

**RANCANG BANGUN SISTEM PELAYANAN PENDAFTARAN  
PASIEN RUMAH SAKIT BERBASIS *WEB SERVICE* DENGAN  
PROTOKOL SOAP WSDL DAN BANTUAN ALAT  
DETEKSI SIDIK JARI**

**SKRIPSI**

**Oleh:**

**HIZBUN NASOR  
NIM. 11650050**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2018**

**RANCANG BANGUN SISTEM PELAYANAN PENDAFTARAN  
PASIEN RUMAH SAKIT BERBASIS *WEB SERVICE* DENGAN  
PROTOKOL SOAP WSDL DAN BANTUAN ALAT  
DETEKSI SIDIK JARI**

**SKRIPSI**

**Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:**

**HIZBUN NASOR  
NIM. 11650050**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2018**

**HALAMAN PERSETUJUAN**

**RANCANG BANGUN SISTEM PELAYANAN PENDAFTARAN  
PASIEN RUMAH SAKIT BERBASIS *WEB SERVICE* DENGAN  
PROTOKOL SOAP WSDL DAN BANTUAN ALAT  
DETEKSI SIDIK JARI**

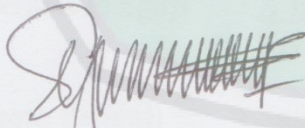
**SKRIPSI**

Oleh:

**HIZBUN NASOR  
NIM. 11650050**

**Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal: Januari 2018**

**Dosen Pembimbing I**



**A'la Syaqui, M.Kom  
NIP. 19771201 200801 1 007**

**Dosen Pembimbing II**



**Suprivono, M.Kom  
NIDT. 19841010 20160801 1 078**

**Mengetahui,  
Ketua Jurusan Teknik Informatika**



**Dr. Cahyo Crysdiyan  
NIP. 19740424 200901 1 008**

HALAMAN PENGESAHAN

**RANCANG BANGUN SISTEM PELAYANAN PENDAFTARAN  
PASIEAN RUMAH SAKIT BERBASIS *WEB SERVICE* DENGAN  
PROTOKOL SOAP WSDL DAN BANTUAN ALAT  
DETEKSI SIDIK JARI**

**SKRIPSI**

Oleh :  
**HIZBUN NASOR**  
**NIM. 11650050**

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan Dinyatakan Diterima sebagai Salah Satu Persyaratan Untuk Memperoleh Gelar Sarjana Komputer Strata Satu (S.Kom)

**Tanggal, Januari 2018**

**Susunan Dewan Penguji :**

**Tanda Tangan**

**Penguji Utama : Irwan Budi Santoso, M.Kom**  
**NIP. 19770103 201101 1 004**

(  )

**Ketua Penguji : Ajib Hanani, M.T**  
**NIDT. 19840731 20160801 1 076**

(  )

**Sekretaris Penguji : A'la Syauqi, M.Kom**  
**NIP. 19771201 200801 1 007**

(  )

**Anggota Penguji : Supriyono, M. Kom**  
**NIDT. 19841010 20160801 1 078**

(  )

**Mengetahui dan Mengesahkan  
Ketua Jurusan Teknik Informatika**



**Dr. Cahyo Crysdiyan**  
**NIP. 19740424 200901 1 008**

## HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan dibawah ini:

Nama : Hizbun Nasor

NIM : 11650050

Jurusan : Teknik Informatika

Fakultas : Sains dan Teknologi

Judul : Rancang Bangun Sistem Pelayanan Pendaftaran Pasien Rumah Sakit Berbasis *Web Service* dengan Menggunakan Protokol SOAP WSDL dan Bantuan Alat Deteksi Sidik Jari

Menyatakan dengan sebenar – benarnya bahwa skripsi yang saya tulis ini tidak terdapat unsur – unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka. Apabila ternyata hasil penelitian ini terbukti terdapat unsur – unsur penjiplakan, maka saya bersedia untuk mempertanggungjawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 19 Desember 2017

Yang membuat pernyataan



Hizbun Nasor  
NIM. 11650050

## HALAMAN PERSEMBAHAN

*Karya ini saya persembahkan kepada:*

*Kedua orang tua saya, bapak Ahmad Rohim dan Ibu Syamsiah yang telah merawat, membesarkan saya serta memberikan pendidikan mental, karakter dan agama dengan penuh kesabaran dan kasih sayang dan tidak henti-hentinya memberikan dorongan semangat, yang begitu saya sayangi, saya cintai dan saya hormati.*

*Nenek saya mbah Satri yang telah memberikan kasih sayang yang begitu besar dan banyak memberikan pelajaran yang begitu berarti.*

*Kakak – kakak saya tercinta, Lailatul Masruroh, Siti Mutammimah, Nurul Hidayati, Abdul Hamid, dan M. Sarifuddin*

*Keponakan – keponakan saya tersayang, Wiwit Nahdia Sawintri, Dwi Quen Gilfana, dan M. Haikal Afif Dosen Teknik Informatika, terutama Bapak. Cahyo, Bapak Syaqui, dan Bapak Supri.*

*Teman – teman jurusan teknik informatika (INTEGER 11), Uje, Anis, Ulfa, Tum, Irvan, Bil dan teman – teman kontrakan, Rizal, Ahmadi, Faiq, Alfan, Nizar, Ridho, Rezak. Dan semua teman – teman yang tidak dapat saya sebutkan satu persatu semoga tetap jalan dan lindungan Allah SWT. Amin*

## MOTTO

أنا عند ظن عبدي بي

*Sesungguhnya Allah berfirman: “Aku sebagaimana prasangka hambaku kepada-Ku. Aku bersamanya jika ia berdoa kepada-Ku.” [HR.Turmudzi]*

إِنْ أَحْسَنْتُمْ أَحْسَنْتُمْ لِأَنْفُسِكُمْ وَإِنْ أَسَأْتُمْ فَلَهَا

*“Jika kamu berbuat baik (berarti) kamu berbuat baik bagi dirimu sendiri dan jika kamu berbuat jahat, Maka (kejahatan) itu bagi dirimu sendiri”*



## KATA PENGANTAR

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Alhamdulillah segala puji dan syukur kehadirat Allah SWT atas berkah, rahmat serta hidayah-Nya, sholawat serta salam penulis haturkan kepada baginda Rosulullah SAW sebagai pembawa rahmat bagi seluruh alam ini yang senantiasa dilimpahkan kepada penulis, sehingga dapat menyelesaikan skripsi ini dengan judul “Rancang Bangun Sistem Pelayanan Pendaftaran Pasien Rumah Sakit Berbasis *Web Service* dengan Menggunakan Protokol SOAP WSDL dan Bantuan Alat Deteksi Sidik Jari”.

Dalam menyelesaikan skripsi ini banyak hambatan dan rintangan yang penulis hadapi namun pada akhirnya dapat terlampaui dengan adanya bimbingan, dukungan dan bantuan dari berbagai pihak baik secara moral maupun spiritual. Untuk itu pada kesempatan ini penulis menyampaikan terimakasih kepada:

1. Bapak Ahmad Rohim dan Ibu Syamsiah, selaku orang tua yang begitu penulis hormati dan banggakan.
2. Nenek dan kakak – kakak serta keponakan yang begitu penulis sayangi.
3. Prof. Dr. Abdul Haris, M.Ag selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang
4. Dr. Sri Harini, M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Malang
5. Dr. Cahyo Crisdian selaku Ketua Jurusan Teknik Informatika yang senantiasa memberikan dorongan semangat, arahan dan motivasi.

6. A'la Syauqi, M.Kom selaku Dosen Wali sekaligus pembimbing I, yang telah meluangkan waktu, membimbing dan mengarahkan penulis dalam penyusunan skripsi ini.
7. Supriyono, M. Kom selaku Dosen pembimbing II yang juga telah begitu banyak meluangkan waktu dan pemikirannya untuk membimbing penulis dalam menyelesaikan skripsi ini.
8. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan dengan begitu sabar selama masa studi.
9. Teman – teman seperjuangan teknik informatika dari berbagai angkatan yang tidak dapat saya sebutkan satu persatu.

“Tak Ada Gading yang Tak Retak” begitupun pepatah mengatakan begitu pula skripsi ini, penulis menyadari sepenuhnya bahwa dalam penulisan skripsi ini masih jauh dari kesempurnaan karena keterbatasan kemampuan yang penulis miliki. Dengan segenap kerendahan hati, penulis mengharapkan kritik dan saran yang membangun penulisan skripsi ini agar dapat memberikan manfaat khususnya bagi penulis sendiri dan umumnya kepada seluruh pembaca skripsi ini. Semoga skripsi ini berguna bagi perkembangan ilmu pengetahuan.

*Wassalamu 'alaikum Warahmatullahi Wabarakatuh.*

Malang, 19 Desember 2017

Hizbun Nasor

## DAFTAR ISI

<b>HALAMAN PENGAJUAN</b> .....	<b>ii</b>
<b>HALAMAN PERSETUJUAN</b> .....	<b>iii</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>iv</b>
<b>HALAMAN PERNYATAAN ORISINALITAS</b> .....	<b>v</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>vi</b>
<b>MOTTO</b> .....	<b>vii</b>
<b>KATA PENGANTAR</b> .....	<b>viii</b>
<b>DAFTAR ISI</b> .....	<b>x</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiii</b>
<b>DAFTAR TABEL</b> .....	<b>xvi</b>
<b>ABSTRAK</b> .....	<b>xvii</b>
<b>ABSTRACT</b> .....	<b>xviii</b>
<b>ملخص</b> .....	<b>xix</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah .....	3
1.4 Tujuan Penelitian .....	4
1.5 Manfaat Penelitian .....	4
1.6 Sistematika Penulisan .....	4
<b>BAB II KAJIAN PUSTAKA</b> .....	<b>5</b>
2.1 Penelitian Terkait .....	6

2.2 Landasan Teori.....	7
2.2.1 SOA ( <i>Service Oriented Architecture</i> ).....	7
2.2.2 <i>Web Service</i> .....	8
2.2.3 XML ( <i>eXtensible Markup Language</i> ).....	10
2.2.4 SOAP ( <i>Simple Object Access Protocol</i> ) .....	11
2.2.5 WSDL ( <i>Web Service Description Language</i> ).....	12
2.2.6 NuSOAP .....	12
2.2.7 Fingerspot Easy Touch SDK.....	12
2.3 Integrasi Sains dan Islam .....	13
<b>BAB III METODE PENELITIAN .....</b>	<b>17</b>
3.1 Tahap Analisis.....	17
3.1.1 Analisis Masalah.....	18
3.1.2 Kebutuhan Sistem .....	18
3.1.3 Analisis Sistem yang Sedang Berjalan .....	19
3.2 Desain Sistem .....	21
3.2.1 Rancang Model Arsitektur Sistem .....	21
3.2.2 Perancangan Sistem dengan SOAP WSDL.....	22
3.2.3 Rancang Pemodelan Sistem .....	27
3.2.4 Perancangan Antar Muka.....	47
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>51</b>
4.1 Kebutuhan Pengujian Sistem .....	51
4.2 Implementasi Sistem .....	52
4.2.1 Implementasi Basis Data.....	52
4.2.2 Implementasi <i>Web Service</i> .....	57

4.2.3 Implementasi <i>Web Client</i> .....	69
4.2.4 Implementasi Aplikasi Finger.....	77
4.3 Pengujian Sistem.....	79
<b>BAB V PENUTUP</b> .....	<b>85</b>
5.1 Kesimpulan .....	85
5.2 Saran.....	86
<b>DAFTAR PUSTAKA</b> .....	<b>87</b>



## DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>web service</i> .....	9
Gambar 3.1 Model pembuatan sistem .....	17
Gambar 3.2 <i>Flowchart</i> pendaftaran pasien yang sedang berjalan .....	19
Gambar 3.3 <i>Flowchart</i> pendaftaran pasien yang dikembangkan .....	20
Gambar 3.4 Model Arsitektur Sistem .....	22
Gambar 3.5 Proses <i>request</i> dan <i>response</i> data <i>web service</i> .....	23
Gambar 3.6 <i>Use Case Diagram</i> .....	28
Gambar 3.7 <i>Sequence diagram</i> login .....	30
Gambar 3.8 <i>Sequence diagram</i> registrasi citra sidik jari .....	30
Gambar 3.9 <i>Sequence diagram</i> verifikasi citra sidik jari .....	31
Gambar 3.10 <i>Sequence diagram</i> registrasi data pasien .....	32
Gambar 3.11 <i>Sequence diagram</i> verifikasi citra sidik jari .....	33
Gambar 3.12 <i>Sequence diagram</i> cari data pasien .....	34
Gambar 3.13 <i>Squence Diagram Update</i> Data Pasien .....	35
Gambar 3.14 <i>Sequence diagram delate</i> data pasien .....	36
Gambar 3.15 <i>Activity diagram login</i> .....	37
Gambar 3.16 <i>Activity diagram registrasi</i> citra sidik jari .....	38
Gambar 3.17 <i>Aktivty diagram verifikasi</i> cintra sidik jari .....	39
Gambar 3.18 <i>Activity diagram registrasi</i> data pasien .....	39
Gambar 3.19 <i>Activity diagram registrasi</i> data pasien .....	40
Gambar 3.20 <i>Activity diagram update</i> data pasien .....	41
Gambar 3.21 <i>Activity diagram delate</i> data pasien .....	41
Gambar 3.22 <i>Activity diagram cari</i> data pasien .....	42

Gambar 3.23 <i>Class Diagram</i> .....	43
Gambar 3.24 Halaman login admin .....	47
Gambar 3.25 <i>Interface</i> halaman registrasi citra sidik jari .....	48
Gambar 3.26 <i>Interface</i> halaman verifikasi citra sidik jari .....	48
Gambar 3.27 <i>Interface</i> halaman registrasi data pasien .....	49
Gambar 3.28 <i>Interface</i> halaman list data .....	49
Gambar 3.29 <i>Interface</i> halaman verifikasi data pasien .....	50
Gambar 3.30 <i>Interface</i> halaman <i>update</i> data pasien .....	50
Gambar 4.1 Tampilan halaman kelola pada 000webhost .....	57
Gambar 4.2 Tampilan Filezilla .....	58
Gambar 4.3 Tampilan hasil WSDL <i>function create</i> .....	59
Gambar 4.4 <i>Source code register function create</i> .....	60
Gambar 4.5 Tampilan hasil WSDL <i>function readbyid</i> .....	61
Gambar 4.6 <i>Source code register readbyid</i> .....	61
Gambar 4.7 Tampilan hasil WSDL <i>function readbyfinger</i> .....	62
Gambar 4.8 <i>Source code register readbyfinger</i> .....	63
Gambar 4.9 Tampilan hasil WSDL <i>function readbycari</i> .....	63
Gambar 4.10 <i>Source code register readbycari</i> .....	64
Gambar 4.11 Tampilan hasil WSDL <i>function updatebyid</i> .....	65
Gambar 4.12 <i>Source code register updatebyid</i> .....	66
Gambar 4.13 Tampilan hasil WSDL <i>function delatebyid</i> .....	67
Gambar 4.14 <i>Source code register delatebyid</i> .....	68
Gambar 4.15 Tampilan hasil WSDL <i>function readall</i> .....	68
Gambar 4.16 <i>Source code register delatebyid</i> .....	69

Gambar 4.17 <i>listing code</i> koneksi ke <i>database</i> fingerprint .....	69
Gambar 4.18 <i>listing code</i> config pada <i>web client</i> .....	70
Gambar 4.19 <i>listing code</i> <i>autoload</i> pada <i>web client</i> .....	70
Gambar 4.20 <i>listing code</i> <i>ceck user</i> pada <i>controler admin</i> .....	70
Gambar 4.21 Halaman login admin .....	71
Gambar 4.22 <i>listing query</i> <i>select convert</i> FingerData .....	71
Gambar 4.23 <i>Listing code</i> <i>call function</i> <i>create</i> .....	72
Gambar 4.24 Halaman registrasi pasien baru .....	73
Gambar 4.25 Halaman verifikasi data pasien .....	74
Gambar 4.26 <i>Listing code</i> <i>call function</i> <i>readbyfinger</i> .....	74
Gambar 4.27 <i>Listing code</i> <i>call function</i> <i>readall</i> .....	74
Gambar 4.28 <i>Listing code</i> <i>call function</i> <i>readbyid</i> .....	75
Gambar 4.29 <i>Listing code</i> <i>call function</i> <i>updatebyid</i> .....	76
Gambar 4.30 <i>Listing code</i> <i>call function</i> <i>deletebyid</i> .....	76
Gambar 4.31 Halaman list data .....	76
Gambar 4.32 Halaman edit data pasien .....	76
Gambar 4.33 <i>listing code</i> <i>load</i> citra untuk proses registrasi .....	77
Gambar 4.34 <i>Listing code</i> <i>load</i> citra untuk proses verifikasi .....	78
Gambar 4.35 <i>Interface</i> registrasi citra sidik jari .....	78
Gambar 4.36 <i>Interface</i> verifikasi citra sidik jari .....	79

## DAFTAR TABEL

Tabel 3.1 Daftar layanan pada <i>web service</i> .....	24
Tabel 3.2 Definisi actor .....	27
Tabel 3.3 Tabel users .....	44
Tabel 3.4 Tabel fingerprint_t .....	44
Tabel 3.5 Tabel webservice .....	44
Tabel 3.6 Tabel wilayah_provinsi .....	45
Tabel 3.7 Tabel wilayah_kabupaten .....	45
Tabel 3.8 Tabel wilayah_kecamatan .....	45
Tabel 3.9 Tabel wilayah_desa .....	45
Tabel 3.10 Tabel t_pasien .....	46
Tabel 4.1 Prosedur pengujian sistem .....	79
Tabel 4.2 Tabel pengujian fungsionalitas sistem .....	80
Tabel 4.3 Tabel pengujian antarmuka .....	81
Tabel 4.4 Hasil pengamatan pada sistem yang berjalan .....	83
Tabel 4.5 Hasil simulasi sistem yang dikembangkan .....	83

## ABSTRAK

**Nasor, Hizbun. 2018. Rancang Bangun Sistem Pelayanan Pendaftaran Pasien Rumah Sakit Berbasis *Web Service* dengan Protokol SOAP WSDL dan Bantuan Alat Deteksi Sidik Jari.** Skripsi. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) A'la Syauqi, M.Kom dan (II) Supriyono, M.Kom

---

Kata Kunci: Rumah Sakit, *Web Service*, SOAP WSDL

Rumah sakit merupakan institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna dan menyediakan pelayanan rawat inap, rawat jalan, dan gawat darurat. Namun masih terdapat beberapa permasalahan yang dihadapi pada sistem pelayanan rumah sakit, salah satunya adalah pada sistem pendaftaran pasien.

Berdasarkan permasalahan diatas penulis menggunakan teknologi *web service* sebagai pemecahan masalah. Dalam pengembangan sistem rumah sakit ini menggunakan beberapa tahapan yaitu analisis sistem, desain sistem, implementasi *web service* dengan menggunakan SOAP WSDL hingga pengujian sistem. Desain sistem dimodelkan dengan menggunakan UML (*Unified Modeling Language*) dan bahasa yang digunakan adalah PHP dengan *framework Codeigniter*.

Hasil dari pengujian fungsionalitas sistem yang dilakukan diperoleh 46.25% responden sangat setuju, 47.5% responden setuju, dan 6.25% responden tidak setuju, adapun dari hasil pengujian antarmuka diperoleh 15% responden sangat setuju, 70% responden setuju dan 15% menyatakan tidak setuju. Pengujian efektivitas sistem menunjukkan bahwa sistem yang dikembangkan lebih cepat 00.22 menit saat pendaftaran pasien baru dan 00.07 menit saat pendaftaran pasien lama.

## ABSTRACT

**Nasor, Hizbun. 2018. Design for Hospital-Based Patient Registration Service System with WSDL SOAP Protocol and Fingerprint Detection Tool Assistance.** Thesis. Department of Informatics, Faculty of Science and Technology, Islamic State University Maulana Malik Ibrahim Malang.

Supervisor: (I) A'la Syauqi, M.Kom and (II) Supriyono, M. Kom

---

Keywords: Hospital, Web Service, SOAP WSDL

Hospital is a health service institution that provides full range of personal health services and provides inpatient, outpatient and emergency care services. But there are still some problems faced in the hospital service system, such as the patient registration system.

Based on the above problems the researcher use web service technology as the problem solver. In the development of this hospital system, it uses several stages of system analysis, system design, web service implementation using SOAP WSDL for the system testing. System design is modeled using UML (Unified Modeling Language) and the language used is PHP with Codeignaiter framework.

The result of the test of system functionality is 46.25% of respondents strongly agree, 47.5% of respondents agree, and 6.25% of respondents disagree, while 15% of respondents strongly agree, 70% of respondents agree and 15% disagree. System effectiveness test shows that the system is developing faster on 00.22 minutes when a new patient registration and 00.07 minutes during registration for the old patient.

## خلاصة

ناصر، حزبون. 2018-خدمة تسجيل بصمات الأصابع على أساس "بنية النظام للمريض المستشفى خدمة ويب" مع WSDL والصابون البروتوكولات تساعد أداة الكشف. أطروحة. قسم هندسة المعلومات، كلية العلوم والتكنولوجيا، المؤسف أوينين مولانا إبراهيم مالك.

المشرف: (1) على سياووي، ماجستير و (2) سوبريونو، ماجستير

الكلمات الرئيسية: المستشفى، خدمات ويب الصابون، WSDL

مستشفى هو مؤسسة تقوم الصحة الخدمات الصحية خدمات الأفراد في الجلسات العامة وتزويد المستشفيات والعيادات الخارجية والطوارئ. ولكن لا تزال هناك بعض المشاكل التي تواجه في نظام الخدمة للمستشفيات، واحد منهم في نظام تسجيل المرضى. استناداً إلى المشكلة المذكورة أعلاه والكاتب يستخدم التكنولوجيا لخدمة ويب كحل مؤقت. في تطوير استخدامات نظام المستشفى عدة مراحل هي: تحليل النظم، تصميم النظم، تنفيذ خدمة ويب باستخدام WSDL الصابون حتى نظام الاختبار. تصميم النظام على غرار استخدام UML (لغة النمذجة الموحدة) واللغة المستخدمة هي بي إتش بي مع إطار كوديجنيتير. نتائج اختبار الأداء الوظيفي للنظام القيام بالحصول على 46.25% في المائة من المجيبين بشدة الاتفاق، وافق 47.5% في المائة المجيبين، واختلف 6.25% في المائة من المجيبين، ولكن اتفق واجهة الحصول على 15% في المائة من المجيبين بشدة من نتائج الاختبار، توافق على 70% في المائة المجيبين وأما وقد اختلف 15%. اختبار فعالية النظام يشير إلى أن النظام الذي وضعه 00.22 أسرع ثانية عندما الجديد المريض المريض التسجيل و 00.07 دقيقة طويلة.

# **BAB I**

## **PENDAHULUAN**

Pada bab pendahuluan ini akan dijelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

### **1.1 LATAR BELAKANG**

Pelayanan pendaftaran pasien merupakan bagian yang sangat penting dalam kaitannya dengan sarana pelayanan kesehatan. Proses pendaftaran pasien merupakan kontak pertama kali antara petugas rumah sakit dengan pasien atau keluarganya. Oleh karena itu, baik buruknya pelayanan pendaftaran pasien dapat menjadi salah satu acuan penilaian bagi masyarakat dalam menilai kualitas pelayanan kesehatan pada sebuah rumah sakit.

Namun masih terdapat permasalahan pada pelayanan pendaftaran pasien yang berjalan saat ini di beberapa rumah sakit atau klinik, sebagaimana hasil penelitian yang dilakukan oleh (Sabarguna : 2004) yang mengatakan bahwa keluhan yang sering terdengar dari pihak pemakai pelayanan kesehatan yang biasanya menjadi sasaran adalah sikap petugas administrasi, sarana yang kurang memadai, kelambatan pelayanan, persediaan obat, tarif pelayanan, dan lain – lain. Hal ini diperkuat oleh penelitian yang dilakukan oleh (Dewanto, Aryo: 2015) yang menunjukkan bahwa kepuasan masyarakat terhadap infrastruktur dan prosedur administrasi rumah sakit dari empat rumah sakit yang ada di kota Batu belum sampai pada nilai memuaskan atau baik.

