	SPS005	Dendi Abdul Rohim, H.dr.SpB	Gateway Pasteur	087646435696000		
19730206	DKT006	001	SPS003	Kurniawan Aditya, dr.SpOK	Sudirman Suites	087333234	110000	
19720207	DKT007	001	SPS006	Arifin Muhammad, dr.SpS.Mkes	Paralyngan Rumah Villa	0876112345	150000	
19710208	DKT008	002	SPS008	Gyats Haitsam, H.dr.SpKK	Melong Green	0876534546	100000	
19710209	DKT009	002	SPS005	Dono Aditia, dr.SpTHT	Pharmindo	0811234532295000		
19720210	DKT010	001	SPS008	Zeffry Irwanto, dr.SpM	Bumi Asri Cihanjuang	087654566	100000	

Gambar 12. Data Dokter di RS Pusat

TAMBAH DOKTER

NIP:

id_dokter:

id_rs:

id_spesialis_dokter:

nama_dokter:

alamat_dokter:

no_telp_dokter:

tarif_dokter:

Gambar 13. Penambahan Dokter di RS Pusat untuk RS Cabang 002

Setelah beberapa saat akan terjadi proses sinkronisasi dimana data dokter tersebut akan muncul di situs SIRS Rumah Sakit Cabang 002 seperti yang ditunjukkan pada Gambar 14.

id_rekam_medis	id_rs	id_pasien	id_dokter	id_perawat	cek_fisik	keluhan	diagnosa	
19650203	DKT003	002	SPS001	Jaidi, dr.SpA	Margahayu Raya	088888767	80000	
19750204	DKT004	002	SPS003	Anugrah Pratama, dr.SPD	Duta Asri Cihanjuang	0876543456	90000	
19700205	DKT005	002	SPS005	Dendi Abdul Rohim, H.dr.SpB	Gateway Pasteur	087646435696000		
19710208	DKT008	002	SPS008	Gyats Haitsam, H.dr.SpKK	Melong Green	0876534546	100000	
19710209	DKT009	002	SPS005	Dono Aditia, dr.SpTHT	Pharmindo	0811234532295000		
1972024000	DKT012	002	SPS003	dr. Andi Suparman, Sp. OG	Perum Cibiru Indah II no. 15	0876446456	124000	

Gambar 14. Tampilan Dokter di Rumah Sakit Cabang 002

Pengujian juga dilakukan untuk data spesialis dokter, jenis ruangan, dan obat. Data ketiganya yang ditambahkan di Rumah Sakit Pusat akan otomatis tersinkronisasi di setiap Rumah Sakit Cabang.

Untuk proses kebalikannya yaitu sinkronisasi data dari Rumah Sakit Cabang ke Rumah Sakit Pusat dilakukan untuk data pasien, rekam medis, poli visit, resep obat, biaya, item obat dapat diisikan di Rumah Sakit Cabang dan otomatis terkumpul di Rumah Sakit Pusat. Sebagai contoh pada Gambar 15 adalah data Rekam Medis di Rumah Sakit Cabang 001. Rekam medis tersebut ditambah dengan data baru seperti yang ditunjukkan pada Gambar 16.

id_rekam_medis	id_rs	id_pasien	id_dokter	id_perawat	cek_fisik	keluhan	diagnosa
1000	001	2017-08-07	PSN111	DKT001	PRW001	Tekanan darah 120/200	Pusing dan pusing sampai nyaris pingsan Tekanan darah terlalu tinggi, disarankan mengurangi konsumsi garam. Diberikan obat penurun tekanan darah
1001	001	2017-08-07	PSN121	DKT010	PRW002	Tekanan normal	Muntah - muntah Diberi obat anti muntah dan maal

Gambar 15. Tampilan Rekam Medis di RS Cabang 001

TAMBAH REKAM_MEDIS

id_rekam_medis:

id_rs:

tgl_rekam_medis:

id_pasien:

id_dokter:

id_perawat:

cek_fisik:

keluhan:

diagnosa:

Gambar 16. Penambahan Rekam Medis di RS Cabang 001

Setelah beberapa saat kemudian akan terjadi sinkronisasi dimana data Rekam Medis tersebut akan terkumpul di RS Pusat seperti yang ditunjukkan pada Gambar 17.

id_rekam_medis	id_rs	id_pasien	id_dokter	id_perawat	cek_fisik	keluhan	diagnosa
1000	001	2017-08-07	PSN111	DKT001	PRW001	Tekanan darah 120/200	Pusing dan pusing sampai nyaris pingsan Tekanan darah terlalu tinggi, disarankan mengurangi konsumsi garam. Diberikan obat penurun tekanan darah
1001	001	2017-08-07	PSN121	DKT010	PRW002	Tekanan normal	Muntah - muntah Diberi obat anti muntah dan maal
1002	001	2017-08-08	PSN116	DKT002	PRW005	Tekanan darah 100/200	Pusing dan maal Tekanan darah terlalu tinggi, diberikan obat penurun tekanan darah
2000	002	2017-08-07	PSN118	DKT003	PRW007	Tekanan darah 80/180	Pusing dan maal Overworked, harus istirahat total3 hari.Diberikan surat istirahat, diberikan obat penenang
2001	002	2017-08-07	PSN120	DKT008	PRW008	Pucat	Pusing dan merasa lemas Diberi multivitamin

Gambar 17. Data Rekam Medis Gabungan di RS Pusat

Pengujian juga dilakukan untuk data lainnya yaitu data pasien, poli visit, resep obat, biaya, dan item obat yang menunjukkan bahwa proses sinkronisasi telah berhasil dilakukan dengan baik.