Salah satu indikator dalam upaya peningkatan kualitas pelayanan kesehatan di rumah sakit adalah pelayanan pendaftaran pasien, sebagai mana yang tercantum di dalam Kepmenkes No.129/Menkes/SK/II/2008 tentang standar pelayanan minimal rumah sakit, yang di dalamnya disebutkan bahwa pelayanan minimal pendaftaran cepat jika waktu tunggu kurang dari 10 menit dan disebut lama jika waktu tunggu lebih dari 10 menit.

Lamanya pelayanan pendaftar pasien rawat jalan selain karena jumlah pasien yang cukup banyak, juga dikarenakan penggunaan *patient card* sebagai bukti telah berobat yang tidak jarang pasien lupa untuk membawanya saat berobat atau rusak dan hilang sehingga tidak jarang *patient card* harus dibuat kembali dan menyebabkan proses pelayanan pendaftaran pasien relatif lebih lama. Selain itu juga pengolahan data yang belum terdistribusi sehingga adanya kemungkinan pasien yang telah berobat di salah satu rumah sakit harus membuat *patient card* lagi saat berobat di rumah sakit yang lain. Hal ini mengakibatkan kurangnya efektifitas dan efisiensi pada proses pelayanan pendaftaran pasien.

Berdasarkan beberapa uraian dari permasalahan diatas, perlu di kembangkan sebuah metode pendaftaran pasien secara *biometric* dengan menggunakan *tool Fingerspot Easy Touch SDK* sebagai teknologi pengenalan sidik jari, karena kecepatan pengenalan sidik jari yang dapat dilakukan sampai 200 sidik jari perdetik dan juga mendukung penggunaan DSN, OLE DB dan ODBC sehingga dapat digunakan pada berbagai macam *database*. Selain itu perlu juga digunakan teknologi *web service* dengan menggunakan spesifikasi protokol SOAP WSDL untuk membuat sistem terdistribusi, karena SOAP WSDL dirancang untuk menangani lingkungan komputasi terdistribusi dan merupakan standar yang berlaku

untuk *web service* sehingga memiliki dukungan dan *tools* dari berbagai vendor, serta memiliki sifat yang ekstensibel yang memungkinkan dalam menggunakan banyak ekstensi (Nurdiyanto, Wisnu:2012).

## 1.2 RUMUSAN MASALAH

Berdasarkan pemaparan pada latar belakang di atas, dapat diambil sebuah rumusan masalah diantaranya:

1. Bagaimana perancangan sistem pendaftaran pasien berbasis *biometric* terhadap peningkatan pelayanan pendaftaran pasien rawat jalan.
2. Bagaimana perancangan distribusi data dengan teknologi *web service* dalam meningkatkan efektifitas pelayanan rumah sakit pada bagian pendaftaran pasien rawat jalan.

## 1.3 BATASAN MASALAH

Terdapat beberapa batasan masalah yang dibuat dalam penelitian ini, adapun batasan – batasan tersebut diantaranya:

1. Penelitian yang dilakukan berdasarkan mekanisme pendaftaran pada rumah sakit RS dr. Etty Asharto.
2. Penelitian yang dilakukan pada proses pendaftaran pasien rawat jalan.
3. Proses bisnis hanya sampai pada proses pendaftaran data pasien belum sampai pada pengolahan data rekam medis.
4. Data yang diolah hanyalah data pribadi pasien.

#### 1.4 TUJUAN PENELITIAN

Adapun tujuan dari penelitian ini diantaranya:

1. Merancang sistem pendaftaran pasien berbasis *biometric* untuk peningkatan pelayanan pendaftaran pasien rawat jalan.
2. Merancang sistem distribusi data dengan teknologi *web service* untuk meningkatkan efektifitas pelayanan rumah sakit pada bagian pendaftaran pasien rawat jalan.

#### 1.5 MANFAAT PENELITIAN

Adapun manfaat dari penelitian ini diantaranya:

1. Dapat menekan biaya cetak dokumen dan ATK,
2. Dapat mempercepat proses persiapan data dan transaksi,
3. Dapat mengurangi resiko kehilangan dan kesalahan data,
4. Dapat meningkatkan pendapatan yang disebabkan oleh meningkatnya kepercayaan pelanggan,
5. Dapat meningkatkan layanan eksternal dari segi kepuasan pelanggan,
6. Dapat meningkatkan image yang disebabkan oleh meningkatnya mutu layanan pada bagian pendaftaran pasien rawat jalan,
7. Meningkatkan kualitas layanan pada bagian pendaftaran pasien rawat jalan.

#### 1.6 SISTEMATIKA PENULISAN

##### **Bab I Pendahuluan**

Pada bab ini diuraikan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penulisan, manfaat penulisan, serta sistematika penulisan.

## **Bab II Landasan Teori**

Pada bab ini dibahas penelitian terkait, landasan teori dan integrasi sains dan islam.

## **Bab III Metode Penelitian**

Pada bab ini dipaparkan metode-metode penelitian yang di gunakan penulis untuk menemukan dan menyelesaikan permasalahan yang ada.

## **Bab IV Hasil dan Pembahasan**

Pada bab ini di paparkan tentang hasil dan pembahasan dari aplikasi yang telah dibuat, dari awal menjalankan aplikasi sampai dengan tahap uji coba.

## **Bab VI Penutup**

Pada bab ini berisi kesimpulan-kesimpulan yang diperoleh dari penelitian yang telah dibuat dan juga berisi saran-saran untuk pengembangan aplikasi berikutnya.

## BAB II

### KAJIAN PUSTAKA

Pada bab kajian pustaka akan dijelaskan tentang penelitian terkait, landasan teori yang meliputi (*service oriented architecture, web service, extensible markup language, simple object access protocol, web service description language, NuSOAP, dan fingerprint easy touch SDK*), serta integrasi sains dan islam.

#### 2.1 Penelitian Terkait

Terdapat begitu banyak penelitian – penelitian yang terkait dengan *web service* yang telah dilakukan. Seperti halnya yang dilakukan oleh Arif Adi dan Riyanto (2013) yang melakukan penelitian tentang pemanfaatan web service sebagai integrasi data farmasi di RSUD Banyumas. Hasil dari penelitian ini adalah terbangunnya web service menggunakan *metode simple obyek acces protocol* untuk memberikan solusi integrasi data farmasi.

Trio Suryaning Adinoto (2013) Melakukan penelitian tentang perancangan absensi karyawan smp negeri 1 kramat tegal. Penelitian ini dilakukan untuk memudahkan proses input absensi karyawan dengan menggunakan mesin fingerprint. Hasil dari penelitian ini berupa sistem informasi yang dapat diakses secara online oleh pegawai.

Martinus Raditia Sigit Surendra (2014) melakukan penelitian tentang impementasi PHP web service sebagai penyedia data aplikasi mobile, penelitian ini difokuskan pada sisi web service server dengan menggunakan teknologi JSON. Hasil dari penelitian ini adalah web service server yang dapat menyediakan

layanan data secara dinamis seiring dengan bertambahnya informasi tempat maupun informasi promo.

Erick Kurniawan (2014) melakukan penelitian tentang implementasi rest web service untuk sales order dan sales tracking berbasis mobile. Penelitian ini dilakukan untuk membuat sistem yang dapat memantau aktifitas dan mempercepat proses pemesanan produk dengan menggunakan aplikasi berbasis mobile. Aplikasi mobile yang dibuat pada penelitian ini memanfaatkan data dari GPS untuk memastikan lokasi tenaga penjual, selain itu juga memiliki fasilitas untuk membaca barcode barang menggunakan kamera untuk mempercepat input data barang. Untuk memanipulasi data digunakan teknologi REST Service.

Adi Irawan, Indah Fitri Astuti, M. Cs, dan Dedy Cahyadi, M. Eng (2015) melakukan penelitian tentang presensi sidik jari (fingerprint) berbasis web service. Web service pada penelitian ini digunakan untuk membantu mahasiswa FMIPA UNMUL yang ingin mengetahui kehadiran dosen dikampus setiap harinya.

## **2.2 Landasan Teori**

### **2.2.1 SOA (Service Oriented Architecture)**

SOA merupakan sebuah arsitektur kerangka kerja berbasis standar terbuka yang memungkinkan perusahaan – perusahaan untuk saling mengintegrasikan data satu sama lain dengan layanan yang diberikan.

Menurut Rahmi Nur Shofa, *et al.*(2013) SOA sendiri merupakan suatu konsep gaya arsitektur yang memodularisasi sistem informasi menjadi *service*,

dengan demikian *web service* merupakan teknologi yang tepat untuk menerapkan konsep SOA, karena *cycle* proses *web service* sama dengan SOA.

SOA menyediakan sebuah layanan yang terdapat pada *web service*, dimana layanan – layanan tersebut dibagi kepada pengguna aplikasi berdasarkan karakteristik SOA. Adapun karakteristik – karakteristik tersebut adalah:

1. Service yang dipanggil oleh *service* lainnya tanpa program pemanggil perlu memperhatikan dimana lokasi *service* yang dipanggil berada sehingga pemanggilan *service* dapat dilakukan secara *run-time*. Karakteristik ini biasa disebut sebagai *Loose Couple*.
2. SOA tersusun dari *service interface* dan *service implementation*.
3. *Service* harus *business oriented*, sehingga setiap *service* yang disediakan harus melakukan aktifitas bisnis tertentu.

### 2.2.2 *Web Service*

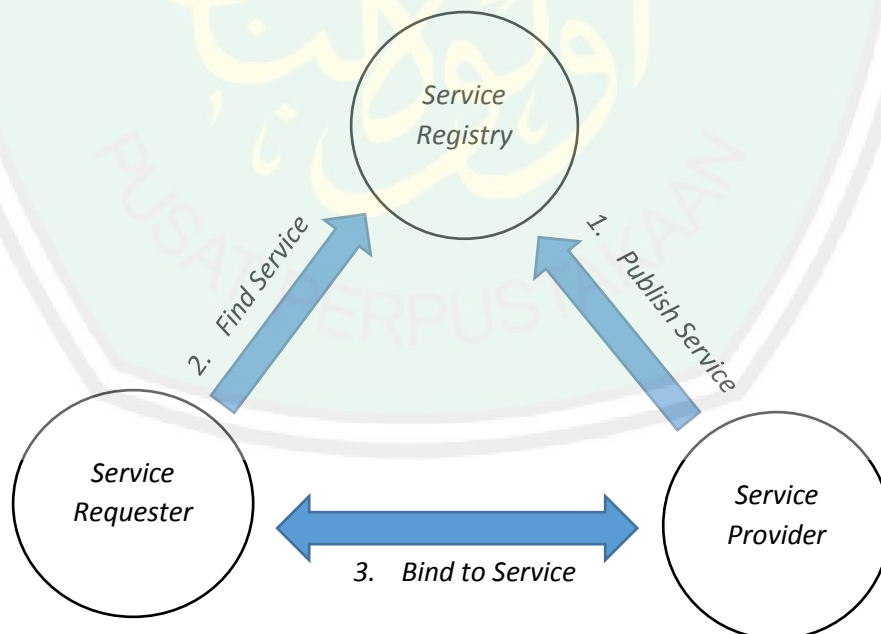
*Web service* merupakan suatu komponen *software* yang dirancang untuk mendukung interaksi antar sistem pada suatu jaringan, *web service* digunakan sebagai sebuah fasilitas pada *web site* untuk menyediakan layanan dalam bentuk informasi kepada sistem lain. Sehingga sistem lain dapat berkomunikasi dengan sistem tersebut melalui *service* yang telah disediakan.

“Utomo Budiyanto dan Khabib Mustofa (2014), *web service* adalah sebuah antar muka yang terletak diantara kode aplikasi dan pengguna kode tersebut, berperan sebagai lapisan abstrak yang memisahkan *platform* dan rincian spesifik bahasa pemrograman tentang bagaimana kode aplikasi sebenarnya dipanggil”.

Berbeda dengan aplikasi web konvensional, *web service* tidak mengembalikan *interface* yang dapat langsung digunakan oleh *user* berupa layanan yang berbentuk kode HTML, tetapi *web service* mengembalikan layanan tersebut berupa kode XML yang berorientasi pada data yang langsung dapat digunakan oleh *user*.

Keunggulan teknologi *web service* dapat menjadi suatu solusi yang tepat bagi pengembangan sebuah sistem informasi, karena penggunaan *internet* yang sudah merasuk kesemua bidang termasuk dunia bisnis dan industri, serta *web service* dapat menjembatani sebuah aplikasi yang berbeda *framework* sekalipun berbeda dalam pemilihan database yang digunakan.

Operasi *web service* digambarkan dengan sebuah arsitektur yang memiliki tiga komponen dan peranan yang berbeda.



Gambar 2.1 Arsitektur *web service*

Keterangan gambar 2.1

1. *Service Provider* berfungsi sebagai lokasi sentral yang mendeskripsikan semua layanan *layanan service* yang sudah di *register*.
2. *Service Registri* merupakan peminta layanan yang mencari dan menemukan layanan yang dibutuhkan oleh *service requestor*, serta menggunakan layanan tersebut.
3. *Service Requestor* merupakan pihak yang membutuhkan sebuah layanan atau *service* dan mengolah sebuah *registry* agar layanan-layanan tersebut dapat tersedia.

### 2.2.3 XML (*eXtensible Markup Language*)

XML merupakan sekumpulan aturan – aturan yang mendefinisikan suatu sintak yang digunakan untuk menjelaskan suatu teks atau data dalam sebuah dokument dengan penggunaan sebuah tag.

Pada dasarnya XML tidak jauh berbeda dengan HTML, dimana elemen yang digunakan sama – sama menggunakan pembuka dan penutup, masing – masing dikembangkan untuk tujuan yang berbeda – beda. HTML digunakan sebagai pemberi informasi yang terbuka sedangkan XML mendeskripsikan susunan informasi dan berfokus pada informasi itu sendiri.

Di dalam *web service* XML berfungsi sebagai komunikasi antar aplikasi dan integrasi data walaupun antar aplikasi yang berkomunikasi tersebut berbeda *platform*. Didalam sebuah jurnal Implementasi *Extensible Markup Language Web Service* (Soetam Rizky Wicaksono, 2008), XML *web service* memiliki beberapa karakteristik diantaranya:

1. Standar dari XML *web service* adalah standar industri yang independen.

2. XML *web service* tidak menyediakan *interface*, tetapi hanya menyediakan fungsi.
3. XML *web service* menggunakan *Hyper Text Transfer Protocol (HTTP)* sebagai protokol standar.
4. XML *web service* menggunakan model *request* dan *response* yang sama seperti pada aplikasi web.
5. XML *web service* ditempatkan didalam sebuah *web server*.
6. XML *web service* lebih ditujukan untuk kepentingan dari *programmer to programmer*.
7. Hasil dari XML *web service* merupakan *plain text* yang berupa berkas XML.
8. Fungsi dari sebuah XML *web service* butuh pengolahan lebih lanjut dalam membentuk sebuah *interface* aplikasi.
9. Secara umum hampir semua bahasa pemrograman yang dapat terkoneksi ke internet akan mampu terkoneksi dengan XML *web service*. Namun tidak semua bahasa pemrograman tersebut dapat memproduksi XML *web service*.

#### 2.2.4 SOAP (*Simple Object Access Protocol*)

SOAP merupakan protokol untuk pertukaran informasi dengan desentralisasi dan terdistribusi. SOAP merupakan gabungan antara HTTP dengan XML, karena SOAP umumnya menggunakan protokol HTTP sebagai sarana *transport* datanya dan data akan di tulis dalam format XML.

Karena SOAP menggunakan HTTP dan XML maka SOAP memungkinkan pihak – pihak yang mempunyai *platform*, sistem informasi dan perangkat lunak yang berbeda dapat saling mempertukarkan datanya. SOAP

mengatur bagaimana *request* dan *response* dari suatu *web service* bekerja (Hartati Devina, 2011).

### 2.2.5 WSDL (*Web Service Description Language*)

WSDL merupakan bahasa berbasis XML yang digunakan untuk mendefinisikan *web service* dan menggambarkan bagaimana cara untuk mengakses *web service* tersebut.

Ketika aplikasi *client* meminta *service*, UDDI akan memberikan informasi tentang letak dari dokumen WSDL, WSDL berisi sebuah pesan dengan skema XML. Elemen – elemen yang ada pada WSDL adalah: *Message*, *operation*, *port type*, *binding*, *port*.

### 2.2.6 NuSOAP

NuSOAP merupakan sebuah *library open source toolkit* dibawah lesensi GNU LGPL yang didistribusikan oleh NuSphere Corporation yang berisikan *class-class* PHP yang memungkinkan *user* untuk mengirim dan menerima pesan SOAP.

### 2.2.7 Fingerspot Easy Touch SDK

Fingerspot Easy Touch SDK merupakan cara termudah untuk menambahkan fitur pengenalan sidik jari ke aplikasi dengan menggunakan sensor U.are.U dari Digital Persona. SDK ini dirancang sesederhana mungkin sehingga dapat dengan mudah untuk dipelajari dan diaplikasikan kedalam sebuah aplikasi.

## 2.3 Integrasi Sains dan Islam

Islam selalu dapat menyentuh semua aspek kehidupan, tidak terkecuali aspek kesehatan seperti halnya yang dijelaskan pada tafsir Hidayatul Insan surat Shaad ayat 29. Allah SWT berfirman:

كُتِبَ أَنْزَلْنَاهُ إِلَيْكَ مُبْرَكٌ لِيَدَّبَّرُوا آيَاتِهِ وَلِيَتَذَكَّرَ أُولُوا الْأَلْبَابِ ﴿٢٩﴾

*Artinya: Ini adalah sebuah Kitab yang kami turunkan kepadamu penuh dengan berkah supaya mereka memperhatikan ayat-ayatnya dan supaya mendapat pelajaran orang-orang yang mempunyai pikiran.*

Pada tafsir Hidayatul Insan dijelaskan bahwa maksud dari kata مُبْرَكٌ adalah didalam Al-Qur'an terdapat kebaikan dan ilmu yang banyak, terdapat petunjuk dari kesehatan, terdapat obat dari penyakit, cahaya sebagai penerang di tengah kegelapan, dan terdapat hukum yang dibutuhkan oleh manusia. Didalamnya terdapat dalil yang paling qath'i untuk semua tuntutan agama, dimana kitab tersebut merupakan kitab paling agung yang datang kealam semesta. Begitu pula pada tafsir Ibnu Katsir yang menjelaskan bahwa ayat ini diturunkan dikarenakan Al-Qur'an memberikan arahan kepada tujuan – tujuan yang benar dan sumber – sumber rasional yang tepat bagi orang – orang yang berakal.

Allah SWT juga menunjukkan sempurnanya perhatian dan kelembutannya kepada hamba – hambanya dengan diciptakannya sebuah minuman yang keluar dari perut lebah (madu) sebagai obat dari berbagai macam penyakit. Allah SWT berfirman di dalam Surat An-Nahl ayat 69.

ثُمَّ كُلِي مِنْ كُلِّ الثَّمَرَاتِ فَاسْلُكِي سُبُلَ رَبِّكِ ذُلَالًا ۗ يَخْرُجُ مِنْ بُطُونِهَا شَرَابٌ مُخْتَلِفٌ أَلْوَانُهُ فِيهِ شِفَاءٌ لِلنَّاسِ ۗ إِنَّ فِي ذَٰلِكَ لَآيَةً لِّقَوْمٍ يَتَفَكَّرُونَ ﴿٦٩﴾

*Artinya: Kemudian makanlah dari tiap-tiap (macam) buah-buahan dan tempuhlah jalan Tuhanmu yang Telah dimudahkan (bagimu). dari perut lebah itu ke luar minuman (madu) yang bermacam-macam warnanya, di dalamnya terdapat obat yang menyembuhkan bagi manusia. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda (kebesaran Tuhan) bagi orang-orang yang memikirkan.*

Tafsir Hidayatul Insan menjelaskan bahwa kata *فِيهِ شِفَاءٌ لِلنَّاسِ* pada ayat diatas merupakan sebuah petunjuk kesempurnaan dan kelembutan Allah Ta'ala kepada hambanya. Oleh karena itu, tidak ada yang berhak diberi kecintaan dan ibadah selain Dia. Pada ilmu kedokteran Nabi *فِيهِ شِفَاءٌ لِلنَّاسِ* berarti bahwa madu itu bisa dipergunakan untuk obat penyakit kedinginan, karena madu bersifat panas. Seperti yang dijelaskan pada tafsir Ibnu Katsir.

Dalam kitab Shahih al-Bukhori disebutkan dari Ibnu Abbas, dimana dia bercerita, Rasulullah SAW bersabda:

*الشِّفَاءُ فِي ثَلَاثَةِ شَرَبَةِ عَسَلٍ وَشَرْطَةِ مِحْجَمٍ وَكَيْتَةِ نَارٍ وَأَنْهَى أُمَّتِي عَنِ الْكَيِّ*

*Artinya: Terapi pengobatan itu ada tiga cara, yaitu minum madu, bekam dan kay (menempelkan besi panas pada daerah yang terluka), sedangkan aku melarang ummatku berobat dengan kay.*

Didalam islam juga menjunjung tinggi kaidah - kaidah kedokteran yang berjalan diatas tiga perkara, diantaranya:

1. Menjaga kesehatan dari segala sesuatu yang membahayakan,
2. Menjaga diri dari bahaya, dan
3. Menghilangkan bahaya.

Kaidah pertama dan kedua terdapat pada adanya perintah makan dan minum serta tidak berlebih – lebihan dan Allah SWT juga memperbolehkan berbuka puasa bagi musafir dan orang sakit untuk menjaga kesehatannya.

Sebagaimana firman Allah pada surat Al-A'raf ayat 31 dan surat Al – Baqarah ayat 185.

﴿ يَبْنِيْٓ اٰدَمَ خُذُوْا زِيْنَتَكُمْ عِنْدَ كُلِّ مَسْجِدٍ وَكُلُوْا وَاشْرَبُوْا وَلَا تُسْرِفُوْا ۗ اِنَّهٗ لَا يُحِبُّ الْمُسْرِفِيْنَ ۗ ﴾

Artinya: Hai anak Adam, pakailah pakaianmu yang indah di setiap (memasuki) mesjid, makan dan minumlah, dan janganlah berlebih-lebihan. Sesungguhnya Allah tidak menyukai orang-orang yang berlebih-lebihan.

شَهْرُ رَمَضَانَ الَّذِي اُنزِلَ فِيْهِ الْقُرْاٰنُ هُدًى لِّلنَّاسِ وَبَيِّنٰتٍ مِّنَ الْهُدٰى وَالْفُرْقٰنِ ۗ فَمَنْ شَهِدَ مِنْكُمُ الشَّهْرَ فَلْيَصُمْهُ ۗ وَمَنْ كَانَ مَرِيْضًا اَوْ عَلٰى سَفَرٍ فَعِدَّةٌ مِّنْ اَيَّامٍ اٰخَرَ ۗ يُرِيْدُ اللّٰهُ بِكُمْ الْيُسْرَ وَلَا يُرِيْدُ بِكُمُ الْعُسْرَ وَلِتُكْمِلُوْا الْعِدَّةَ وَلِتُكَبِّرُوْا اللّٰهَ عَلٰى مَا هَدٰنَكُمْ وَلَعَلَّكُمْ تَشْكُرُوْنَ ۗ

Artinya: (beberapa hari yang ditentukan itu ialah) bulan Ramadhan, bulan yang di dalamnya diturunkan (permulaan) Al Quran sebagai petunjuk bagi manusia dan penjelasan-penjelasan mengenai petunjuk itu dan pembeda (antara yang hak dan yang bathil). Karena itu, barangsiapa di antara kamu hadir (di negeri tempat tinggalnya) di bulan itu, Maka hendaklah ia berpuasa pada bulan itu, dan barangsiapa sakit atau dalam perjalanan (lalu ia berbuka), Maka (wajiblah baginya berpuasa), sebanyak hari yang ditinggalkannya itu, pada hari-hari yang lain. Allah menghendaki kemudahan bagimu, dan tidak menghendaki kesukaran bagimu. dan hendaklah kamu mencukupkan bilangannya dan hendaklah kamu mengagungkan Allah atas petunjuk-Nya yang diberikan kepadamu, supaya kamu bersyukur.

Kaidah ketiga dapat dimisalkan dengan diperbolehkannya bagi orang yang sedang melakukan ihram, dimana kepalanya terganggu oleh kutu dan sebagainya, sebagaimana sabda Rasulullah SAW.