Sebagai informasi, aplikasi *SymmetricDS* yang dipergunakan juga dapat dijalankan sebagai layanan atau service dengan cara menjalankan aplikasi dengan perintah `./bin/sym_service` pada *Terminal* dengan akses *root*. Untuk menjalankan layanan secara permanen setiap komputer dihidupkan dapat dilakukan dengan cara menjalankan perintah `./bin/sym_service install` pada *Terminal* dengan akses *root*. Mematikan layanan dapat pula dilakukan dengan menjalankan perintah `./bin/sym_service uninstall` pada *Terminal* dengan akses *root*

VI. PEMBAHASAN

Penelitian ini telah berhasil mendemonstrasikan kemungkinan integrasi dari dua simulasi SIRS dengan Rekam Medisnya ke dalam sebuah SIRS dan Rekam Medis terintegrasi di Rumah Sakit Pusat. Keberhasilan ini memiliki beberapa keterbatasan yaitu:

1. Skema basis data dari setiap Rumah Sakit Cabang harus sama, beserta skema basisdata di Rumah Sakit Pusat juga harus mirip dengan Rumah Sakit Cabangnya. Ini menunjukkan skenario yang mungkin untuk sebuah konsorsium atau grup rumah sakit yang SIRS-nya dikembangkan oleh pengembangan yang sama. Integrasi dari berbagai SIRS dengan skema basis data yang berbeda membutuhkan pengujian yang mendalam khususnya yang berkaitan dengan konsistensi data, aturan penamaan data, kelengkapan data dan lain-lainnya.
2. Ujicoba yang dilakukan masih skala kecil dengan data yang masih *dummy* yang dibuat *serealistic* mungkin. Ujicoba dengan data yang nyata sekarang ini mengalami kendala terutama berkaitan dengan peraturan tentang kerahasiaan pasien.
3. Ujicoba yang dilakukan masih dalam lingkup laboratorium komputer dengan koneksi intranet. Koneksi dengan jaringan yang lebih kompleks dan dengan koneksi Internet akan menimbulkan berbagai tantangan lainnya yang perlu diantisipasi.

Penelitian awal ini ditunjukkan untuk memberikan contoh bahwa sinkronisasi dari SIRS dan Rekam Medis sangat mungkin dilakukan. Data terintegrasi khususnya bagian Rekam Medis akan membuka peluang bagi berbagai pemangku kepentingan di tingkatan yang lebih tinggi dari Rumah Sakit misalnya dinas Kesehatan untuk bisa meningkatkan layanan kesehatan bagi masyarakat dengan lebih efisien, efektif dan komprehensif.

VII. KESIMPULAN

Layanan kesehatan merupakan salah satu faktor pendukung yang dibutuhkan oleh masyarakat, dan salah satu wujudnya adalah layanan Rumah Sakit. Setiap Rumah Sakit memiliki SIRS dan juga Rekam Medis, namun biasanya setiap SIRS dan Rekam Medis tersebut dikembangkan secara mandiri sehingga sulit untuk diintegrasikan menjadi

sebuah pangkalan data SIRS dan Rekam Medis yang terpusat.

Penelitian ini berhasil mendemonstrasikan integrasi dari simulasi SIRS dan Rekam Medis dari dua buah Rumah Sakit Cabang yang disinkronisasikan ke sebuah Rumah Sakit Pusat dengan bantuan aplikasi *Open Source* untuk replikasi basis data *SymmetricDS*. Beberapa pengujian telah membuktikan bahwa integrasi berbagai data yang seharusnya ada di SIRS dan Rekam Medis dapat digabungkan dalam sebuah pangkalan data terintegrasi.

Temuan dalam penelitian ini masih memiliki beberapa keterbatasan yang perlu diselidiki dan dipelajari lebih lanjut. Penelitian tahap selanjutnya akan berusaha merumuskan dan menjawab beberapa tantangan misalnya integrasi SIRS dan Rekam Medis dari rumah sakit dengan skema yang berbeda, pengujian dengan koneksi Internet dan berbagai skenario lainnya yang semakin menunjukkan kondisi yang semakin nyata.

UCAPAN TERIMA KASIH

Terima kasih kami ucapkan untuk BP2D Provinsi Jawa Barat yang telah memberikan pendanaan untuk penelitian ini, dan juga untuk Universitas Kristen Maranatha yang memberikan dukungan dalam penyelesaian penelitian ini.