كَأَنَّ هَوَامَّ رَأْسِكَ تُؤْذِيكَ فَقُلْتُ أَجَلَ قَالَ فَاحْلِفْهُ وَأَذْبَحْ شَاةً أَوْ صُمْ ثَلَاثَةَ أَيَّامٍ أَوْ تَصَدَّقْ بِثَلَاثَةِ  
أَصْعٍ مِنْ تَمْرٍ بَيْنَ سِنَّتَيْ مَسَاكِينٍ

Artinya: “Sepertinya kutu pada rambut kepalamu telah melukaimu.” Saya menjawab, “Benar” Beliau lalu bersabda: “Cukurlah rambutmu. Kemudian sembelihlah seekor kambing, atau kamu berpuasa tiga hari, atau bersedekah sebanyak tiga sha' kurma untuk dibagikan kepada enam orang miskin”. (HR. Muslim, Ahmad dan Abu Daud).



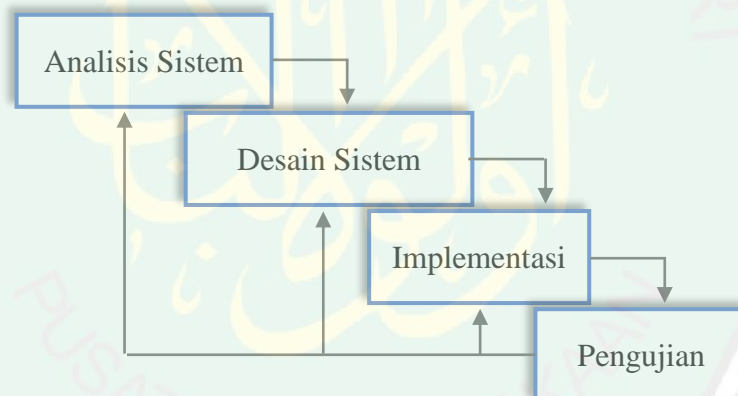
## BAB III

### METODELOGI PENELITIAN

Pada bab metodologi penelitian ini akan dijelaskan tentang tahap analisis yang meliputi (analisis masalah, dan analisis kebutuhan), serta desain sistem yang meliputi (rancang model arsitektur sistem, perancangan sistem dengan SOAP WSDL, rancang pemodelan sistem, dan perancangan antarmuka).

#### 3.1 Metode Pengembangan Perangkat Lunak

Metode pengembangan perangkat lunak yang digunakan pada perancangan sistem ini terdiri dari beberapa tahapan seperti yang di gambarkan pada gambar 3.1.



Gambar 3.1 Metode pengembangan sistem

#### 3.2 Tahap Analisis

Fase ini dilakukan untuk menguraikan sistem menjadi komponen - komponen dengan tujuan untuk mengetahui seberapa bagus komponen - komponen tersebut bekerja. Adapun bagian - bagian analisis yang dilakukan diantaranya:

1. Analisis masalah
2. Analisis kebutuhan sistem.
3. Analisis sitem yang sedang berjalan

### 3.2.1 Analisis Masalah

Di tengah –tengah pertengahan perkembangan teknologi yang begitu pesat hampir belum ada rumah sakit yang bisa terintegrasi antar rumah sakit dan antara rumah sakit dengan klinik, selain itu belum ada terobosan untuk mengganti *Patient Card* dengan teknologi *biometric* berbasis fingerprint yang telah berkembang saat ini. Sehingga perlu adanya sebuah terobosan baru untuk memudahkan manajemen dan pertukaran data. Hal ini memungkinkan untuk digunakannya teknologi *web service* dan *Fingerspot™ Easy Touch SDK* untuk menyelesaikan permasalahan yang dihadapi.

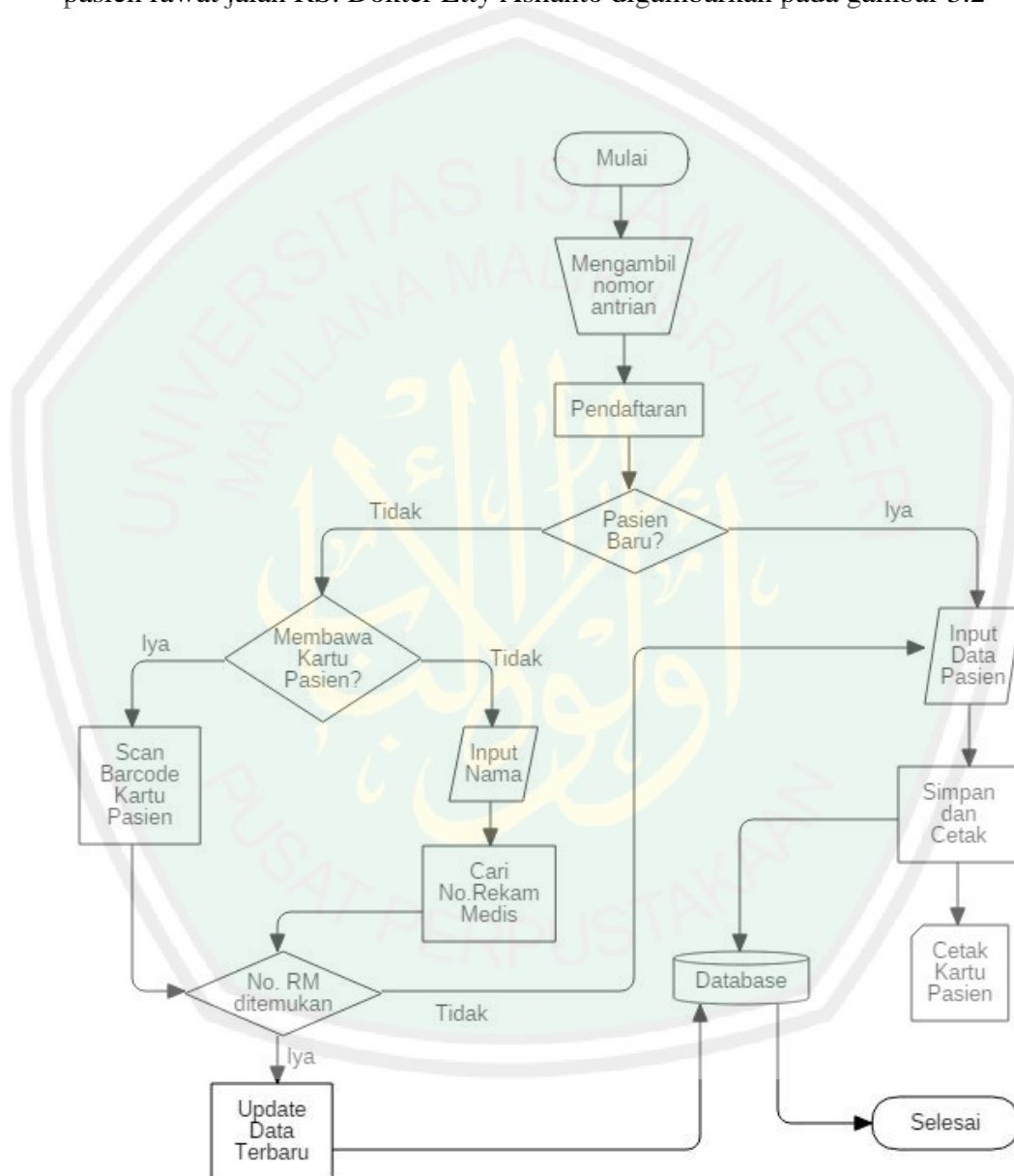
### 3.2.2 Kebutuhan Sistem

Kebutuhan perangkat lunak yang digunakan dalam perancangan sistem ini terdiri dari beberapa jenis aplikasi, diantaranya:

1. *Server: Apache Server 3.2.2*
2. *DBMS: MySQL*
3. *OS Windows 7 64 bit*
4. *Development Tools: Notepad ++, PHP, Visual Basic 6.0, Fingerspot Easy Touch SDK*

### 3.2.3 Analisis Sistem yang Sedang Berjalan

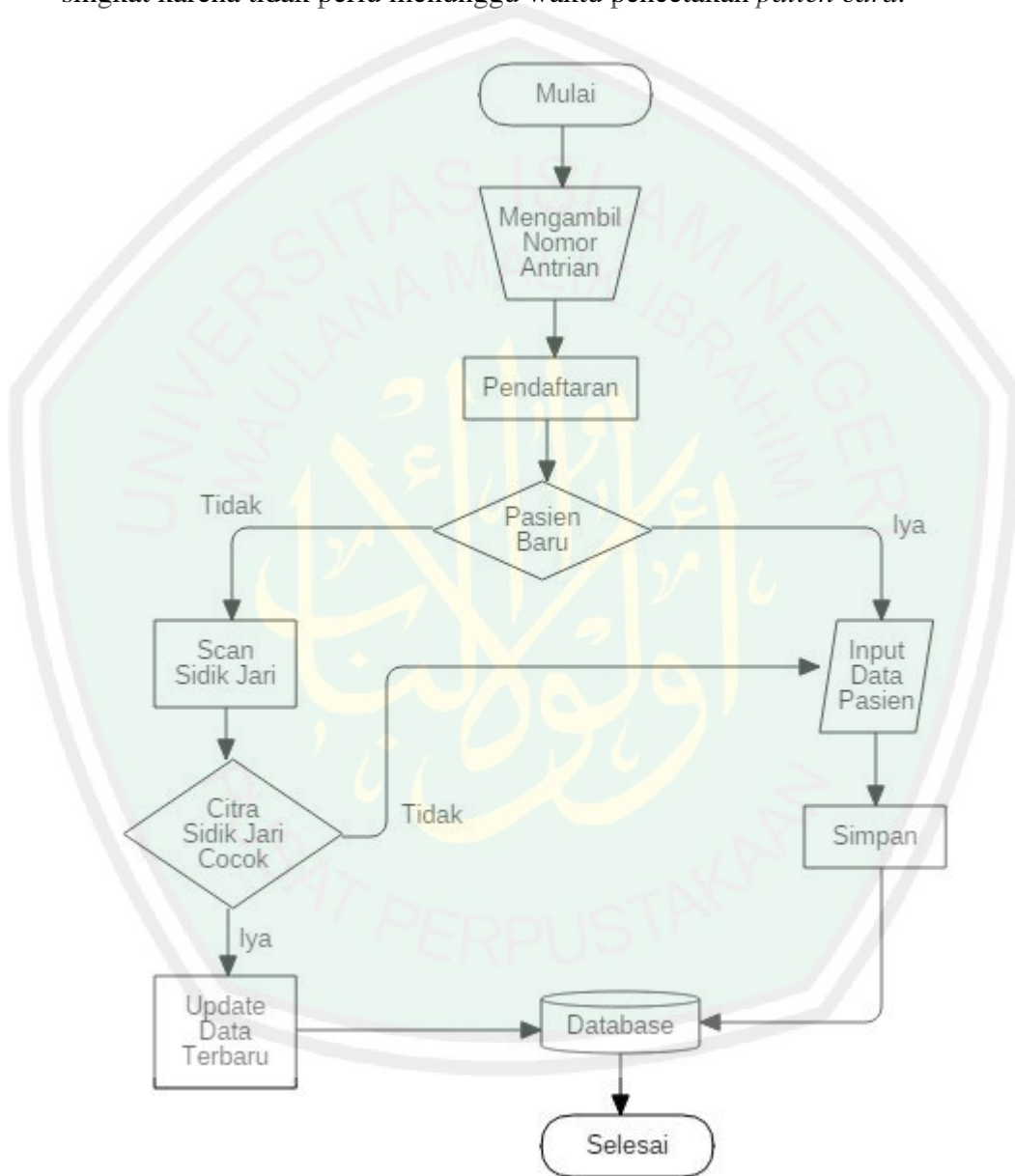
Hasil analisis sistem yang sedang berjalan pada sistem pendaftaran pasien rawat jalan RS. Dokter Etty Ashanto digambarkan pada gambar 3.2



Gambar 3.2 *Flowchart* pendaftaran pasien rawat jalan yang sedang berjalan

Dari hasil analisis yang dilakukan pada sistem yang sedang berjalan saat ini, maka dapat terlihat perbedaan yang jelas dengan sistem yang sedang di kembangkan, seperti yang terlihat pada gambar 3.3. Pada sistem yang sedang

dikembangkan pasien tidak perlu membawa atau khawatir dengan kartu pasien yang tertinggal, rusak atau hilang karena digunakannya sistem *biometric* sebagai pengganti *patient card*. Selain itu proses pendaftaran pasien barupun akan lebih singkat karena tidak perlu menunggu waktu pencetakan *patient card*.



Gambar 3.3 Flowchart pendaftaran pasien rawat jalan yang dikembangkan

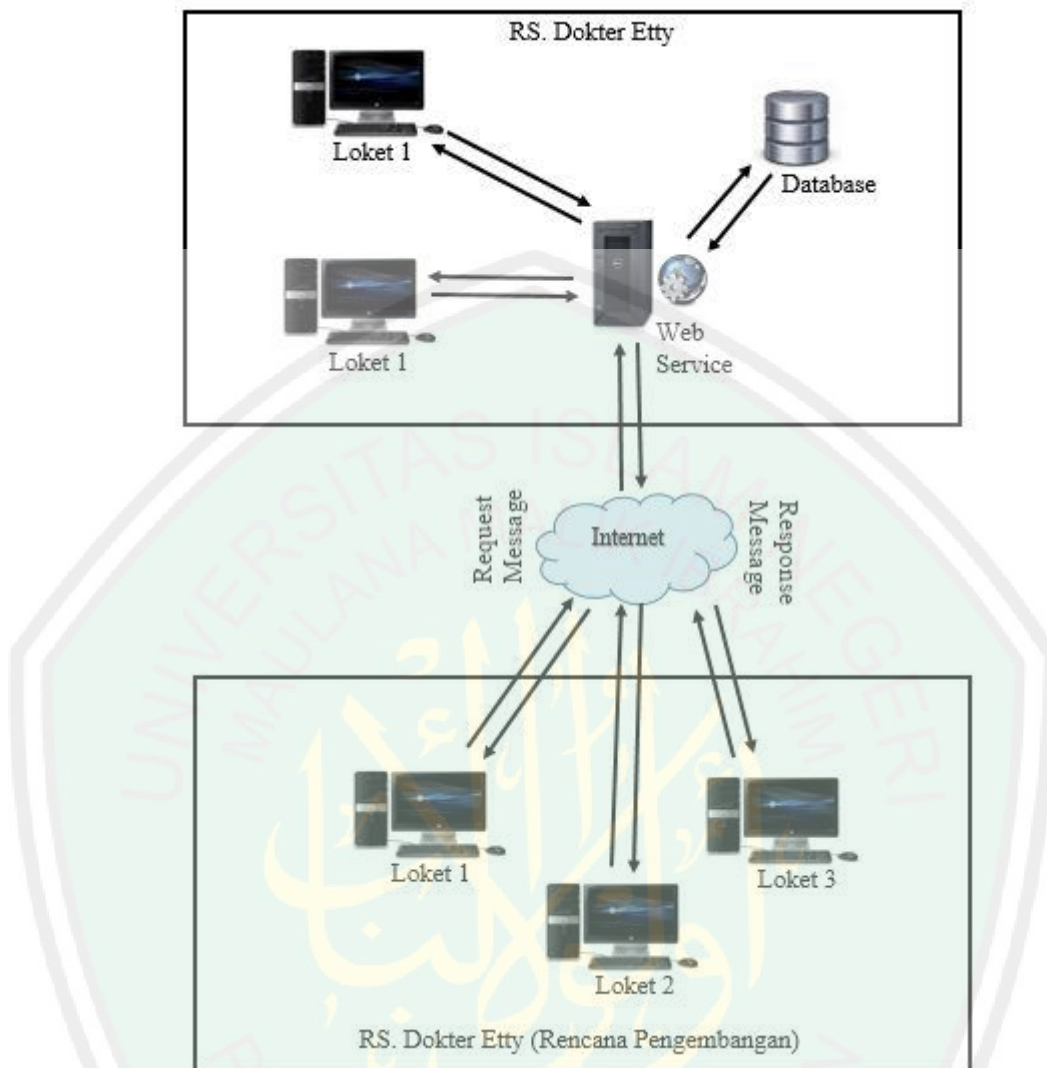
### 3.3 Desain Sistem

Desain sistem dilakukan untuk merancang atau mendesain system yang akan dibuat, menganalisis sejauh mana proses *host to host* rancang bangun system dengan mengaplikasikan metode SOAP WSDL yang digunakan dapat berjalan dengan baik.

Perancangan system yang dilakukan meliputi perancangan arsitektur dan pemodelan serta menjelaskan komponen – komponen yang digunakan dalam pembangunan system seperti *Entity Relationship Diagram*, *Database Design*, *Use Case*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram* dan antar muka system.

#### 3.3.1 Rancangan Model Arsitektur Sistem

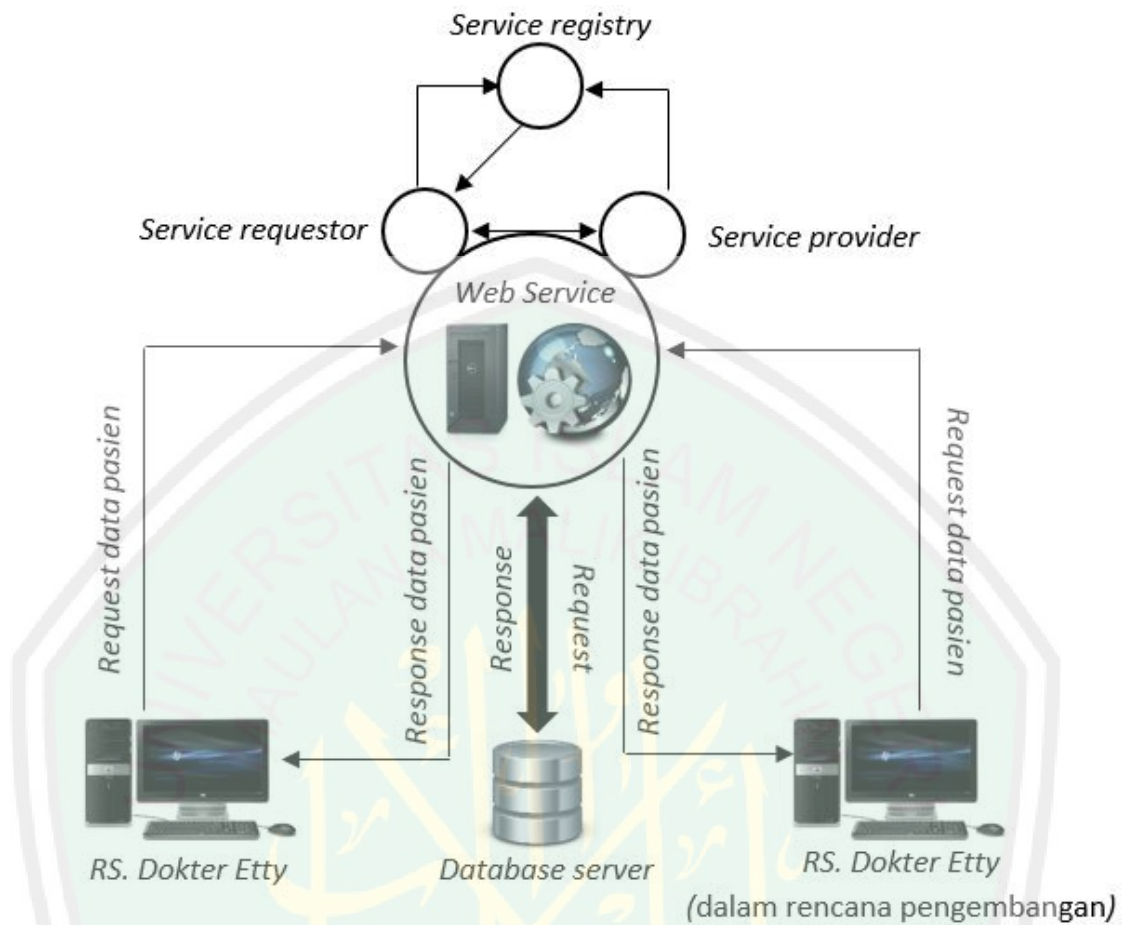
Rancang model arsitektur sistem merupakan rancangan bentuk komunikasi antar client dengan server dalam sebuah *web service*. Model arsitektur ini dirancang sebagai rancangan umum bagaimana antar aplikasi berkomunikasi. Berikut arsitektur system *web service* pendaftaran pasien. Lihat gambar 3.4



Gambar 3.4 Model Arsitektur Sistem

### 3.3.2 Perancangan Sistem dengan SOAP WSDL

Perancangan system dengan SOAP WSDL dilakukan untuk menggambarkan bagaimana prosedur pengiriman data pasien dari *client* kepada *database server* dan juga daftar layanan yang diberikan oleh *web service* serta apa saja yang menjadi parameter input dan output.



Gambar 3.5 Proses *request* dan *response* data *web service*

Penyimpanan data pada perancangan sistem pendaftaran pasien ini dilakukan dengan sistem terpusat pada *database server* seperti halnya yang digambarkan pada gambar 3.3. Sehingga masing – masing *aplikasi client* hanya melakukan penyimpanan data sidik jari, sedangkan data identitas pasien tersimpan secara terpusat pada *database server*.

Dari gambar 3.3 dijelaskan bahwa aplikasi client melakukan *request* terhadap aplikasi *web service*, kemudian *web service* menerima *service requestor* yang kemudian diproses dengan melakukan pengecekan terhadap *service registry* yang tersedia sebelum akhirnya dilakukan komunikasi dengan *database server*.

Berikut merupakan tabel daftar *service* atau layanan yang akan digunakan dalam perancangan *web service* pendaftaran pasien.

Table 3.1 Daftar layanan pada *web service*

LAYANAN	PARAMETER	OUTPUT	KETERANGAN
function create()	\$dataFingerString, \$Nama_P, \$Jenis_Kel, \$Tgl_Lhr, \$Tgl_dft, \$Agama, \$PekerjaanP, \$PendidikanP, \$Status, \$Nama_A, \$Nama_I, \$Nama_SI, \$Pekerjaan_SI, \$Pendidikan_SI, \$Pekerjaan_A, \$Pekerjaan_I, \$Pendidikan_A, \$Pendidikan_I, \$Alamat, \$RtRw, \$Kelurahan, \$Kecamatan, \$Kota, \$Provinsi, \$Negara, \$Nomor_P, \$Kel_P, \$Nama_Peru, \$No_Tlp,	Return (String)	Service yang digunakan untuk melakukan input data pasien baru.

	\$No_Hp, \$No_Fax, \$Email		
function readbyid()	\$ID	Return (Array)	Service yang digunakan untuk melakukan pencarian berdasarkan ID.
function readbycari()	\$Nama_P	Return (Array)	Service yang digunakan untuk melakukan pencarian data pasien berdasarkan nama pasien.
function readbyfinger()	\$dataFingerString	Return (Array)	Service yang digunakan untuk melakukan validasi data pasien yang pernah berobat.
function readall()	-	Return (Array)	Service yang digunakan untuk menampilkan

			semua data pasien secara real time.
function updatebyid()	\$ID, \$Nama_P, \$Jenis_Kel, \$Tgl_Lhr, \$Tgl_dft, \$Agama, \$PekerjaanP, \$PendidikanP, \$Status, \$Nama_A, \$Nama_I, \$Nama_SI, \$Pekerjaan_SI, \$Pendidikan_SI, \$Pekerjaan_A, \$Pekerjaan_I, \$Pendidikan_A, \$Pendidikan_I, \$Alamat, \$RtRw, \$Kelurahan, \$Kecamatan, \$Kota, \$Provinsi, \$Negara, \$Nomor_P, \$Kel_P, \$Nama_Peru, \$No_Tlp, \$No_Hp, \$No_Fax, \$Email	Return (String)	Service yang digunakan untuk melakukan update data pasien berdasarkan ID.

function deletebyid()	\$ID	Return (String)	Service yang digunakan untuk melakukan penghapusan data pasien.
--------------------------	------	--------------------	---

### 3.3.3 Rancangan Pemodelan Sistem

Pemodelan sistem dilakukan untuk menggambarkan sebuah komponen, elemen dan notasi dapat berkomunikasi dan memberikan informasi satu sama lain. Dalam pemodelan system ini ada beberapa diagram yang dirancang yang dirancang untuk menggambarkan system pendaftaran pasien, diantaranya *use case diagram*, *sequence diagram*, *activity diagram*, dan *class diagram*.

#### 1. Use Case Diagram

*Use case diagram* dibuat untuk menggambarkan bagaimana proses-proses bisnis dalam lingkungan system pendaftaran pasien yang dirancang. Berikut merupakan gambaran yang terdapat pada system pendafaran pasien.