DAFTAR PUSTAKA

- [1] G. Susanto and Sukadi, "Sistem Informasi Rekam Medis pada Rumah Sakit Umum Daerah (RSUD) Pacitan Berbasis Web," *Jurnal Speed - Sentra Penelitian Engineering dan Edukasi*, vol. 3, no. 4, pp. 18 - 24, 2011.
- [2] R. V. Imbar and Y. Kurniawan, "Perancangan Sistem Informasi Pelayanan Medis Rawat Jalan Poliklinik Kebidanan dan Kandungan pada RSUD Batam," *Jurnal Sistem Informasi*, vol. 7, no. 1, pp. 53 - 67, 2012.
- [3] R. Afriany N. and B. Purnama, "Analisis dan Perancangan Sistem Informasi Rekam Medis di Rumah Sakit TK. IV dr. Bratanata Jambi," *Jurnal Manajemen Sistem Informasi*, vol. 1, no. 2, pp. 147 - 158, 201.
- [4] H. Antonius and E. Widjaja, "Data Warehouse pada Rumah Sakit," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) 2010*, Yogyakarta, 2010.
- [5] E. Buulolo, "Implementasi Algoritma Apriori pada Sistem Persediaan Obat (Studi Kasus: Apotik Rumah Sakit Estomihi Medan)," *Pelita Informatika Budi Darma*, vol. 4, no. 1, pp. 71 - 83, 2013.
- [6] W. Kaswidjanti, H. Sofyan and L. Hawari, "Aplikasi e-CRM Berbasis Web pada Rumah Sakit," *TELEMATIKA*, vol. 8, no. 1, pp. 17 - 24, 2011.
- [7] R. Retnowati and A. Pujiyanta, "Implementasi Case Based Reasoning pada Sistem Pakar dalam Menentukan Jenis Gangguan Kejiwaan," *Jurnal Sarjana Teknik Informatika*, vol. 1, no. 1, pp. 69 - 78, 2013.
- [8] B. E. Purnama and A. Ashari, "Distributed Data Patient in Medical Record Information System," *International Journal of Scientific dan Technology Research*, vol. 2, no. 8, pp. 116 - 122, 2013.
- [9] M. Y. Ricky, "Analisis dan Perancangan Sistem Informasi Laboratorium Rumah Sakit," *Comtech*, vol. 1, no. 2, pp. 561 - 574,

- 2010.
- [10] Y. B. Hege, U. Lestari and E. Kumalasari, "Sistem Informasi Geografis (SIG) Pelayanan Kesehatan di Kotamadya Yogyakarta Berbasis Web," *Jurnal SCRIPT*, vol. 1, no. 2, 2014.
- [11] A. B. Saputra, "Identifikasi Faktor - Faktor Keberhasilan Implementasi Sistem Informasi Manajemen Rumah Sakit," *Jurnal Penelitian Pers dan Komunikasi Pembangunan*, vol. 19, no. 3, pp. 135 - 148, 2016.
- [12] M. F. Pataha, G. Y. Sanjaya and L. Trisnantoro, "Penggunaan Open Source Software (OSS) di Rumah Sakit," *Jurnal Informatika*, vol. 8, no. 2, pp. 933 - 942, 2014.
- [13] E. Hariana, G. Y. Sanjaya, A. R. Rahmanti, B. Murtiningsih and E. Nugroho, "Penggunaan Sistem Informasi Manajemen Rumah Sakit (SIMRS) di DIY," in *Seminar Nasional Sistem Informasi Indonesia 2013*, Surabaya, 2013.
- [14] Kementerian Kesehatan RI, "Peraturan Menteri Kesehatan nomor 269/MENKES/PER/III/2008 tentang Rekam Medis," Maret 2008. [Online]. Available: http://dinkes.surabaya.go.id/portal/files/permenkes/dok_dinkes_87.pdf. [Accessed 4 September 2017].
- [15] Kementerian Kesehatan RI, "Peraturan Menteri Kesehatan nomor 1171/MENKES/PER/VI/2011 tentang Sistem Informasi Rumah Sakit," Juni 2011. [Online]. Available: <http://rsjmutiarasukma.ntbprov.go.id/file/Permenkes%201171%20Ttg%20SIRS%20Rev-6.pdf>. [Accessed 4 September 2017].
- [16] JumpMind Inc., "SymmetricDS Open Source Database Replication," [Online]. Available: <http://www.symmetricds.org/>. [Accessed 4 September 2017].
- [17] A. Tunk, "Rancangan Database Sistem Informasi," April 2015. [Online]. Available: <http://ainatunk.blogspot.co.id/2015/04/rancangan-database-sistem-informasi.html>. [Accessed 4 September 2017].
- [18] H. Mulyanarko, B. E. Purnama and Sukadi, "Pembangunan Sistem Informasi Billing pada Rumah Sakit Umum Daerah (RSUD) Kabupaten Pacitan Berbasis Web," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 4, no. 2, pp. 73 - 78, 2013.