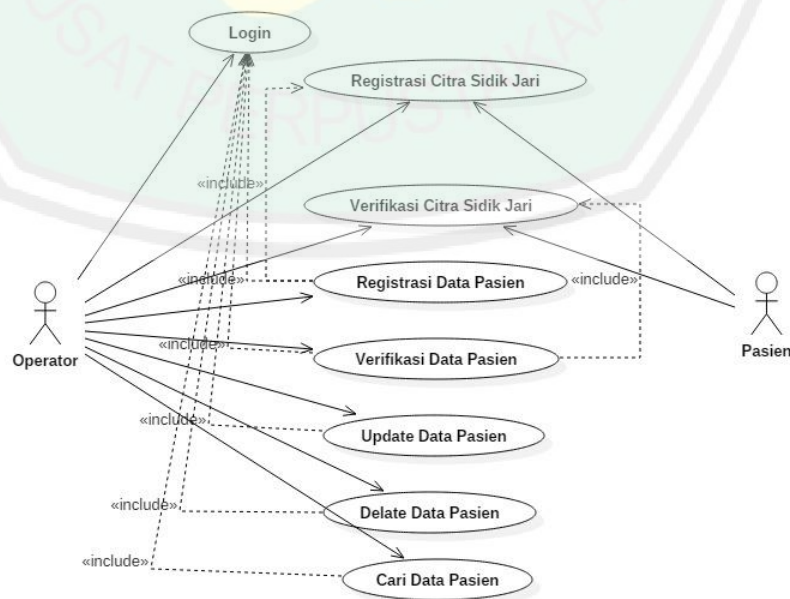
Table 3.2 Definisi actor

No	Actor	Keterangan
1.	Pasien	Merupakan sebuah actor yang melakukan pendaftaran pasien dengan memberikan data citra sidik cari beserta data diri.
2.	Operator	Merupakan actor yang memiliki hak akses untuk melakukan penginputan data pasien, pencarian data pasien, validasi dan pengolahan data.

Penjelasan:

- a. Dalam melakukan proses pendaftaran pasien terlebih dulu memasukkan citra sidik jari melalui fingerprint dan kemudian memberikan data diri yang diperlukan. Sedangkan untuk melakukan proses verifikasi yaitu dengan pasien memasukkan citra sidik jari yang sebelumnya digunakan dalam proses registrasi untuk dilakukan proses pencocokan dengan data yang tersimpan.
- b. Pada proses pendaftaran pasien baru operator terlebih dulu menjalankan aplikasi Scane Sidik Jari untuk mengambil citra sidik jari pasien dan mengakses *website local* untuk menginputkan data pasien baru, sedangkan untuk proses verifikasi operator melakukan pencarian data berdasarkan data sidik jari yang telah dilakukan pencocokan dengan data sidik jari yang tersimpan sebelumnya

Untuk lebih jelasnya mengenai *use case diagram* yang dibuat dapat dilihat pada gambar..



Gambar 3.6 Use Case Diagram

## 2. *Sequence Diagram*

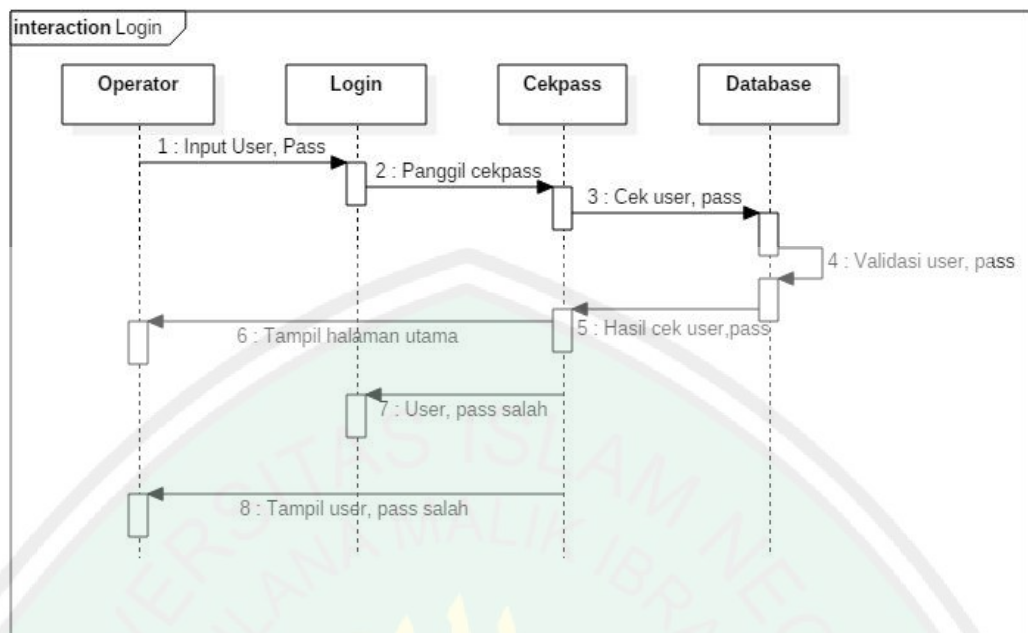
*Sequence diagram* digunakan untuk menggambarkan interaksi antar objek disekitar sistem berdasarkan urutan proses.

### a. *Sequence Diagram Login*

*Sequence diagram login* merupakan sebuah bentuk pemodelan interaksi untuk menggambarkan ketika *user login* kedalam system.

Urutan proses login sebagai berikut:

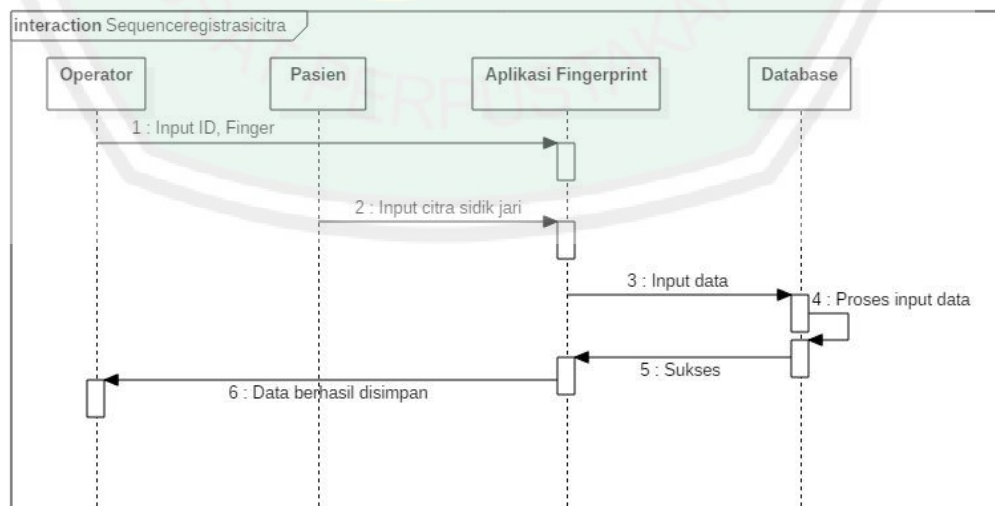
- a) Operator menginputkan data *user* dan *password* agar dapat masuk kedalam system pendaftaran pasien.
- b) Data *user* dan *password* yang diinputkan akan dieksekusi pada kelas cekpass, dari kelas tersebut akan dicek kedalam database akan kebenaran data yang telah diinputkan.
- c) System akan memvalidasi hasil *login* lalu memberikan hasil *login* kedalam kelas cekpass, dan kemudian hasil tersebut akan diteruskan kepada operator.
- d) Jika data *user* dan *password* valid, maka akan ditampilkan halaman utama, namun jika tidak valid maka akan dimunculkan pesan kesalahan.



Gambar 3.7 *Sequence diagram* login

b. *Sequence Diagram* Registrasi Citra Sidik Jari

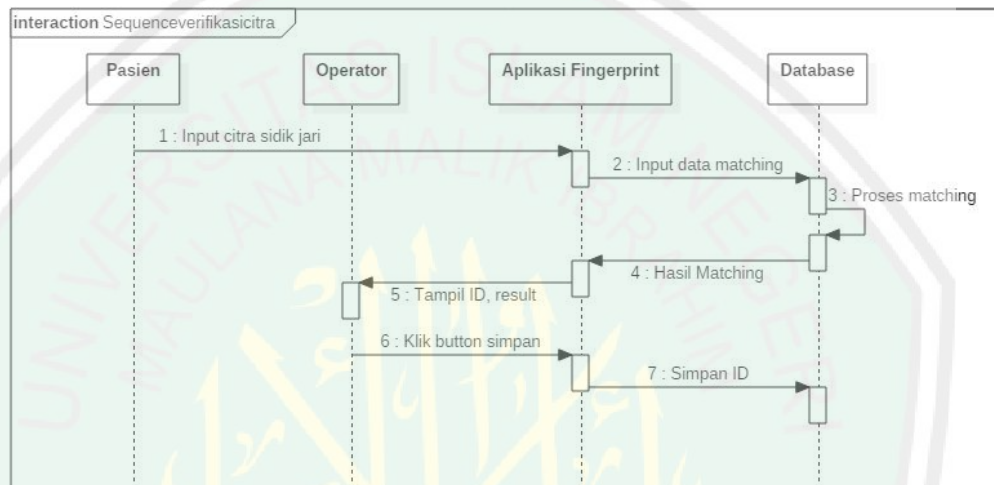
*Sequence diagram* login merupakan sebuah bentuk pemodelan interaksi untuk menggambarkan proses pengambilan citra sidik jari pasien baru. Urutan proses registrasi citra sidik jari adalah sebagai berikut:



Gambar 3.8 *Sequence diagram* registrasi citra sidik jari

c. *Sequence Diagram* Verifikasi Citra Sidik Jari

*Sequence diagram* verifikasi citra sidik jari merupakan sebuah bentuk pemodelan interaksi untuk menggambarkan ketika pasien akan melakukan verifikasi data diri yang telah terdaftar sebelumnya. Urutan proses login sebagai berikut:



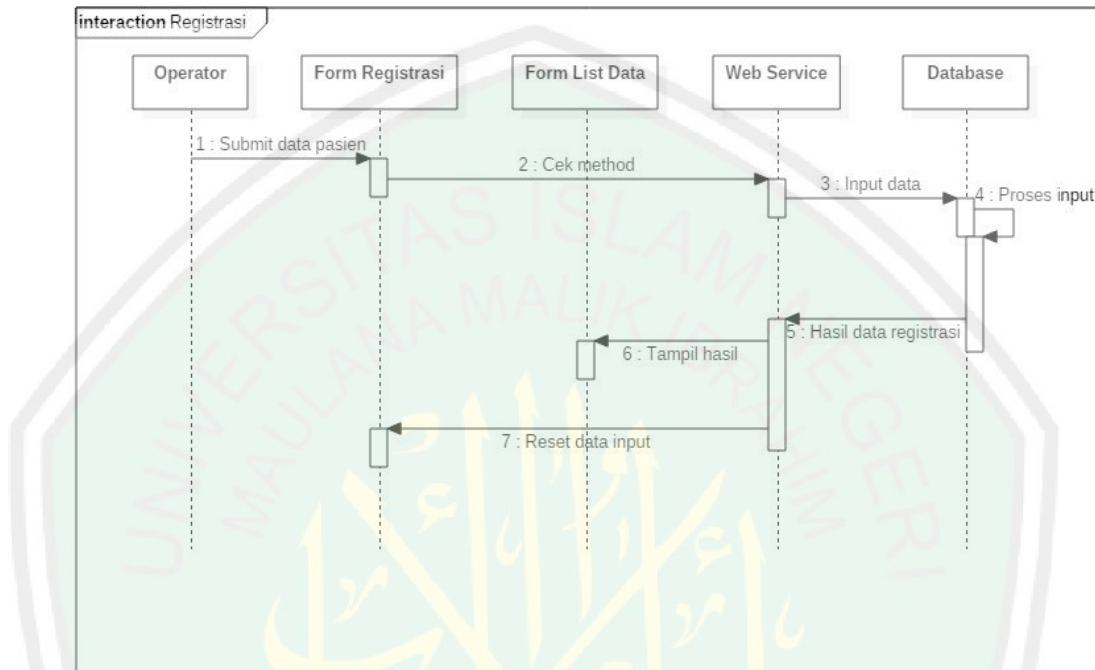
Gambar 3.9 *Sequence diagram* verifikasi citra sidik jari

d. *Sequence Diagram* Registrasi Data Pasien

*Sequence diagram* registrasi merupakan sebuah model interaksi yang digunakan ketika operator melakukan registrasi pasien baru melalui *web service*. Adapun bentuk pemodelan dari *sequence diagram* registrasi sebagai berikut.

- a) Operator melakukan input data pada halaman registrasi, dan kemudian menekan tombol simpan.
- b) Data yang diinputkan akan diteruskan kedalam *web service* untuk kemudian dilakukan pengecekan arah *method* yang digunakan.
- c) Hasil dari *web service* akan diinputkan kedalam database server.

- d) Hasil proses yang telah dilakukan database akan diteruskan kedalam *web service* yang kemudian akan menampilkan data pada form list data dan memberikan respon sukses.



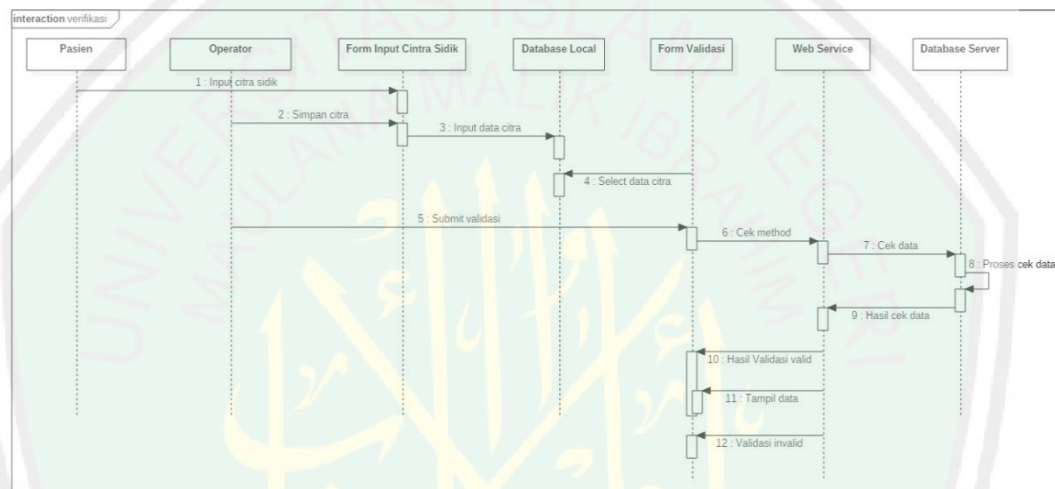
Gambar 3.10 *Sequence diagram* registrasi data pasien

e. *Sequence Diagram* Verifikasi Data Pasien

*Sequence diagram* verifikasi merupakan sebuah bentuk pemodelan interaksi yang terjadi ketika operator melakukan validasi data pasien yang telah berobat sebelumnya. Adapun urutan dari proses validasi adalah sebagai berikut.

- a) Pasien menginputkan data citra sidik jari melalui fingerprint.
- b) Operator menginputkan ID dan index jari yang digunakan oleh pasien.
- c) Nilai ID, indek sidik jari dan data citra sidik jari diinputkan kedalam database local.
- d) Form validasi melakukan select data kedalam database local.

- e) Operator menginputkan data validasi dengan menekan tombol validasi.
- f) Data validasi akan diteruskan kedalam *web service* untuk dicek arah *method* yang digunakan. *Web service* kemudian akan memvalidasi data dari database untuk kemudian diteruskan kedalam form verifikasi.



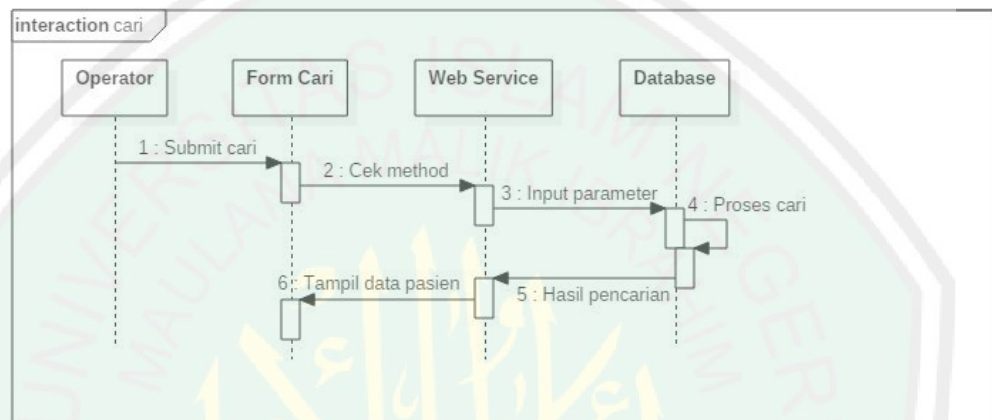
Gambar 3.11 *Sequence diagram* verifikasi citra sidik jari

f. *Sequence Diagram* Cari Data Pasien

*Sequence diagram* cari merupakan sebuah bentuk pemodelan interaksi yang terjadi ketika operator melakukan pencarian data pasien yang telah berobat sebelumnya. Adapun urutan dari proses pencarian data pasien adalah sebagai berikut.

- a) Operator menginputkan nama pasien sebagai parameter pencarian dengan menekan tombol *Search*.
- b) Parameter input akan diteruskan kedalam *web service* untuk dilakukan pengecekan *method* yang digunakan.

- c) Setelah *method* yang digunakan telah di temukan, *web service* melakukan proses pencarian kedalam *database* berdasarkan *query* yang ada pada *method* tersebut.
- d) Ketika data telah ditemukan maka *web service* mengirimkan data hasil pencarian untuk selanjutnya ditampilkan di dalam form.

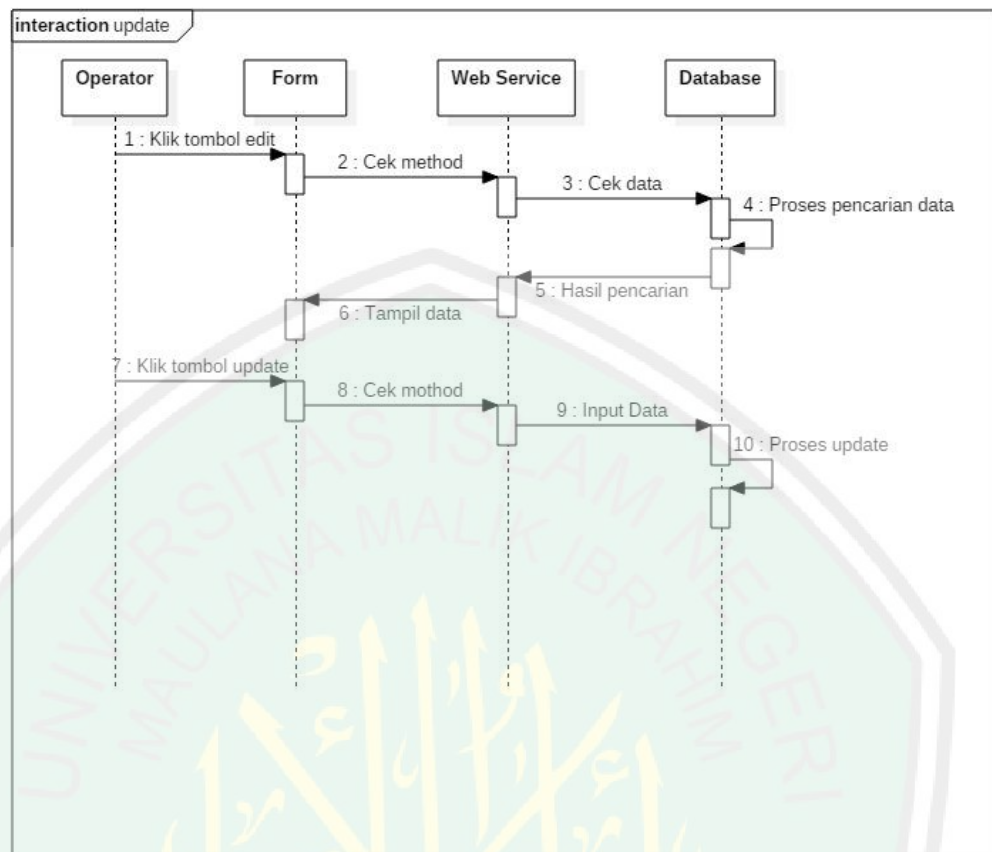


Gambar 3.12 *Sequence diagram* cari data pasien

g. *Sequence Diagram Update* Data Pasien

*Sequence diagram update* merupakan sebuah bentuk pemodelan interaksi yang terjadi ketika operator melakukan *update* data pasien yang telah berobat sebelumnya. Adapun urutan dari proses *update* data pasien adalah sebagai berikut.

- a) Operator mengklik tombol edit.
- b) *Web service* melakukan pencarian data berdasarkan *method* dan parameter input yang diterima.
- c) Setelah data ditemukan, data ditampilkan kedalam form *update*.
- d) Setelah data pasien selesai dirubah, operator melakukan klik tombol *update* untuk selanjutnya diproses oleh *web service* dengan menggunakan *function updatebyid*.

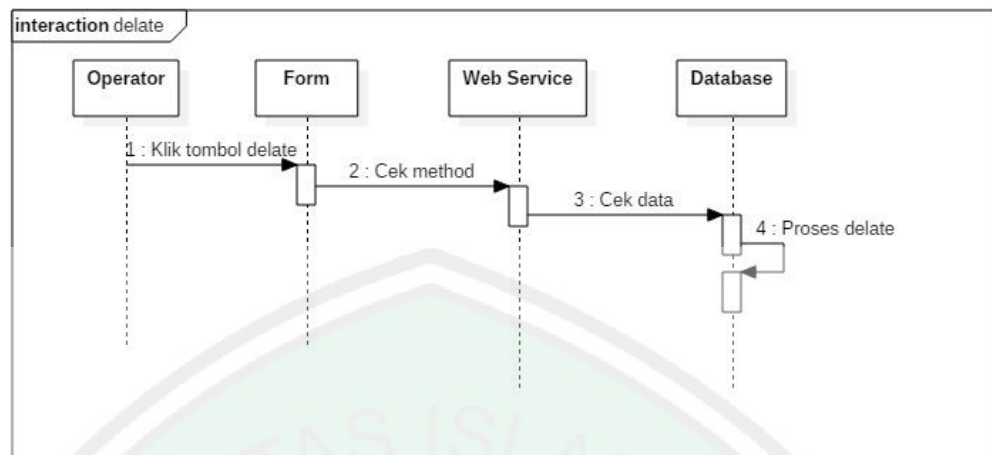


Gambar 3.13 *Sequence Diagram Update Data Pasien*

#### h. *Sequence Diagram Delate Data Pasien*

*Sequence diagram delate* merupakan sebuah bentuk pemodelan interaksi yang terjadi ketika operator melakukan penghapusan terhadap data pasien yang telah berobat sebelumnya. Adapun urutan dari proses penghapusan data pasien adalah sebagai berikut.

- a) Operator mengklik tombol *delate*.
- b) *Web service* mengecek *method* dan parameter yang diterima.
- c) Data pasien yang sesuai dengan parameter ID dihapus dari *database server*.



Gambar 3.14 *Sequence diagram delete data pasien*

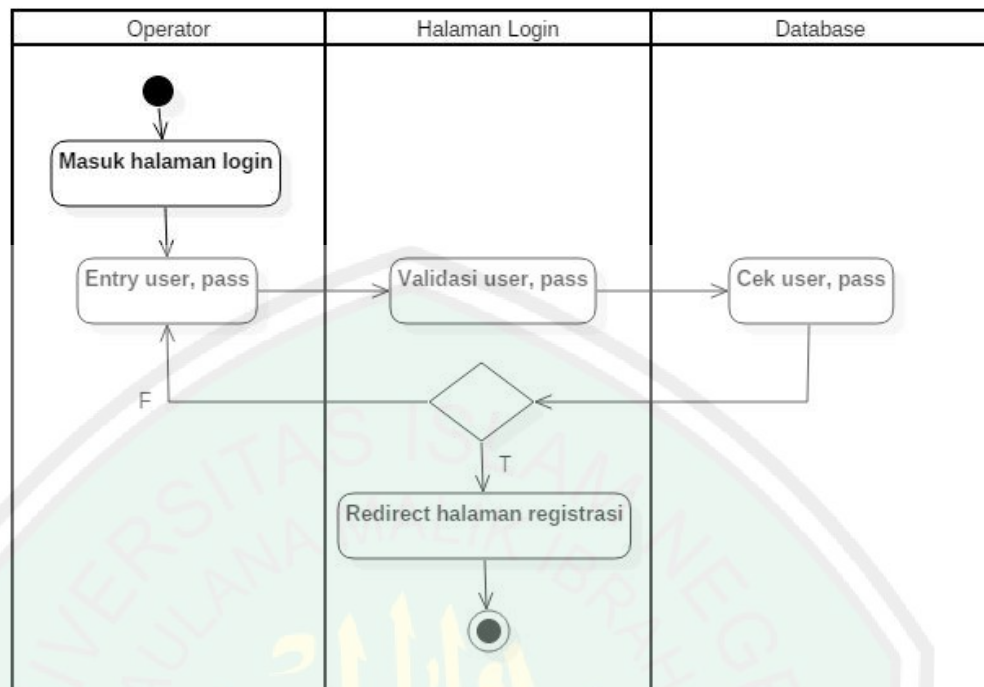
### 3. Activity Diagram

*Activity Diagram* digunakan sebagai pemodelan untuk menjelaskan bagaimana proses bisnis, logika prosedural, dan jalur kerja pada system yang dibuat.

Untuk proses bisnis pendaftaran pasien diawali dengan calon pasien mengambil nomor antrian, kemudian operator memanggil nomor antrian yang kemudian dilanjutkan dengan pengambilan data sidik jari menggunakan fingerprint, jika pasien telah melakukan pendaftaran sebelumnya maka akan dilakukan proses validasi, namun jika pasien belum pernah mendaftar sebelumnya maka akan dilakukan proses registrasi dengan mengisi data pribadi calon pasien baru.

#### a. *Activity Diagram Login*

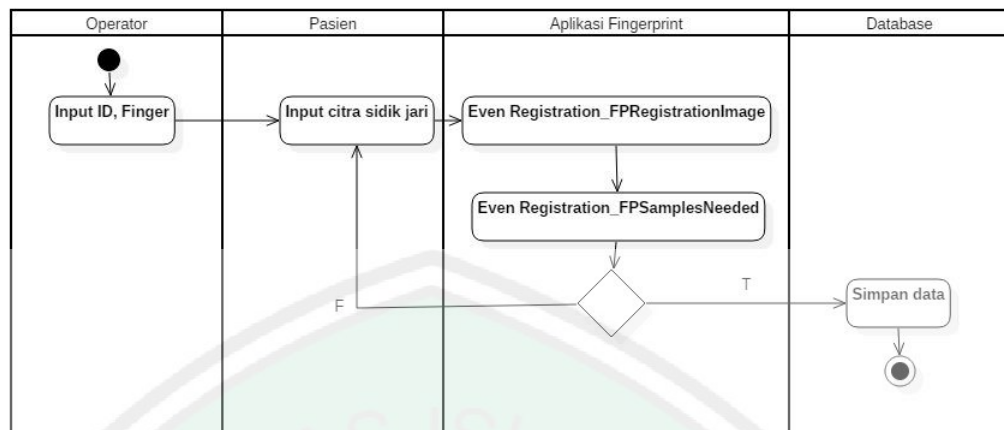
Pada proses ini operator harus memasukkan *username* dan *password* kedalam halaman *login*, setelah *username* dan *password* terinput, sistem akan melakukan pengecekan apakah *username* dan *password* yang diinputkan sama dengan yang tersimpan di dalam *database*.



Gambar 3.15 Activity diagram login

b. Activity Diagram Registras Citra Sidik Jari

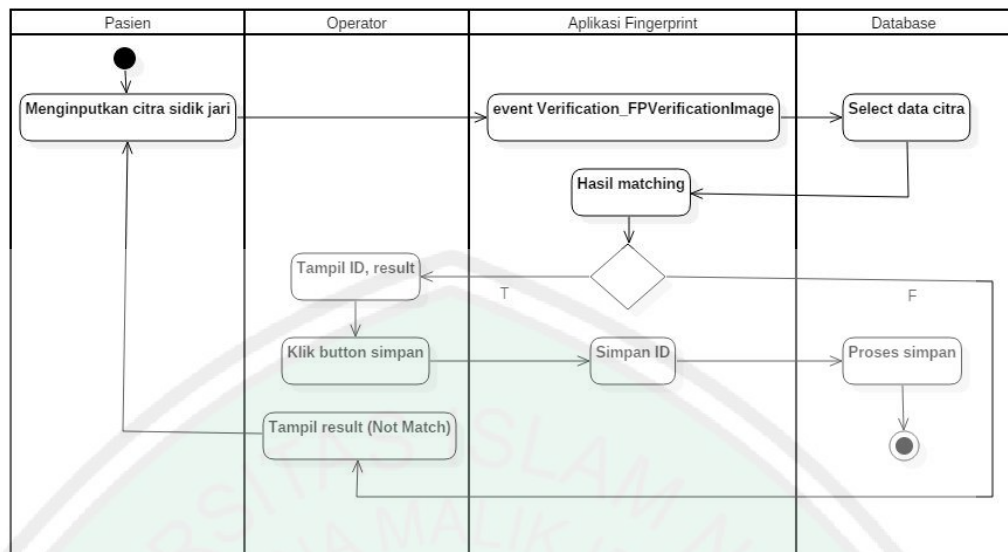
Pada proses ini operator harus terlebih dulu menginputkan nilai ID dan index Finger sidik jari pasien. Setelah field ID dan finger terisi pasien dapat melakukan input citra sidik jari. Sidik jari yang ditempelkan diatas sensor akan otomatis diload oleh SDK sebanyak 4 kali. Ketika data citra sidik jari telah tersimpan akan muncul pesan “Data Berhasil Disimpan”.



Gambar 3.16 Activity diagram registrasi citra sidik jari

c. Activity Diagram Verifikasi Citra Sidik Jari

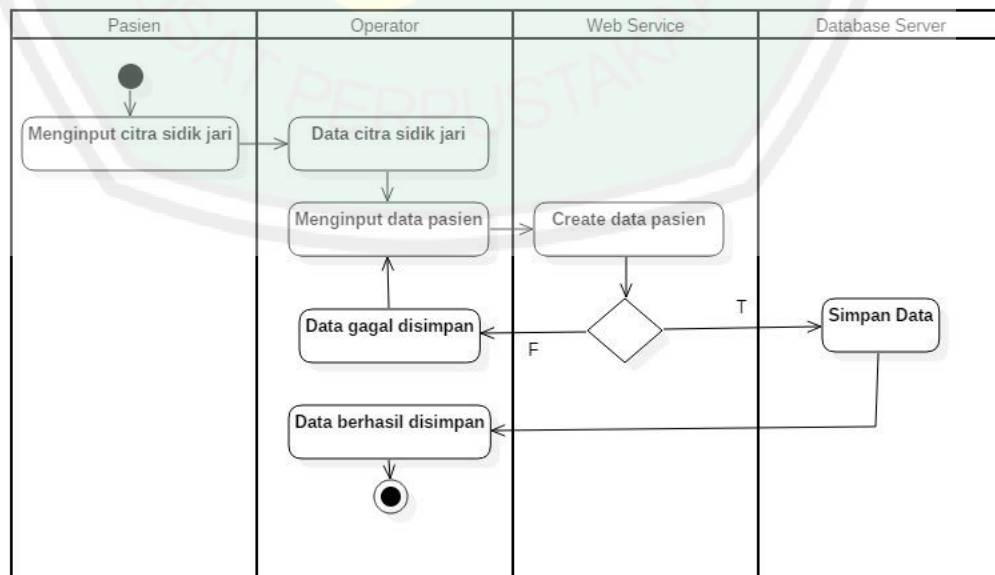
Aktifitas ini diawali dengan pasien meletakkan sidik jari diatas sensor, setelah itu aplikasi fingerprint melakukan verifikasi dengan menggunakan event `Verification_FPVerificationImage` yang merupakan SDK bawaan dari `fingerspot`. Jika data ditemukan akan dimunculkan data ID dan result berupa “match”, namun jika data tidak sama maka akan muncul result “No match”. Setelah data ditemukan operator menyimpan data ID kedalam database sebelum akhirnya akan dijadikan parameter *foreign key* dengan field `FingerData` pada tabel `fingerprint_t`.



Gambar 3.17 Activity diagram verifikasi cintra sidik jari

d. Activity Diagram Registrasi Data Pasien

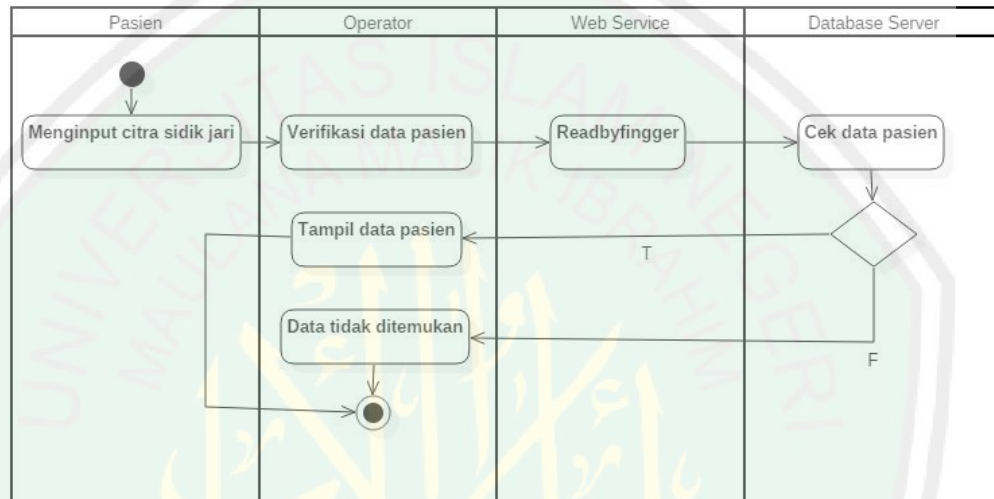
Pada proses ini data sidik jari yang tersimpan di dalam database akan otomatis di *load* kedalam halaman *registrasi*, setelah itu operator melakukan pengisian data pasien. Ketika semua data yang dibutuhkan telah terisi dengan benar maka akan muncul pesan “data berhasil disimpan” namun jika data salah maka muncul peringatan “data salah”.



Gambar 3.18 Activity diagram registrasi data pasien

e. *Activity Diagram Verifikasi Data Pasien*

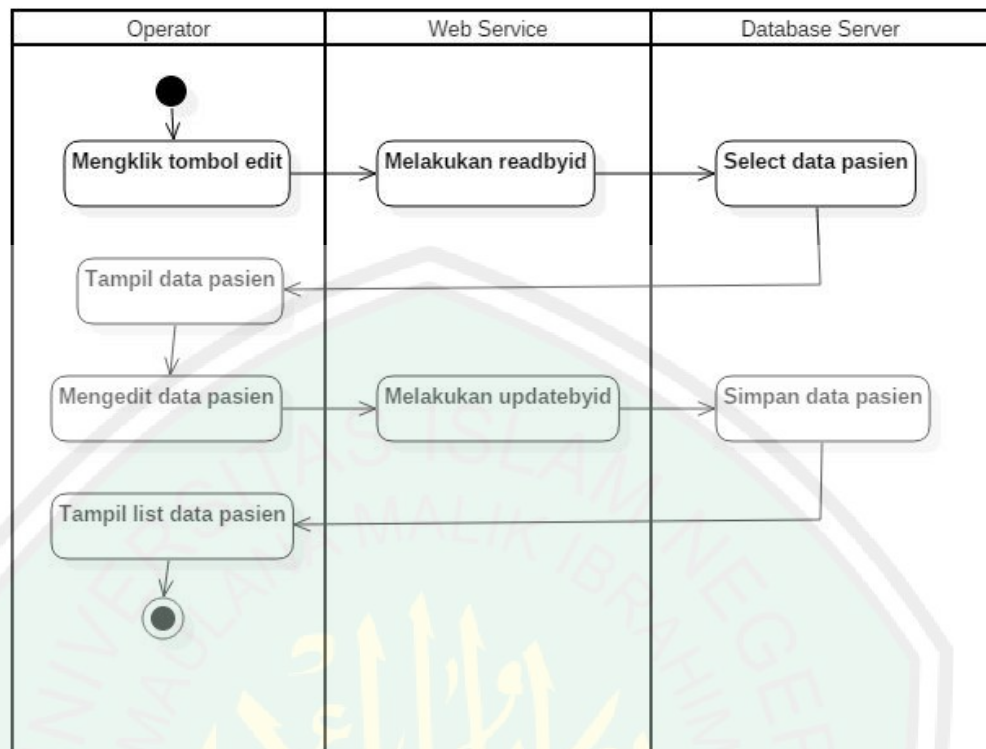
Sama halnya dengan proses *registrasi*, data hasil dari verifikasi citra sidik jari yang tersimpan akan otomatis di load kedalam halaman verifikasi. Data sidik jari inilah yang menjadi parameter input untuk pencarian data pasien.



Gambar 3.19 *Activity diagram verifikasi data pasien*

f. *Activity Diagram Update Data Pasien*

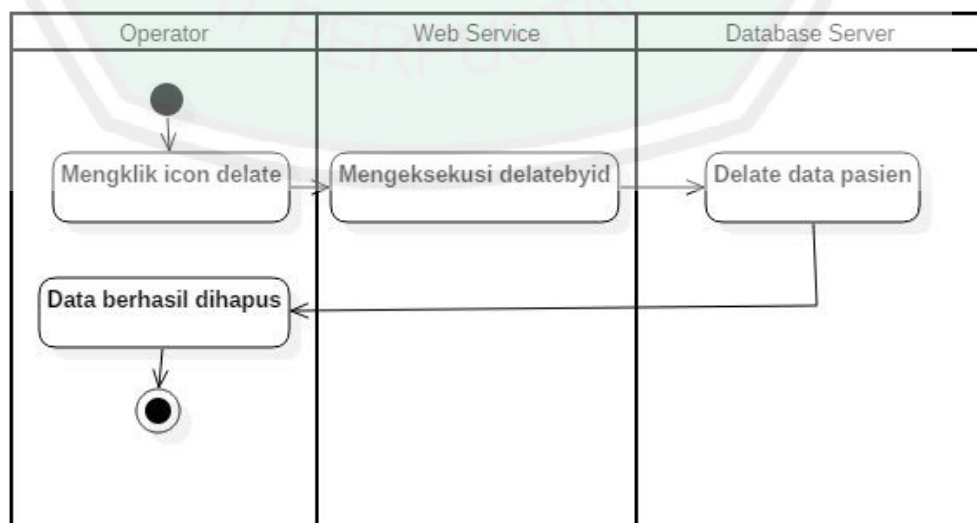
Pada proses ini operator mengklik tombol *edit* yang berada pada list data, setelah itu sistem akan melakukan pencarian data kedalam *database* berdasarkan parameter input ID yang dipilih. Setelah data ditemukan sistem akan menampilkan data kedalam halaman edit. Pada halaman ini operator bisa melakukan edit data pasien, setelah data selesai di edit, operator mengklik tombol *update* untuk melakukan *update* data.



Gambar 3.20 Activity diagram update data pasien

g. Activity Diagram Delete Data Pasien

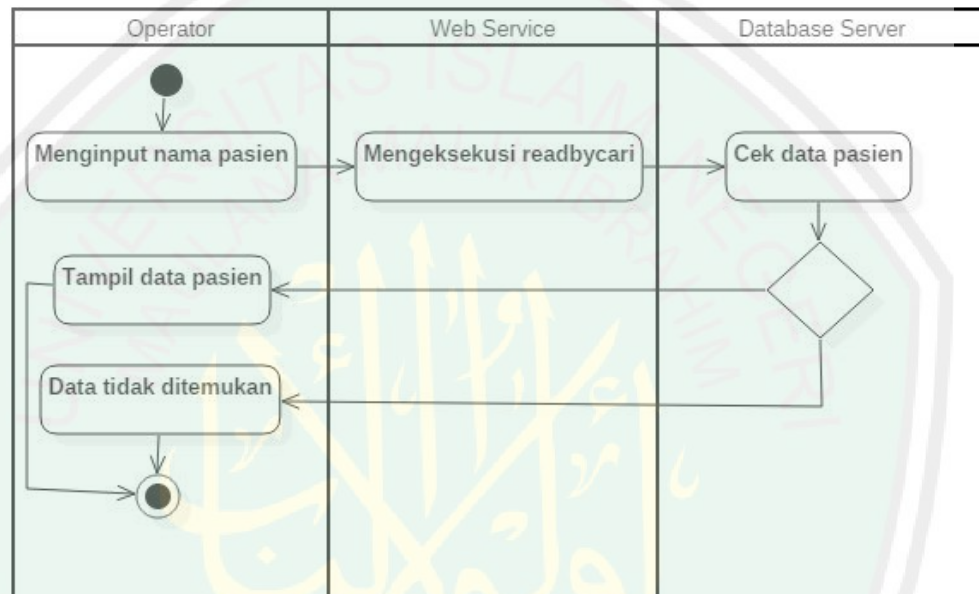
Pada proses ini operator mengklik icon *delete* pada halaman list data, setelah proses *delete* berhasil, maka akan muncul pesan “data berhasil dihapus”.



Gambar 3.21 Activity diagram delete data pasien

#### h. Activity diagram Cari Data Pasien

Pada proses ini operator harus menginputkan nama pasien pada *field search* sebagai parameter input, kemudian operator mengklik tombol *search*. Jika data ditemukan maka akan tampil data pasien, namun jika data tidak ditemukan maka akan muncul pesan “Data tidak ditemukan”.

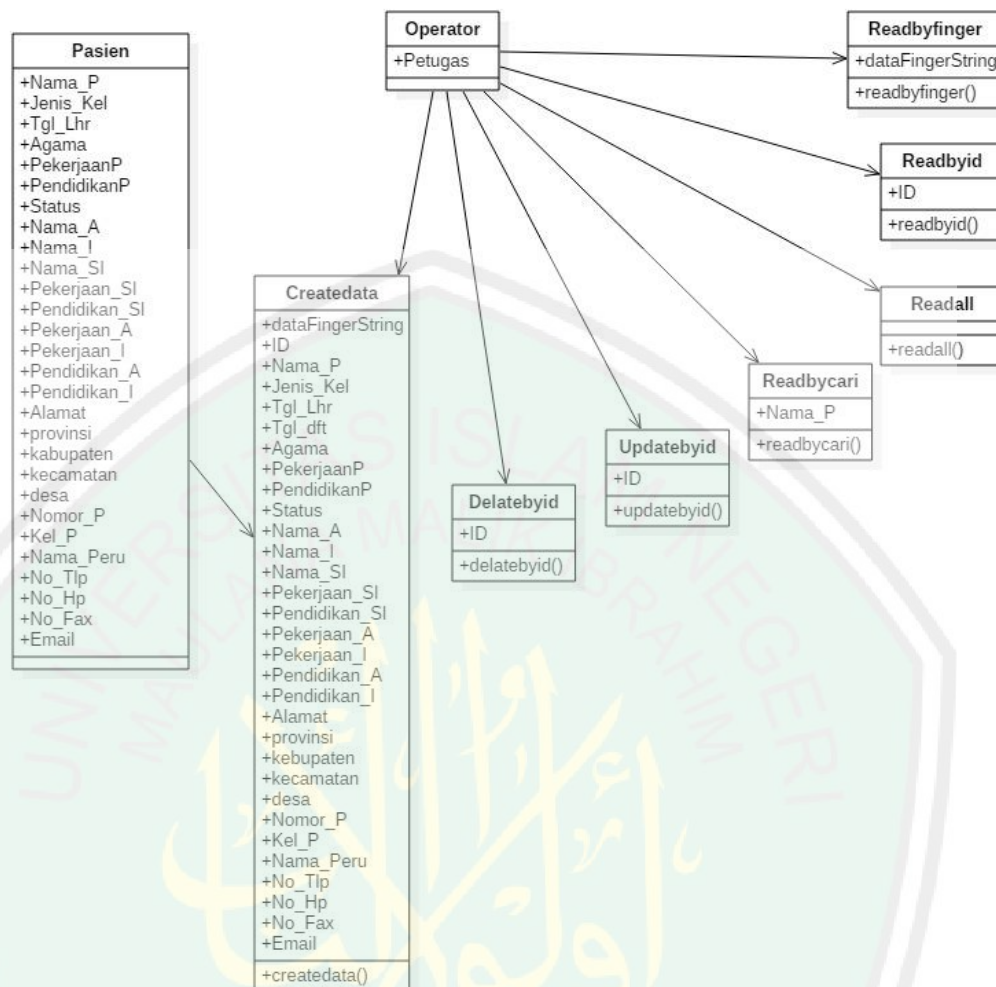


Gambar 3.22 Activity diagram cari data pasien

#### 4. Class Diagram

*Class diagram* digunakan untuk mendeskripsikan jenis – jenis objek di dalam sistem dan berbagai hubungan statis yang terjadi dalam sebuah perancangan perangkat lunak yang juga menunjukkan sebuah *class* dan batasan yang terdapat dalam sebuah hubungan dengan objek.

Berdasarkan *use case* sebelumnya terdapat sembilan *class* yang terbentuk, diantaranya: Operator, Pasien, Createdata, Delatebyid, Updatebyid, Readbyid, Readbycari, Readall, dan Readbyfinger.



Gambar 3.23 Class Diagram

## 5. Database Design

Rancang Bangun Deteksi Sidik Jari Menggunakan Fingerspot™ Easy Touch Sdk Berbasis *Web service* ini diimplementasikan dalam dua buah basis data dengan nama fingerprint sebagai basis data client dan id3791760\_fingerprint\_server sebagai basis data server. Terdapat 7 table pada basis data fingerprint, yaitu users, t\_fingerprint, webservice, provinsi, kabupaten, kecamatan dan desa. Sedangkan pada basis data id3791760\_fingerprint\_server terdapat satu table, yaitu table t\_pasien.

Tabel *users* memiliki 7 *field*, dengan *user\_id* sebagai *primarykey*, detail tabel *users* secara lengkap disajikan pada tabel 3.3.

Tabel 3.3 Tabel users

Nama Field	Type	Deskripsi
create_at	Date	
email	Varchar (255)	
gender	Varchar (255)	
password	Varchar (255)	
phone	Varchar (255)	
username	Varchar (255)	
user_id	Int (11)	AUTO_INCREMENT

Tabel *fingerprint\_t* memiliki 3 *field*, dengan ID sebagai *primarykey*, detail tabel *fingerprint\_t* secara lengkap disajikan pada tabel 3.4.

Table 3.4 Tabel fingerprint\_t

Nama Field	Type	Deskripsi
ID	Varchar (10)	Primary key (ID)
Finger	Tinyint (1)	
FingerData	Varbinary (2000)	

Sama halnya dengan tabel *fingerprint\_t*, Tabel *webservice* juga memiliki 3 *field*, dengan ID sebagai *primarykey*, detail tabel *users* secara lengkap disajikan pada tabel 3.5.

Tabel 3.5 Tabel webservice

Nama Field	Type	Deskripsi
ID	Varchar (10)	Primary key (ID)
Finger	Tinyint (1)	
FingerData	Varbinary (2000)	

Tabel *wilayah\_provinsi* memiliki 2 *field*, dengan *Id* sebagai *primarykey*, detail tabel *wilayah\_provinsi* secara lengkap disajikan pada tabel 3.6.

Tabel 3.6 Tabel wilayah\_provinsi

Nama Field	Type	Deskripsi
Id	Varchar(2)	Primary key (ID)
nama	Varchar(30)	

Tabel wilayah\_kabupaten memiliki 3 *field*, dengan *Id* sebagai *primarykey* dan *Provinsi\_id* sebagai *foreign key* provinsi, detail tabel wilayah provinsi secara lengkap disajikan pada tabel 3.7.

Table 3.7 Tabel wilayah\_kabupaten

Nama Field	Type	Deskripsi
Id	Varchar(4)	Primary key (ID)
Provinsi_id	Varchar(2)	
Nama	Varchar(30)	

Tabel wilayah\_kecamatan memiliki 3 *field*, dengan *Id* sebagai *primarykey* dan *kecamatan\_id* sebagai *foreign key* kabupaten, detail tabel wilayah\_kecamatan secara lengkap disajikan pada tabel 3.8.

Table 3.8 Tabel wilayah\_kecamatan

Nama Field	Type	Deskripsi
Id	Varchar(7)	Primary key (ID)
Kabupaten_id	Varchar(4)	
Nama	Varchar(30)	

Tabel wilayah\_desa memiliki 3 *field*, dengan *Id* sebagai *primarykey* dan *kecamatan\_id* sebagai *foreign key* kecamatan, detail tabel wilayah\_desa secara lengkap disajikan pada tabel 3.9.

Table 3.9 Tabel wilayah\_desa

Nama Field	Type	Deskripsi
Id	Varchar(10)	Primary key (ID)
Kecamatan_id	Varchar(7)	
Nama	Varchar(30)	

Tabel t\_pasien yang berada pada database id3791760\_fingerprint\_server memiliki 30 *field*, dengan *ID* sebagai *primarykey*, detail tabel t\_pasien secara lengkap disajikan pada tabel 3.10.

Table 3.10 Tabel t\_pasien

Nama Field	Type	Deskripsi
ID	int (10)	Primary key
dataFingerString	Varchar (100)	
Nama_P	Varchar (25)	
Jenis_Kel	Varchar (2)	
Tgl_lhr	Date	
Tgl_dft	Date	
Agama	Varchar (20)	
PendidikanP	Varchar (20)	
PekerjaanP	Varchar (10)	
Status	Varchar (11)	
Nama_A	Varchar (25)	
Nama_I	Varchar (25)	
Pendidikan_SI	Varchar (20)	
Pekerjaan_SI	Varchar (20)	
Pekerjaan_A	Varchar (20)	
Pekerjaan_I	Varchar (20)	
Pendidikan_A	Varchar (20)	
Pendidikan_I	Varchar (20)	
Alamat	Varchar (50)	
Provinsi	Varchar(30)	
Kabupaten	Varchar(30)	
Kecamatan	Varchar(30)	
Desa	Varchar(30)	
Nomor_P	Int	
Kel_P	Varchar(25)	
Nama_Peru	Varchar(25)	
No_Tlp	Int	

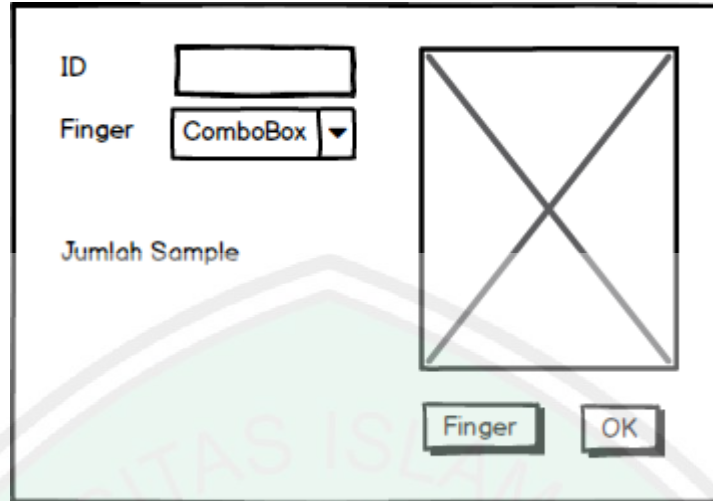
No.Hp	Int	
No.Fax	Int	
Email	Varchar(20)	

### 3.3.4 Perancangan Antar Muka

Pada bagian ini akan ditampilkan rancangan antarmuka (*interface*) sistem pendaftaran pasien. Beberapa rancangan tampilan diantaranya adalah halaman login, halaman registrasi citra sidik jari, halaman verifikasi citra sidik jari, halaman registrasi data pasien, halaman verifikasi data pasien, halaman list data, dan halaman update data pasien.

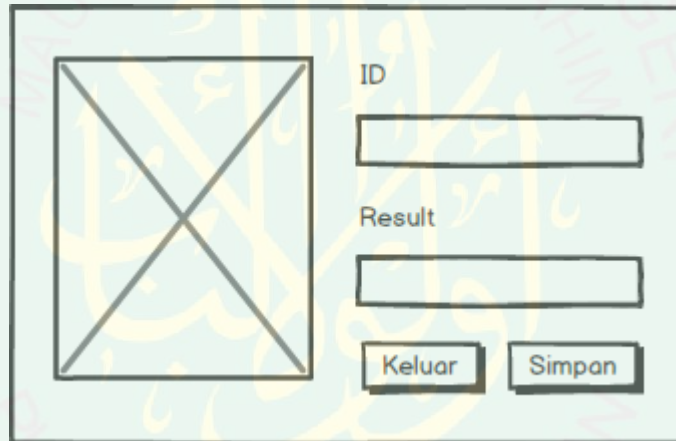


Gambar 3.24 Halaman *login* admin



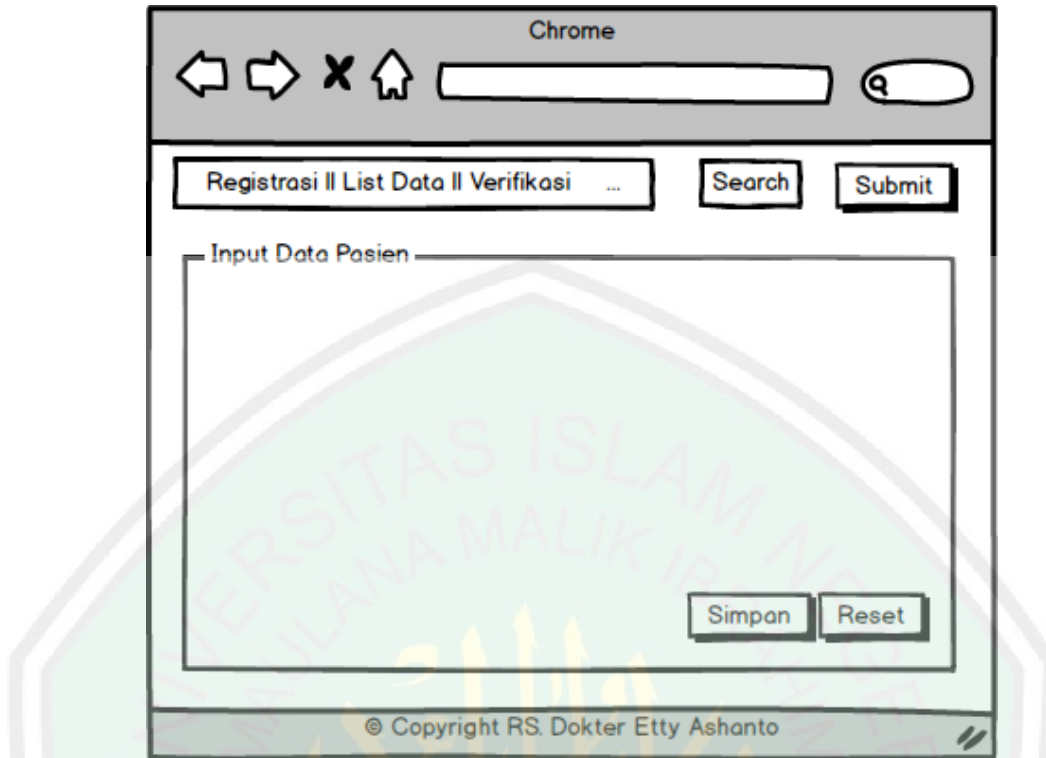
The registration interface consists of a rectangular window. On the left side, there are three labels: 'ID' above a text input field, 'Finger' above a 'ComboBox' dropdown menu, and 'Jumlah Sample' above a text input field. On the right side, there is a large square area with a diagonal 'X' across it, indicating a placeholder for a fingerprint image. At the bottom right of the window, there are two buttons: 'Finger' and 'OK'.

Gambar 3.25 *Interface* halaman registrasi citra sidik jari



The verification interface consists of a rectangular window. On the left side, there is a large square area with a diagonal 'X' across it, indicating a placeholder for a fingerprint image. On the right side, there are two labels: 'ID' above a text input field, and 'Result' above another text input field. At the bottom right of the window, there are two buttons: 'Keluar' and 'Simpan'.

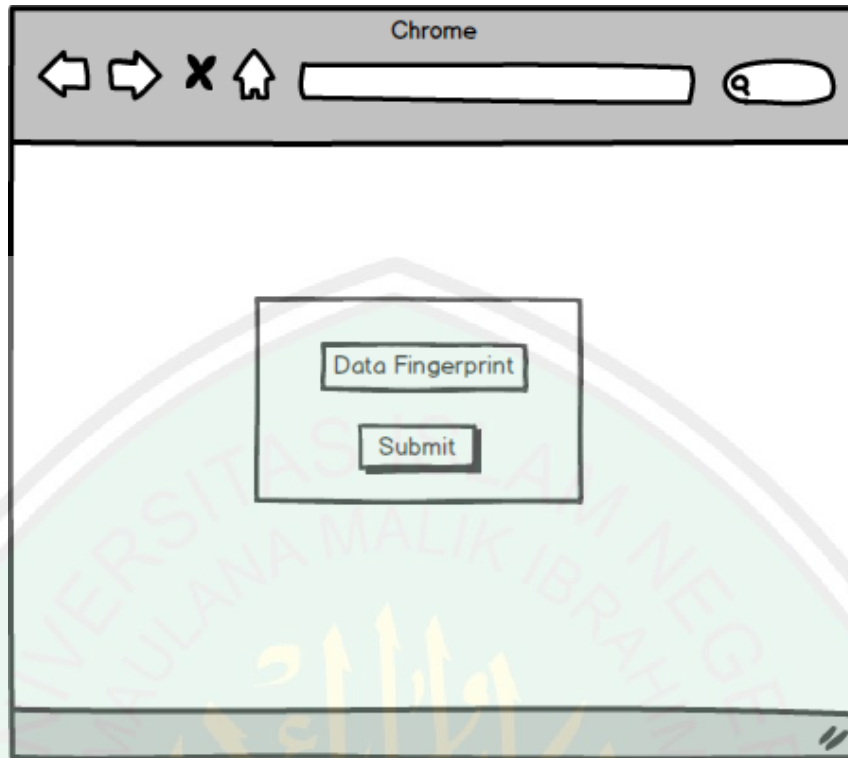
Gambar 3.26 *Interface* halaman verifikasi citra sidik jari



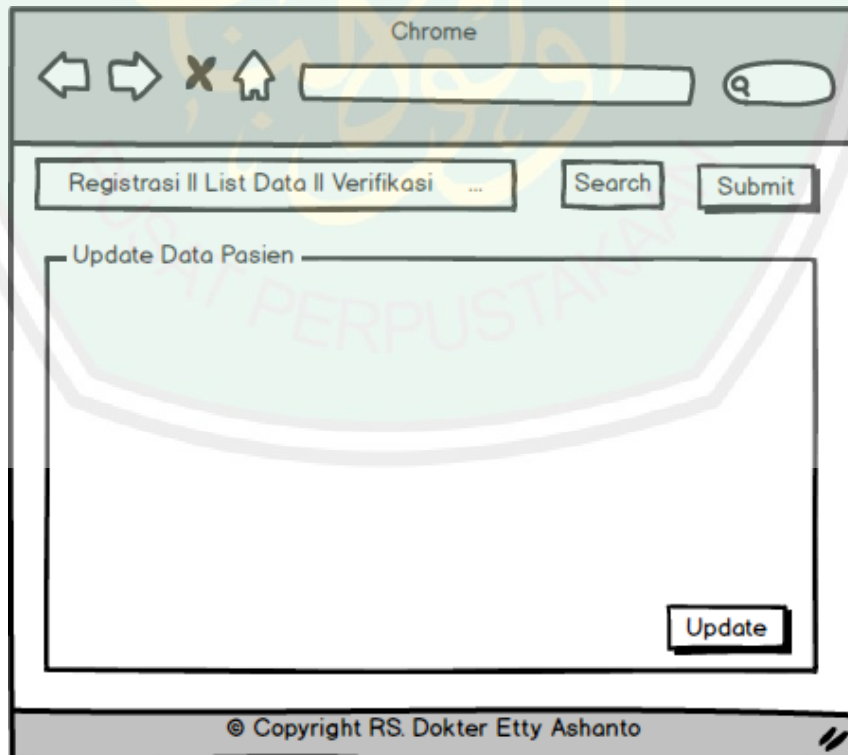
Gambar 3.27 *Interface* halaman registrasi data pasien



Gambar 3.28 *Interface* halaman list data



Gambar 3.29 *Interface* halaman verifikasi data pasien



Gambar 3.30 *Interface* halaman *update* data pasien

## BAB IV

### HASIL DAN PEMBAHASAN

Pada bab hasil dan pembahasan ini akan dijelaskan tentang implementasi sistem yang meliputi (implementasi basis data, implementasi *web service*, implementasi *web client*, dan implementasi aplikasi finger), serta pengujian sistem.

#### 4.1 Kebutuhan Pengujian Sistem

Kebutuhan pengujian sistem meliputi kebutuhan perangkat keras dan kebutuhan perangkat lunak seperti berikut:

- .1 Kebutuhan Perangkat Keras
  - a. *Procesor AMD E-350*
  - b. *Memory RAM 4GB DDR3*
  - c. *Hardisk 320GB*
  - d. *Monitor 14"*
  - e. *Keyboard*
  - f. *Mouse*
  - g. *Fingerspot*
- .2 Kebutuhan Perangkat Lunak
  - a. *Sistem Operasi Windows 10 64 bit*
  - b. *Visual Basic 6.0*
  - c. *FileZilla Client*
  - d. *Mysql Connector ODBC Versi 5.3.7*
  - e. *Notepad++ Versi 7.3.3*
  - f. *Driver Easy Touch SDK*

- g. *Xampp Versi 3.2.2*
- h. *Google Chrome 58.0.3029.11*

## 4.2 Implementasi Sistem

### 4.2.1 Implementasi Basis Data

Berdasarkan desain basis data yang telah dibuat sebelumnya, basis data diletakkan kedalam dua host yang berbeda, yaitu basis data *fingerprint* diletakkan pada localhost client sedangkan basis data *id3791760\_fingerprint\_server* pada <https://databases-auth.000webhost.com/index.php>.

Berikut merupakan *script* yang digunakan untuk membangun table pada *database fingerprint*.

-----  
 --- Table structure for table `users`  
 -----

```
CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `gender` varchar(255) NOT NULL,
  `phone` varchar(15) NOT NULL,
  `create_at` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `users`
```

```
ADD PRIMARY KEY (`user_id`);
```

```
-----  
--- Table structure for table `fingerprint_t`  
-----
```

```
CREATE TABLE `fingerprint_t` (  
  `ID` int(10) NOT NULL,  
  `Finger` tinyint(2) NOT NULL,  
  `FingerData` varbinary(2000) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
ALTER TABLE `fingerprint_t`  
  ADD PRIMARY KEY (`ID`);
```

```
-----  
--- Table structure for table `web service`  
-----
```

```
CREATE TABLE `webservice` (  
  `ID` int(10) NOT NULL,  
  `Finger` tinyint(2) NOT NULL,  
  `FingerData` varbinary(2000) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
DELIMITER $$
```

```
CREATE TRIGGER `webservice_insert` BEFORE INSERT ON `webservice`  
FOR EACH ROW SET NEW.FingerData = (SELECT FingerData FROM  
fingerprint_t WHERE ID = NEW.ID)
```

```
$$
```

DELIMITER ;

-----  
 --- Table structure for table 'wilayah\_provinsi'

```
CREATE TABLE `wilayah_provinsi` (
  `id` varchar(2) NOT NULL,
  `nama` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `wilayah_provinsi`
ADD PRIMARY KEY (`id`);
```

-----  
 --- Table structure for table 'wilayah\_kabupaten'

```
CREATE TABLE `wilayah_kabupaten` (
  `id` varchar(4) NOT NULL,
  `provinsi_id` varchar(2) NOT NULL DEFAULT "",
  `nama` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `wilayah_kabupaten`
ADD PRIMARY KEY (`id`);
```

-----  
 --- Table structure for table 'wilayah\_kecamatan'

```
CREATE TABLE `wilayah_kecamatan` (
  `id` varchar(7) NOT NULL,
```

```

`kabupaten_id` varchar(4) NOT NULL DEFAULT "",
`nama` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `wilayah_kecamatan`
  ADD PRIMARY KEY (`id`);

```

```

-----
--- Table structure for table `desa`
-----

```

```

CREATE TABLE `wilayah_desa` (
  `id` varchar(10) NOT NULL,
  `kecamatan_id` varchar(7) DEFAULT NULL,
  `nama` varchar(40) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

Berikut merupakan *script* yang digunakan untuk membangun table pada *database* id3791760\_fingerprint\_server.

```

-----
--- Struktur table untuk t_pasien
-----

```

```

CREATE TABLE `t_pasien` (
  `ID` int(5) NOT NULL,
  `FingerData` varchar(100) NOT NULL,
  `Nama_P` varchar(25) NOT NULL,
  `Jenis_Kel` varchar(20) NOT NULL,
  `Tgl_Lhr` date NOT NULL,

```

`Tgl\_dft` date NOT NULL,  
`Agama` varchar(100) NOT NULL,  
`PekerjaanP` varchar(15) NOT NULL,  
`PendidikanP` varchar(15) NOT NULL,  
`Status` varchar(11) NOT NULL,  
`Nama\_A` varchar(25) NOT NULL,  
`Nama\_I` varchar(25) NOT NULL,  
`Nama\_SI` varchar(25) NOT NULL,  
`Pekerjaan\_SI` varchar(20) NOT NULL,  
`Pendidikan\_SI` varchar(20) NOT NULL,  
`Pekerjaan\_A` varchar(20) NOT NULL,  
`Pekerjaan\_I` varchar(20) NOT NULL,  
`Pendidikan\_A` varchar(20) NOT NULL,  
`Pendidikan\_I` varchar(20) NOT NULL,  
`Alamat` varchar(50) NOT NULL,  
`provinsi` varchar(30) NOT NULL,  
`kabupaten` varchar(30) NOT NULL,  
`kecamatan` varchar(30) NOT NULL,  
`kelurahan` varchar(30) NOT NULL,  
`Provinsi` varchar(20) NOT NULL,  
`Nomor\_P` int(11) NOT NULL,  
`Kel\_P` varchar(25) NOT NULL,  
`Nama\_Peru` varchar(25) NOT NULL,  
`No\_Tlp` int(11) NOT NULL,

```

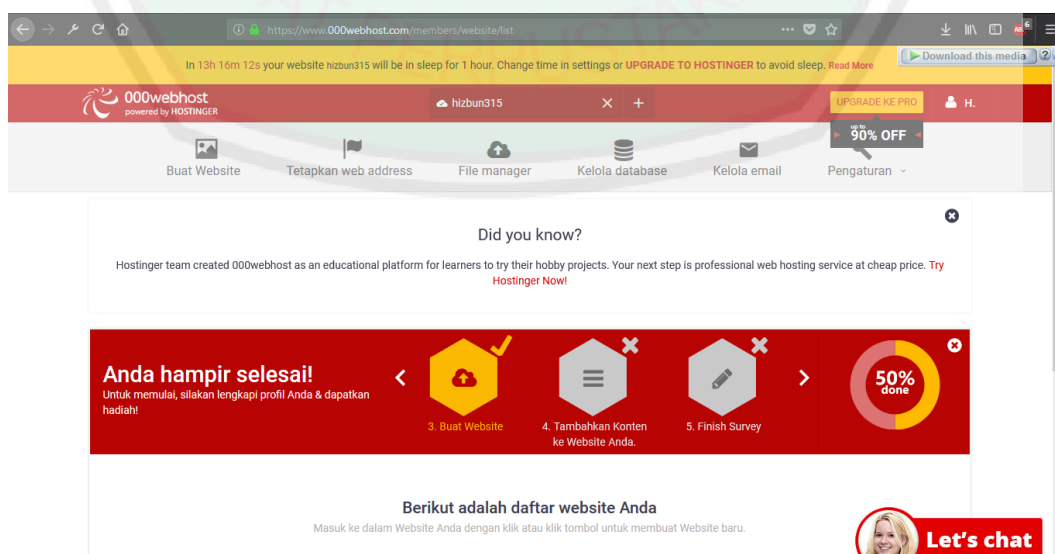
`No_Hp` int(11) NOT NULL,
`No_Fax` int(11) NOT NULL,
`Email` varchar(20) NOT NULL,
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `t_pasien`
ADD PRIMARY KEY (`ID`);

```

#### 4.2.2 Implementasi Web Service

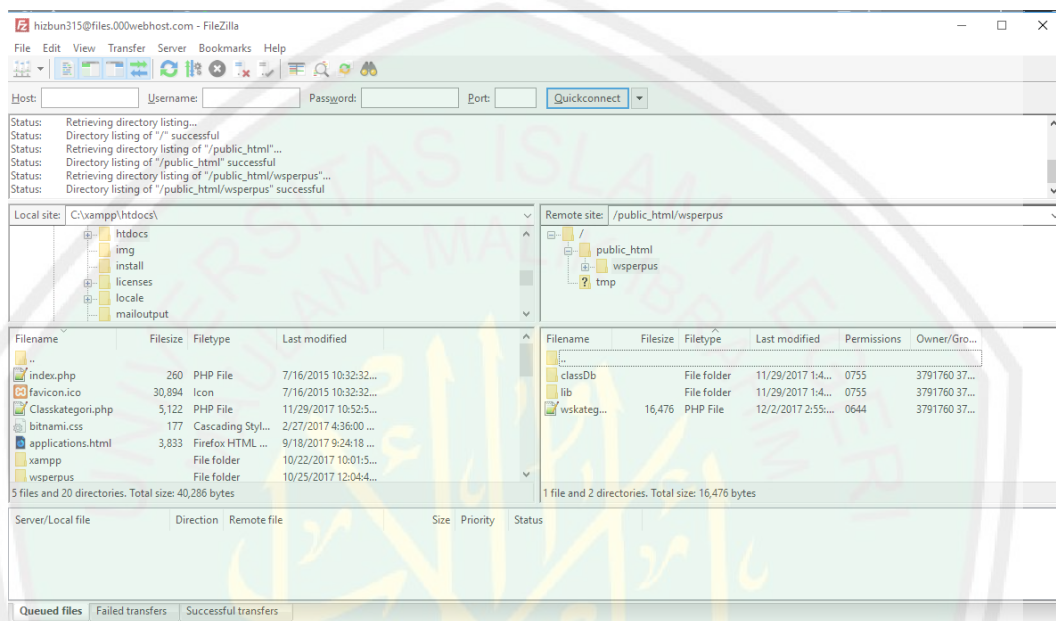
Untuk melakukan implementasi *web service* ini, penulis menggunakan layanan web hosting gratis dari 000webhost dengan domain <http://hizbun315.000webhostapp.com>.

Adapun penerapan *web service* pendaftaran pasien dilakukan dengan beberapa tahapan, diantaranya adalah upload *database* kedalam host yang telah disediakan pada domain <http://hizbun315.000webhostapp.com>, upload *web service* dengan menggunakan filezilla, dan melihat hasil dari konfigurasi WSDL untuk memastikan apakah *web service* telah berfungsi sebagaimana mestinya.



Gambar 4.1 Tampilan halaman kelola pada 000webhost

Untuk mengupload *web service* ke dalam web hosting, penulis menggunakan filezilla client sebagai protokol pengiriman data. Sebagaimana yang ditunjukkan pada gambar 4.2 bahwa alamat host yang dituju adalah `hizbun315@files.000webhost.com`.

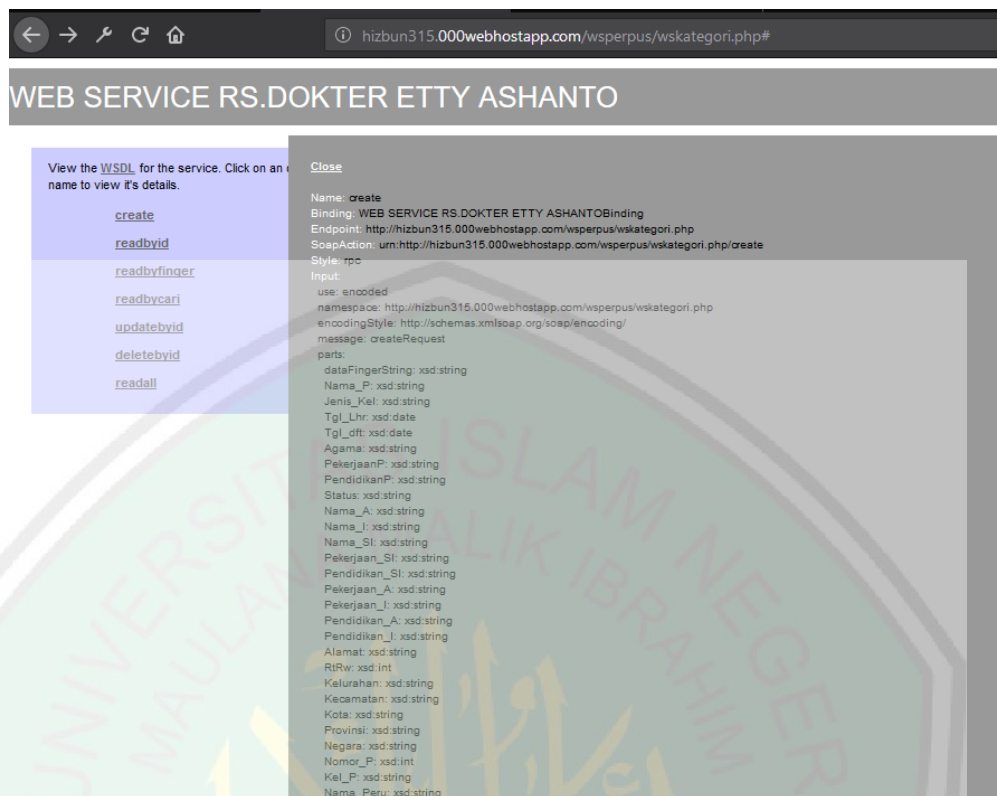


Gambar 4.2 Tampilan Filezilla

Setelah *database* dan file *web service* telah di upload, selanjutnya dilakukan pengujian pada masing-masing function yang telah dibuat untuk memastikan apakah konfigurasi WSDL telah sesuai atau tidak.

#### 1. Halaman Hasil Konfigurasi WSDL *Create*

Halaman ini merupakan hasil dari konfigurasi *function create* pada WSDL yang dibuat untuk menangani proses input data pasien baru dengan menggunakan layanan *web service*.



Gambar 4.3 Tampilan hasil WSDL *function create*

Berdasarkan hasil WSDL *function create* diatas, terdapat beberapa *element – element* yang terbentuk dari *function* tersebut. Diantaranya: *elemen operation* “create”, *element message* “createRequest dan createResponse”, *element binding* “WEB SERVICE RS. DOKTER ETTY ASHANTOBinding”, dan *element service* “urn:http://hizbun315.000webhostapp.com/wsperpus/wskategori.php/create”.

Adapun *source code register function create* dapat dilihat pada gambar 4.4.

```

1 //create data pasien
2 $input_create = array(
3     'dataFingerString' =>"xsd:string",
4     'Nama_P' => "xsd:string",
5     'Jenis_Kel' => "xsd:string",
6     'Tgl_Lhr' => "xsd:date",
7     'Tgl_dft' => "xsd:date",
8     'Agama' => "xsd:string",
9     'PekerjaanP' => "xsd:string",

```

```

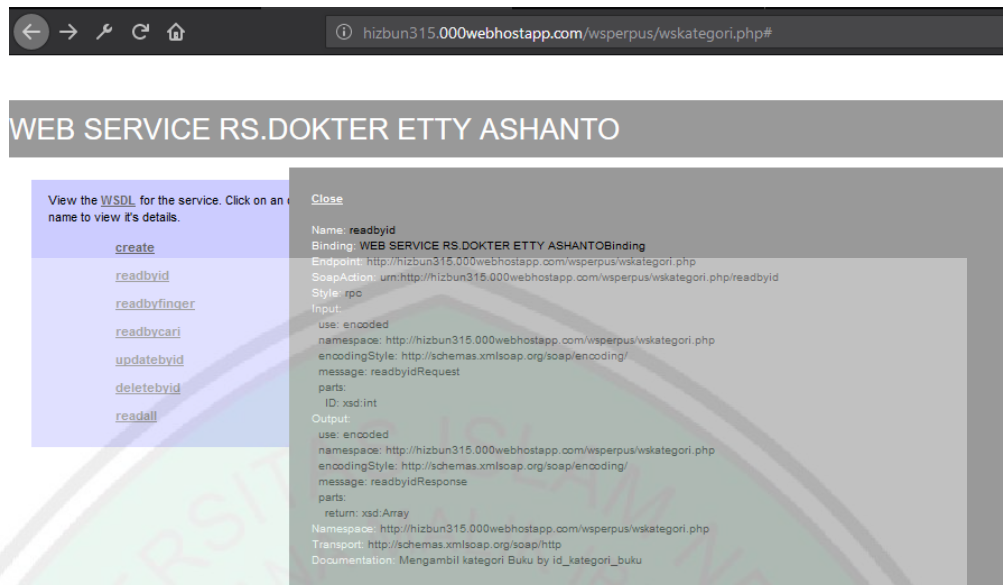
10 'PendidikanP' => "xsd:string",
11 'Status' => "xsd:string",
12 'Nama_A' => "xsd:string",
13 'Nama_I' => "xsd:string",
14 'Nama_SI' => "xsd:string",
15 'Pekerjaan_SI' => "xsd:string",
16 'Pendidikan_SI' => "xsd:string",
17 'Pekerjaan_A' => "xsd:string",
18 'Pekerjaan_I' => "xsd:string",
19 'Pendidikan_A' => "xsd:string",
20 'Pendidikan_I' => "xsd:string",
21 'Alamat' => "xsd:string",
22 'desa' => "xsd:string",
23 'kecamatan' => "xsd:string",
24 'kabupaten' => "xsd:string",
25 'provinsi' => "xsd:string",
26 'Negara' => "xsd:string",
27 'Nomor_P' => "xsd:int",
28 'Kel_P' => "xsd:string",
29 'Nama_Peru' => "xsd:string",
30 'No_Tlp' => "xsd:int",
31 'No_Hp' => "xsd:int",
32 'No_Fax' => "xsd:int",
33 'Email' => "xsd:string"); // parameter input
34 $return_create = array("return" => "xsd:string");
35 $server->register('create',
36     $input_create,
37     $return_create,
38     $ns,
39     "urn:". $ns."/create",
40     "rpc",
41     "encoded",
42     "Menyimpan Data Pasien Baru");
43 //end create data pasien
44

```

Gambar 4.4 Register function create

## 2. Halaman Hasil Konfigurasi WSDL Readbyid

Halaman ini merupakan hasil dari konfigurasi *function readbyid* pada WSDL yang dibuat untuk menangani proses pencarian data pasien berdasarkan nomor ID dengan menggunakan layanan *web service*.



Gambar 4.5 Tampilan hasil WSDL function readbyid

Berdasarkan hasil WSDL function readbyid diatas, terdapat beberapa elemen – elemen yang terbentuk dari function tersebut. Diantaranya: elemen operation “readbyid”, *element message* “readbyidRequest dan readbyidResponse”, *element binding* “WEB SERVICE RS. DOKTER ETTY ASHANTOBinding”, dan *element service* “urn:http://hizbun315.000webhostapp.com/wsperpus/wskategori.php/readbyid”.

Adapun *source code register function readbyid* dapat dilihat pada gambar 4.6.

```

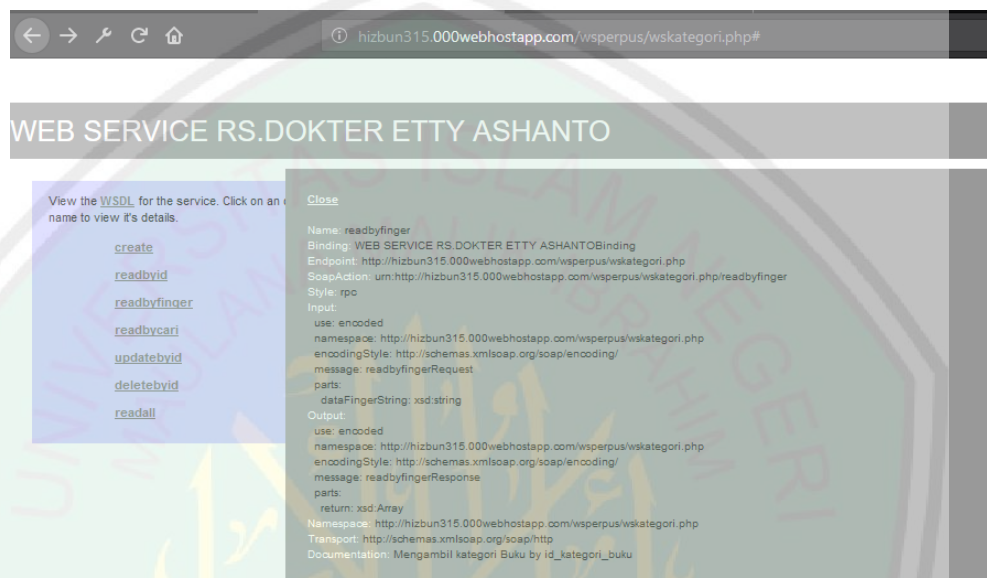
1 //readbyid data pasien
2 $input_readbyid = array('ID' => "xsd:int");
3 $return_readbyid = array(
4     "return" => "xsd:Array");
5 $server->register('readbyid',
6 $input_readbyid,
7 $return_readbyid,
8 $ns,
9 "urn:".$ns."/readbyid",
10 "rpc",
11 "encoded",
12 "Mengambil Data Pasien Berdasarkan ID");
13 //end readbyid

```

Gambar 4.6 Source code register readbyid

### 3. Halaman Hasil Konfigurasi WSDL Readbyfinger

Halaman ini merupakan hasil dari konfigurasi *function readbyfinger* pada WSDL yang dibuat untuk menangani proses verifikasi data pasien berdasarkan data *finger* yang tersimpan pada *database client*.



Gambar 4.7 Tampilan hasil WSDL *function readbyfinger*

Berdasarkan hasil WSDL *function readbyfinger* diatas, terdapat beberapa *element – element* yang terbentuk dari *function* tersebut. Diantaranya: *elemen operation* “readbyfinger”, *element message* “readbyfingerRequest dan readbyfingerResponse”, *element binding* “WEB SERVICE RS. DOKTER ETTY ASHANTOBinding”, dan *element service* “urn:http://hizbun315.000webhostapp.com/wsperpus/wskategori.php/readbyfinger”.

Adapun *source code register function readbyfinger* dapat dilihat pada gambar 4.8.

1	/readbyfinger data pasien
2	\$input_readbyfinger = array(
3	'dataFingerString' => "xsd:string");
4	\$return_readbyfinger = array(
5	"return" => "xsd:Array");

```

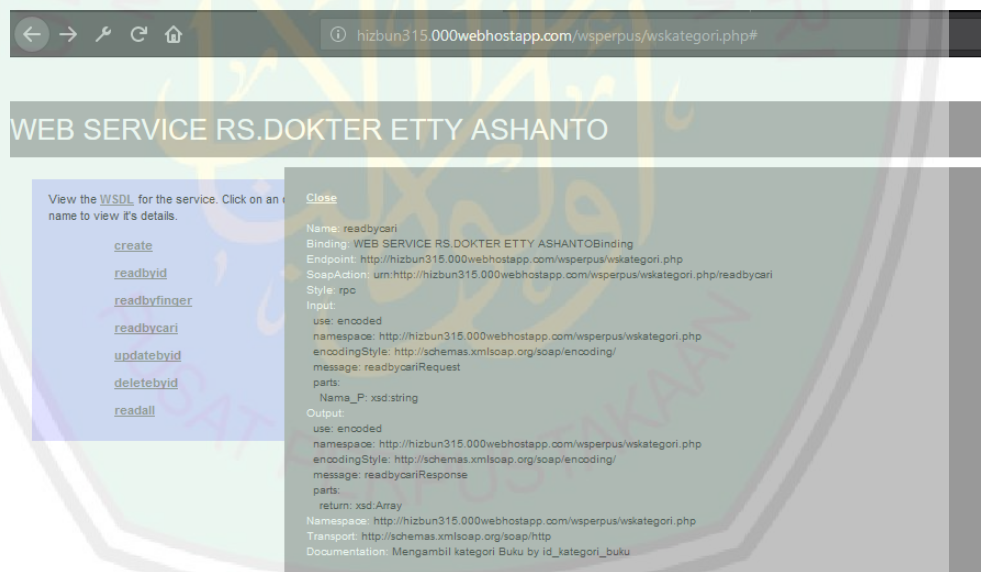
6 $server->register('readbyfinger',
7 $input_readbyfinger,
8 $return_readbyfinger,
9 $ns,
10 "urn:". $ns."/readbyfinger",
11 "rpc",
12 "encoded",
13 "Mengambil Data Pasien Berdasarkan Data
14     Fingerprint");
15 //end readbfinger

```

Gambar 4.8 Source code register readbyfinger

#### 4. Halaman Hasil Konfigurasi WSDL Readbycari

Halaman ini merupakan hasil dari konfigurasi *function readbycari* pada WSDL yang dibuat untuk menangani proses pencarian data pasien berdasarkan nama pasien.

Gambar 4.9 Tampilan hasil WSDL *function readbycari*

Berdasarkan hasil WSDL *function readbycari* diatas, terdapat beberapa *element – element* yang terbentuk dari *function* tersebut. Diantaranya: *elemen operation* “readbycari”, *element message* “readbycariRequest dan readbycariResponse”, *element binding* “WEB SERVICE RS. DOKTER ETTY ASHANTOBinding”, dan *element service*

“urn:http://hizbun315.000webhostapp.com/wsperpus/wskategori.php/readbycari”.

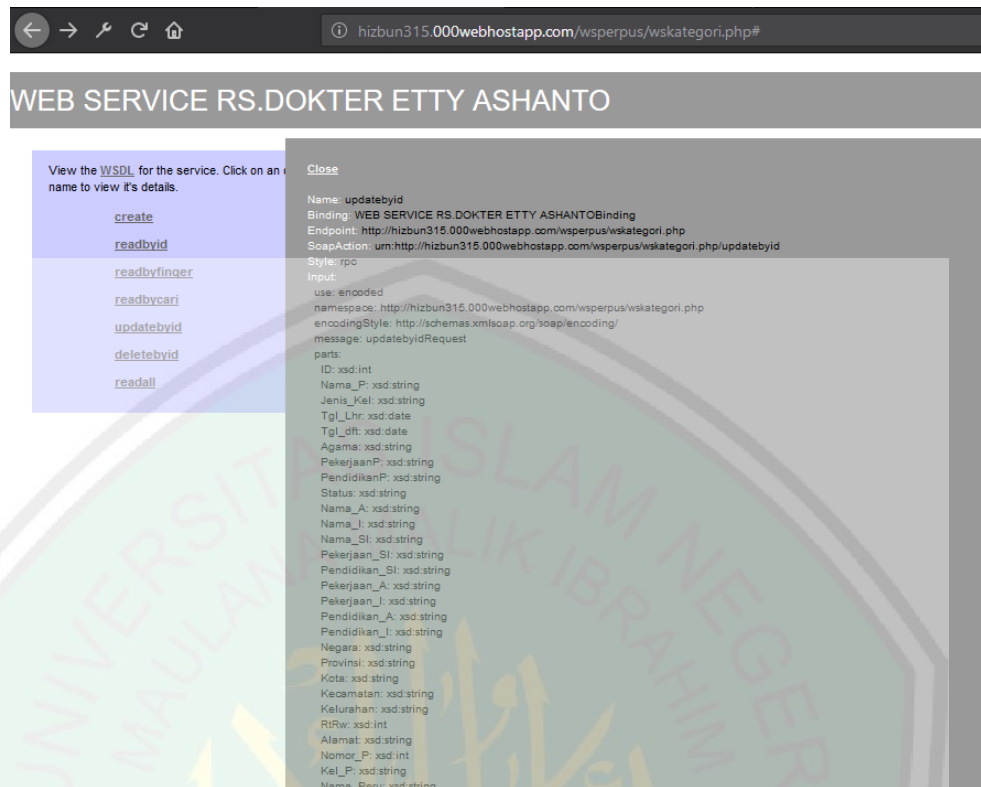
Adapun *source code register function readbycari* dapat dilihat pada gambar 4.10.

```
1 //readbycari data pasien
2 $input_readbycari = array(
3     'Nama_P' => "xsd:string");
4 $return_readbycari = array(
5     "return" => "xsd:Array");
6 $server->register('readbycari',
7     $input_readbycari,
8     $return_readbycari,
9     $ns,
10    "urn:". $ns."/readbycari",
11    "rpc",
12    "encoded",
13    "Mengambil Data Pasien Berdasarkan Nama
14     Pasien");
15 //end readbycari
```

Gambar 4.10 *Source code register readbycari*

#### 5. Halaman Hasil Konfigurasi WSDL Updatebyid

Halaman ini merupakan hasil dari konfigurasi *function updatebyid* pada WSDL yang dibuat untuk menangani proses *update* data pasien berdasarkan nomer ID pasien.



Gambar 4.11 Tampilan hasil WSDL *function updatebyid*

Berdasarkan hasil WSDL *function updatebyid* diatas, terdapat beberapa *element – element* yang terbentuk dari *function* tersebut. Diantaranya: *elemen operation* “updatebyid”, *element message* “updatebyidRequest dan updatebyidResponse”, *element binding* “WEB SERVICE RS. DOKTER ETTY ASHANTOBinding”, dan *element service* “urn:http://hizbun315.000webhostapp.com/wsperpus/wskategori.php/updatebyid”.

Adapun *source code register function updatebyid* dapat dilihat pada gambar 4.12.

1	//update data pasien
2	\$input_update = array(
3	'ID' => "xsd:int",
4	'Nama_P' => "xsd:string",
5	'Jenis_Kel' => "xsd:string",
6	'Tgl_Lhr' => "xsd:date",
7	'Tgl_dft' => "xsd:date",

```

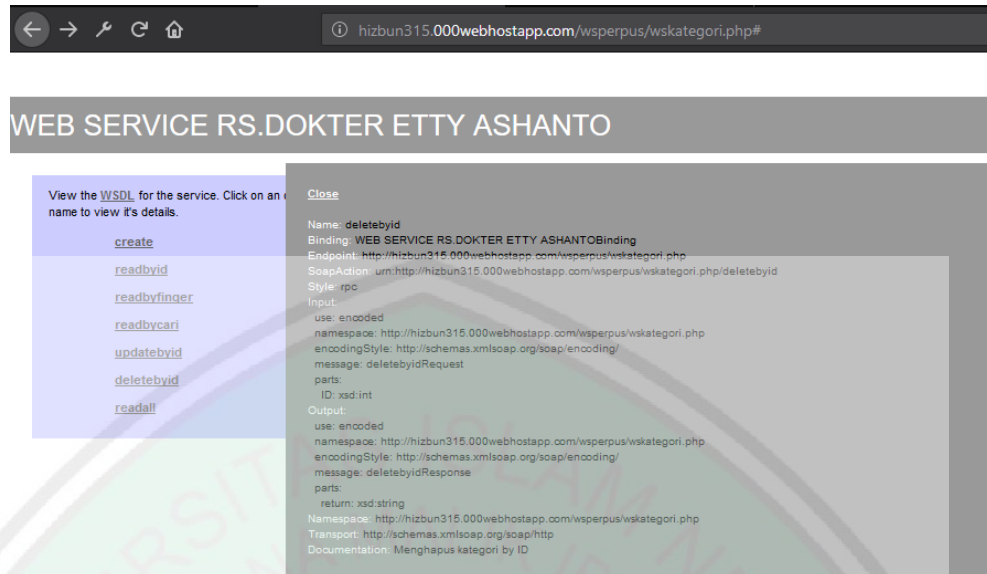
8      'Agama' => "xsd:string",
9      'PekerjaanP' => "xsd:string",
10     'PendidikanP' => "xsd:string",
11     'Status' => "xsd:string",
12     'Nama_A' => "xsd:string",
13     'Nama_I' => "xsd:string",
14     'Nama_SI' => "xsd:string",
15     'Pekerjaan_SI' => "xsd:string",
16     'Pendidikan_SI' => "xsd:string",
17     'Pekerjaan_A' => "xsd:string",
18     'Pekerjaan_I' => "xsd:string",
19     'Pendidikan_A' => "xsd:string",
20     'Pendidikan_I' => "xsd:string",
21     'Alamat' => "xsd:string",
22     'provinsi' => "xsd:string",
23     'kabupaten' => "xsd:string",
24     'kecamatan' => "xsd:string",
25     'desa' => "xsd:string",
26     'Negara' => "xsd:string",
27     'Nomor_P' => "xsd:int",
28     'Kel_P' => "xsd:string",
29     'Nama_Peru' => "xsd:string",
30     'No_Hp' => "xsd:int",
31     'No_Hp' => "xsd:int",
32     'No_Fax' => "xsd:int",
33     'Email' => "xsd:string");// parameter update
34     $return_update = array(
35         "return" => "xsd:string");
36     $server->register('updatebyid',
37     $input_update,
38     $return_update,
39     $ns,
40     "urn:.$ns./updatebyid",
41     "rpc",
42     "encoded",
43     "Mengupdate Data Pasien Berdasarkan ID");
44     //end update

```

Gambar 4.12 Source code register updatebyid

## 6. Halaman Hasil Konfigurasi WSDL Delatebyid

Halaman ini merupakan hasil dari konfigurasi *function delatebyid* pada WSDL yang dibuat untuk menangani proses penghapusan data pasien berdasarkan nomer ID pasien.



Gambar 4.13 Tampilan hasil WSDL *function delatebyid*

Berdasarkan hasil WSDL *function delatebyid* diatas, terdapat beberapa *element – element* yang terbentuk dari *function* tersebut. Diantaranya: *elemen operation* “delatebyid”, *element message* “delatebyidRequest dan delatebyidResponse”, *element binding* “WEB SERVICE RS. DOKTER ETTY ASHANTOBinding”, dan *element service* “urn:http://hizbun315.000webhostapp.com/wsperpus/wskategori.php/delatebyid”.

Adapun *source code register function delatebyid* dapat dilihat pada gambar 4.14.

```

1 //delete data pasien
2 $input_delete = array(
3     'ID' => "xsd:int"); // parameter delate
4 $return_delete = array(
5     "return" => "xsd:string");
6 $server->register('deletebyid',
7     $input_delete,
8     $return_delete,
9     $ns,
10    "urn:". $ns."/deletebyid",
11    "rpc",
12    "encoded",
13    "Menghapus Data Pasien Berdasarkan ID");

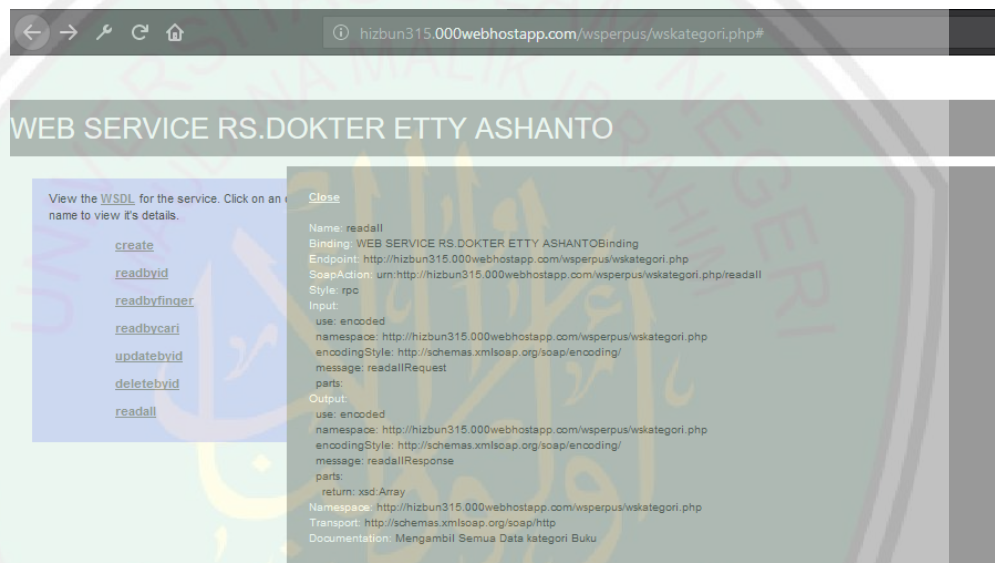
```

14	//end delete
----	--------------

Gambar 4.14 Source code register delatebyid

## 7. Tampilan Hasil Konfigurasi WSDL Readall

Halaman ini merupakan hasil dari konfigurasi *function readall* pada WSDL yang dibuat untuk menangani proses *select* semua data yang ada pada *database server*.

Gambar 4.15 Tampilan hasil WSDL *function readall*

Berdasarkan hasil WSDL *function readall* diatas, terdapat beberapa *element – element* yang terbentuk dari *function* tersebut. Diantaranya: *elemen operation* “readall”, *element message* “readallRequest dan readallResponse”, *element binding* “WEB SERVICE RS. DOKTER ETTY ASHANTO Binding”, dan *element service* “urn:http://hizbun315.000webhostapp.com/wisperpus/wskategori.php/readall”.

Adapun *source code register function readall* dapat dilihat pada gambar 4.16.

1	//Ambil Semua Data pasien
2	\$input_readall = array(); // param ambil data

```

3   $return_readall = array(
4       "return" => "xsd:Array");
5   $server->register('readall',
6   $input_readall,
7   $return_readall,
8   $ns,
9   "urn:". $ns."/readall",
10  "rpc",
11  "encoded",
12  "Mengambil Semua Data Pasien Tanpa
13   Parameter");
14  //End Ambil Semua Data

```

Gambar 4.16 Source code register delatebyid

### 4.2.3 Implementasi Web Client

Pengimplementasian *web client* pendaftaran pasien ini menggunakan *framework codeigniter* dengan model MVC (model, view, controller) dan menggunakan *Apache* sebagai *web server* serta *database* MYSQL. Sebelum menggunakan *codeigniter*, terlebih dulu harus dilakukan pengaturan pada bagian *database*, *config*, dan *autoload*.

Pengaturan pada file *database.php* dilakukan untuk mengatur koneksi antara *web client* dengan *database*. Pengaturan ini dilakukan dengan merubah *hostname*, *username*, *password*, dan nama *database client*.

```

1   $db['default'] = array(
2       'dsn'=> '',
3       'hostname' => 'localhost',
4       'username' => 'root',
5       'password' => '',
6       'database' => 'fingerprint',

```

Gambar 4.17 listing code koneksi ke database fingerprint

Pengaturan pada file *config.php* dilakukan untuk menentukan URL dasar yang digunakan oleh *web client*.

```

1   $http = 'http' . ((isset($_SERVER['HTTPS'])
2   && $_SERVER['HTTPS'] == 'on') ? 's' : '') . '://';

```

```

3 $newurl=str_replace("index.php","",
4 $_SERVER['SCRIPT_NAME']);
5 $config['base_url']="http".$_SERVER['SERVER_NAME']." . "" .
6 $newurl;

```

Gambar 4.18 *listing code config pada web client*

Pengaturan pada file `autoload.php` dilakukan untuk menentukan *library* yang nantinya akan sering digunakan pada *web client*.

```

1 $autoload['libraries']=array('database',
2 'session','form_validation');

```

Gambar 4.19 *listing code autoload pada web client*

### 1. Implementasi *Login*

Halaman login merupakan halaman awal yang akan ditampilkan oleh sistem ketika operator akan masuk ke halaman *administrator*. Proses login berfungsi untuk menghindari adanya penyalahgunaan wewenang dalam mengakses *web client*. Pada saat operator melakukan *login*, maka akan ditampilkan form *login* dari view yang dipanggil melalui controller admin. Berikut merupakan *listing code* controller admin dan tampilan halaman login admin.

```

1 // Check user pada database
2 $this->db->select('*');
3 $this->db->from('users');
4 $this->db->where(array('username' => $username,
5 'password' => $password));
6 $query = $this->db->get();
7 $user = $query->row();

```

Gambar 4.20 *listing code cek user pada controler admin*

Adapun tampilan halaman login admin dapat dilihat pada gambar 4.21.

Gambar 4.21 Halaman login admin

## 2. Implementasi Halaman Registrasi

Halaman ini merupakan halaman registrasi pasien baru. Pada halaman ini terdapat banyak field yang meliputi field mengenai data identitas pasien, alamat lengkap, kerjasama, dan data komunikasi. Pada halaman ini juga terdapat field No. Finger yang merupakan hasil dari pengambilan data sidik jari yang telah di *convert* sebelumnya.

```

1 function registrasi(){
2   $query=$this->db->query("SELECT CONVERT(
3     FingerData, char(255)charset binary) AS
4     fdata_con, CAST(FingerData AS CHAR(255)
5     CHARSET utf8) AS fdata_cast from
6     fingerprint_t");
7   return $query->result();
8 }

```

Gambar 4.22 listing query select convert FingerData

Proses registrasi data pasien baru menggunakan *function createdata* pada *controller welcome*. *Function createdata* melakukan *call function create*

pada *web service* berdasarkan parameter input data pasien. Berikut merupakan *listing code call function create*.

```

1  $result = $client->call("create", array(
2      "dataFingerString" => $this->input->post
3          ('dataFingerString'),
4      "Nama_P" => $this->input->post('Nama_P'),
5      "Jenis_Kel" => $this->input->post('Jenis_Kel'),
6      "Tgl_Lhr" => date('Y-m-d', strtotime( $this->
7          input->post('Tgl_Lhr'))),
8      "Tgl_dft" => $this->input->post=date('Y-m-d'),
9      "Agama" => $this->input->post('Agama'),
10     "PekerjaanP" => $this->input->post('PekerjaanP'),
11     "PendidikanP" => $this->input->post
12         ('PendidikanP'),
13     "Status" => $this->input->post('Status'),
14     "Nama_A" => $this->input->post('Nama_A'),
15     "Nama_I" => $this->input->post('Nama_I'),
16     "Nama_SI" => $this->input->post('Nama_SI'),
17     "Pekerjaan_SI" => $this->input->post
18         ('Pekerjaan_SI'),
19     "Pendidikan_SI" => $this->input->post
20         ('Pendidikan_SI'),
21     "Pekerjaan_A" => $this->input->post
22         ('Pekerjaan_A'),
23     "Pekerjaan_I" => $this->input->post
24         ('Pekerjaan_I'),
25     "Pendidikan_A" => $this->input->post
26         ('Pendidikan_A'),
27     "Pendidikan_I" => $this->input->post
28         ('Pendidikan_I'),
29     "Alamat" => $this->input->post('Alamat'),
30     "desa" => $this->input->post('desa'),
31     "kecamatan" => $this->input->post('kecamatan'),
32     "kabupaten" => $this->input->post('kabupaten'),
33     "provinsi" => $this->input->post('provinsi'),
34     "Negara" => $this->input->post('Negara'),
35     "Nomor_P" => $this->input->post('Nomor_P'),
36     "Kel_P" => $this->input->post('Kel_P'),
37     "Nama_Peru" => $this->input->post('Nama_Peru'),
38     "No_Tlp" => $this->input->post('No_Tlp'),
39     "No_Hp" => $this->input->post('No_Hp'),
40     "No_Fax" => $this->input->post('No_Fax'),
41     "Email" => $this->input->post('Email')
42 ));

```

Gambar 4.23 *Listing code call function create*

Adapun tampilan antar muka halaman registrasi pasien dapat dilihat pada gambar dibawah ini.

Gambar 4.24 Halaman registrasi pasien baru

### 3. Implementasi Halaman Verifikasi

Pada halaman ini hanya terdapat satu field inputan yang otomatis terisi berdasarkan data terakhir pada tabel *web service* kolom FingerData yang telah di *foreign key* dengan kolom FingerData pada tabel fingerprint\_t. Sama halnya dengan data yang sidik jari pada halaman registrasi, data Finger pada halaman ini juga merupakan hasil *convert* dari data sidik jari yang telah tersimpan pada kolom FingerData. Gambar antar muka halaman verifikasi data pasien dapat dilihat pada gambar dibawah ini.

Gambar 4.25 Halaman verifikasi data pasien

Proses verifikasi data pasien berdasarkan data sidik jari dilakukan dengan menggunakan *function* verifikasi yang berada pada *controller welcome*. Pada proses ini *function* melakukan *call function readbyfinger* yang ada pada layanan *web service*. Adapun *listing code call function readbyfingerprint* sebagai berikut.

```

1 | $result = $client->call("readbyfinger", array(
2 |     "dataFingerString" => $this->input->post
3 |     ('dataFingerString')));

```

Gambar 4.26 Listing code call function readbyfinger

#### 4. Implementasi Halaman List Data

Di dalam halaman list data ditampilkan daftar data terbaru yang masuk kedalam *database* server secara *real time*, pada halaman ini juga terdapat icon edit untuk melakukan *update* data dan juga icon *delate* untuk menghapus data.

```

1 | $result = $client->call("readall");

```

Gambar 4.27 Listing code call function readall

Pada icon edit dilakukan pemanggilan *function readbyid* untuk melakukan *select* data pasien berdasarkan nilai ID, data hasil *select* ditampilkan kedalam form edit seperti pada gambar 4.32.

```

1 $result = $client->call("readbyid", array("ID" => $this
   ->uri->segment(3)));

```

Gambar 4.28 Listing code call function readbyid

Pada form edit terdapat button *update* untuk melakukan perubahan atau update data berdasarkan ID yang dikirimkan. Button *update* akan melakukan *call function updatebyid* untuk dapat melakukan *update* data pada *database server*.

```

1 $result = $client->call("updatebyid", array(
2 "ID" => $this->input->post('ID'),
3 "Nama_P" => $this->input->post('Nama_P'),
4 "Jenis_Kel" => $this->input->post('Jenis_Kel'),
5 "Tgl_Lhr" => date('Y-m-d', strtotime( $this->
6 input->post('Tgl_Lhr'))),
7 "Tgl_dft" => $this->input->post=date('Y-m-d'),
8 "Agama" => $this->input->post('Agama'),
9 "PekerjaanP"=> $this->input->post('PekerjaanP'),
10 "PendidikanP" => $this->input->post
11 ('PendidikanP'),
12 "Status" => $this->input->post('Status'),
13 "Nama_A" => $this->input->post('Nama_A'),
14 "Nama_I" => $this->input->post('Nama_I'),
15 "Nama_SI" => $this->input->post('Nama_SI'),
16 "Pekerjaan_SI" => $this->input->post
17 ('Pekerjaan_SI'),
18 "Pendidikan_SI" => $this->input->post
19 ('Pendidikan_SI'),
20 "Pekerjaan_A" => $this->input->post
21 ('Pekerjaan_A'),
22 "Pekerjaan_I" => $this->input->post
23 ('Pekerjaan_I'),
24 "Pendidikan_A" => $this->input->post
25 ('Pendidikan_A'),
26 "Pendidikan_I" => $this->input->post
27 ('Pendidikan_I'),
28 "Negara" => $this->input->post('Negara'),
29 "provinsi" => $this->input->post('provinsi'),
30 "kabupaten" => $this->input->post('kabupaten'),
31 "kecamatan" => $this->input->post('kecamatan'),
32 "kelurahan" => $this->input->post('kelurahan'),
33 "desa" => $this->input->post('desa'),
34 "Alamat" => $this->input->post('Alamat'),
35 "Nomor_P" => $this->input->post('Nomor_P'),
36 "Kel_P" => $this->input->post('Kel_P'),
37 "Nama_Peru" => $this->input->post('Nama_Peru'),
38 "No_Tlp" => $this->input->post('No_Tlp'),

```

```

39 "No_Hp" => $this->input->post('No_Hp'),
40 "No_Fax" => $this->input->post('No_Fax'),
41 "Email" => $this->input->post('Email'));

```

Gambar 4.29 Listing code call function updatebyid

Icon *delete* pada halaman list data akan mengeksekusi *function deletebyid* yang disediakan oleh *web service*. Proses ini akan mengirimkan data ID sebagai parameter input untuk melakukan *delete* data pasien.





```

1 $result = $client->call("deletebyid", array("ID" =>
    $this->input->post('ID')));

```

Gambar 4.30 Listing code call function deletebyid

Gambar antar muka halaman list data pasien dan halaman edit data dapat dilihat pada gambar 4.31 dan gambar 4.32.

No	No Rekam Medis	Nama Pasien	Tanggal Masuk	Alamat	Action
1	86	hizbun	2017-12-12		 
2	84	jaksja	2017-12-12		 

Gambar 4.31 Halaman list data

**Form Input Data**

No. Rekam Medis:

Nama Pasien:

Jenis Kelamin:  Pria  Wanita

Tanggal Lahir:

Agama:

Pekerjaan Pasien:

Pendidikan Pasien:

Status Pernikahan:

Nama Ayah:

Nama Ibu:

Nama Suami/Istri:

Pekerjaan Suami/Istri:

Pendidikan Suami/Istri:

Pekerjaan Ayah:

Pekerjaan Ibu:

Pendidikan Ayah:

Pendidikan Ibu:

Alamat:

Provinsi:

Kabupaten/Kota:

Kecamatan:

Desa:

Nomor Pasien:

Kelompok Pasien:

Nama Perusahaan:

No. Telephone:

No. Handphone:

No. Fax:

E-Mail:

Gambar 4.32 Halaman edit data pasien

#### 4.2.4 Implementasi Aplikasi Finger

Pada aplikasi finger dilakukan input citra sidik jari sebanyak empat kali inputan sebagai data *semple* sidik jari, citra sidik jari yang telah terbaca oleh scanner akan disimpan kedalam *database* dengan data *type varbinary*. Adapun *listing code* yang digunakan adalah seperti pada gambar 4.33

```

1 Private Sub Form_Load()
2 Call SetKoneksi
3 Set RecordSet = New RecordSet
4
5 SQL = "SELECT ID FROM fingerprint_t ORDER BY ID
6     DESC LIMIT 1"
7 RecordSet.Open SQL, Koneksi, adOpenStatic, adLockReadOnly
8 Call tampil
9 If FPDatabase.ActiveConnection("DSN=webservice;
10     SERVER=localhost;UID=root;DATABASE=fingerprint;
11     PORT=3306") = sc_Success Then
12 Set Registration = New FingerspotETouchSDK.
13     FinFPRegistration
14     Registration.PictureSamplePath = App.Path &
15     "\FPTemp.BMP"
16     Registration.PictureSampleHeight = sidik.Height
17     Registration.PictureSampleWidth = sidik.Width
18 Else
19 MsgBox "Database Connection Fail"
20 End If
21 End Sub

```

Gambar 4.33 *listing code load* citra untuk proses registrasi

Pada form verifikasiP dilakukan satu kali inputan citra sidik jari sebagai parameter pencocokan citra dengan nilai citra yang telah tersimpan di dalam *database*. Setelah data citra sidik jari ditemukan, akan ditampilkan ID citra yang nantinya akan diinputkan kembali kedalam tabel webservice sebagai paramater input untuk dilakukan fungsi *foreign key* dengan kolom FingerData pada tabel fingerprint\_t. Berikut merupakan *listing code load* citra sidik jari pada form verifikasiP.

```

1 Private Sub Form_Load()
2 On Error GoTo ErrMsg

```

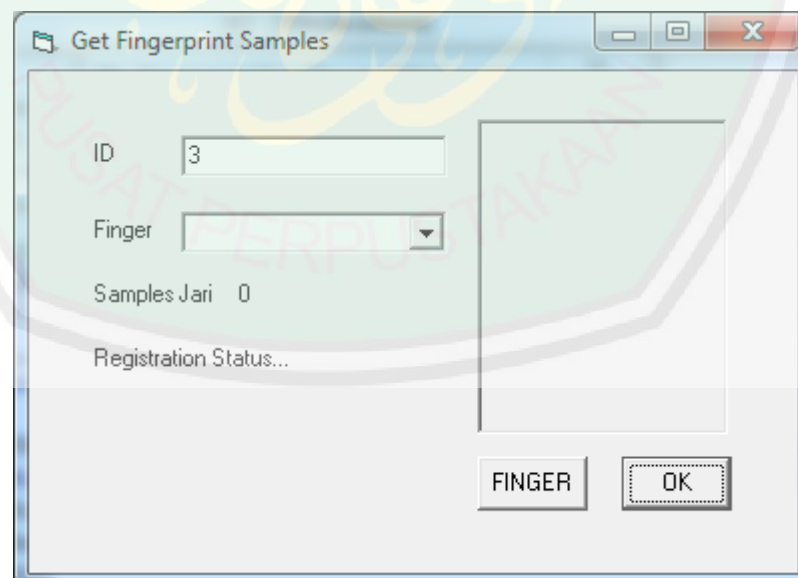
```

3  If FPDatabase.ActiveConnection("DSN=webservice;SERVER=
4  localhost;UID=root;DATABASE=fingerprint;PORT=3306") =
5  sc_Success Then
6      Set Verification = New FingerspotETouchSDK.
7      FinFPVerification
8      Verification.PictureSamplePath = App.Path &
9      "\FPTemp.BMP"
10     Verification.PictureSampleHeight = sidik.Height
11     Verification.PictureSampleWidth = sidik.Width
12     Verification.FPVerification
13 Else
14     MsgBox "Databse Connecion Fail"
15 End If
16 Exit Sub
17 ErrMsg:
18     If Err.Number = 430 Then
19         MsgBox "You need to recompile your aplication to
20         use the latest SDK version"
21     Else
22         MsgBox Err.Description, , Err.Number
23     End If
24 End Sub

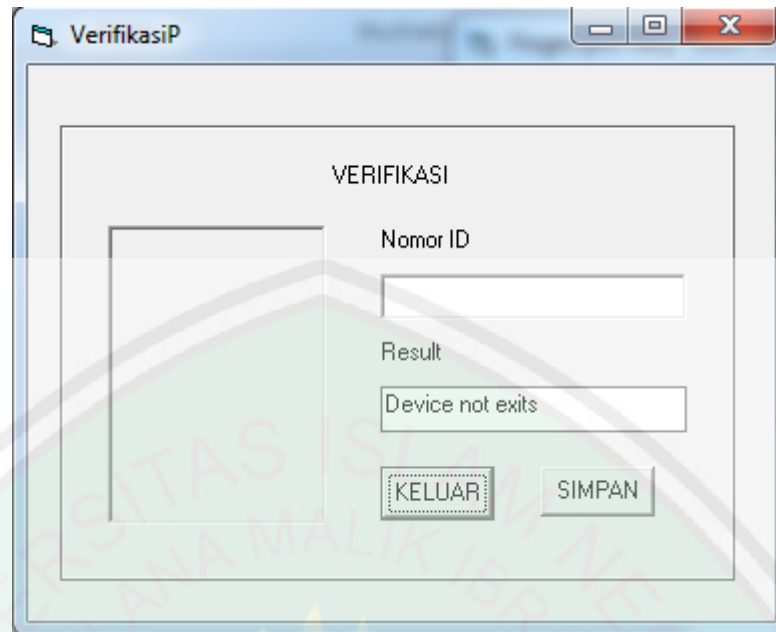
```

Gambar 4.34 Listing code load citra untuk proses verifikasi

Berikut merupakan *interface* untuk proses registrasi citra sidik jari dan verifikasi citra sidik jari.



Gambar 4.35 Interface registrasi citra sidik jari



Gambar 4.36 *Interface* verifikasi citra sidik jari

### 4.3 Pengujian Sistem

Dalam melakukan pengujian system ini digunakan metode *blackbox*, yang memfokuskan pada keperluan fungsional dari sistem, kemudian dilihat hasil yang diperoleh sesuai dengan yang diharapkan atau tidak. Berikut hasil pengujian terhadap Rancang Bangun Sistem Pelayanan Pasien Rumah Sakit Berbasis *Web Service* dengan Protokol SOAP WSDL dan Bantuan Alat Deteksi Sidik Jari.

Tabel 4.1 Prosedur Pengujian sistem

Nama Use Case	Kriteria Evaluasi Hasil
<b>Login</b>	Sistem memeriksa apakah <i>username</i> dan <i>password</i> sudah benar.
<b>Registrasi Citra Sidik Jari</b>	Nilai <i>biner</i> citra sidik jari pasien tersimpan di dalam <i>database</i> .
<b>Verifikasi Citra Sidik Jari</b>	Sistem menampilkan ID yang cocok dengan data citra sidik jari inputan.

<b>Registrasi Data Pasien</b>	Data pasien baru ditampilkan pada halaman list data.
<b>Verifikasi Data Pasien</b>	Tampil data pasien berdasarkan paramater dataFingerString yang sesuai.
<b>Update Data Pasien</b>	Data pasien yang telah tersimpan berubah sesuai dengan perubahan data yang dilakukan
<b>Delate Data Pasien</b>	Data yang dipilih terhapus dari halaman list data.
<b>Cari Data Pasien</b>	Tampil data yang sesuai dengan nama pasien yang di munculkan.

Skenario pengujian ini dilakukan oleh pengguna dengan mengakses sistem melalui *web server (localhost)*. Setelah pengujian selesai dilakukan, pengguna mengisi kuesioner yang telah diberikan berdasarkan hasil pengujiannya. Pengujian terdiri dari pengujian fungsionalitas dan antar muka sistem. Pengujian ini dilakukan dengan melibatkan 10 responden dari kalangan mahasiswa dan pekerja. Berikut hasil pengujian fungsionalitas sistem seperti yang terlihat pada tabel 4.2 dan antar muka sistem pada tabel 4.3.

Tabel 4.2 Tabel pengujian fungsionalitas sistem

No	Fungsional	SS	S	TS	STS
1	Sistem memeriksa apakah <i>username</i> dan <i>password</i> sudah benar.	7	3	-	-
2	Nilai <i>biner</i> citra sidik jari pasien tersimpan di dalam <i>database</i> .	4	6	-	-
3	Sistem menampilkan ID yang cocok dengan data citra sidik jari inputan.	3	7	-	-
4	Data pasien baru ditampilkan pada halaman list data.	5	5	-	-

5	Tampil data pasien berdasarkan parameter dataFingerString yang sesuai.	4	6	-	-
6	Data pasien yang telah tersimpan berubah sesuai dengan perubahan data yang dilakukan	7	3	-	-
7	Data yang dipilih terhapus dari halaman list data.	7	3	-	-
8	Tampil data yang sesuai dengan nama pasien yang di munculkan.		5	5	-
<b>Total</b>		37	38	5	0

Tabel 4.3 Tabel pengujian antarmuka

No	Pertanyaan	SS	S	TS	STS
1	Apakah layout antarmuka sudah <i>user friendly</i> sehingga mempermudah anda dalam menggunakan sistem?	1	7	2	-
2	Apakah antarmuka sistem sudah memenuhi unsur kompatibilitas?	-	6	4	-
3	Apakah antarmuka sistem sudah familiar bagi anda?	3	7	-	-
4	Apakah sistem sudah cukup responsive?	2	8	-	-
<b>Total</b>		6	28	6	0

Berdasarkan hasil pengujian sistem yang ditunjukkan pada tabel 4.2 dan 4.3, digunakan rumus perhitungan presentase hasil kuesioner menurut Sugiyono (2008),

$$p = f/n \times 100$$

keterangan:

p= prosentase

f= frekuensi dari setiap jawaban angket

n=jumlah responden

Sehingga dari tabel 4.2 dapat dilakukan perhitungan sebagai berikut:

$$p=37/80 \times 100 = 46.25\%$$

$$p=38/80 \times 100 = 47.5\%$$

$$p=5/80 \times 100 = 6.25\%$$

dan perhitungan yang dilakukan pada tabel 4.3 adalah:

$$p=6/40 \times 100 = 15\%$$

$$p=28/40 \times 100 = 70\%$$

$$p=6/40 \times 100 = 15\%$$

dari hasil perhitungan pengujian fungsionalitas sistem diatas memperlihatkan bahwa 46.25% responden sangat setuju dan 47.5% responden setuju serta 6.25% responden tidak setuju. Adapun dalam pengujian antarmuka sistem berdasarkan tabel 4.3 menunjukkan bahwa 15% responden sangat setuju bahwa antar muka sistem sudah cukup baik, dan 70% menyatakan setuju sedangkan 15% sisanya menyatakan tidak setuju.

Karena keterbatasan waktu dan kondisi sistem yang digunakan saat ini terus berjalan sehingga tidak memungkinkan untuk dilakukan pengujian langsung di lapangan, maka dilakukan pengamatan pada proses pendaftaran yang sedang berjalan dan simulasi pengujian pada sistem yang sedang dikembangkan untuk mengetahui lama waktu yang dibutuhkan dalam serangkaian proses yang ditentukan menggunakan data pasien yang sama. Pengamatan dan simulasi sistem dilakukan pada tujuh pasien seperti yang digambarkan pada tabel 4.4 dan 4.5 dimana P1 merupakan pasien pertama dan seterusnya.

Tabel 4.4 Hasil pengamatan pada sistem yang berjalan

No	Pasien	Lama Waktu Pasien Baru (menit)	Lama Waktu Pasien Lama (menit)
1	P1	03.21	00.15
2	P2	03.19	00.17
3	P3	03.00	00.25
4	P4	03.05	00.15
5	P5	03.06	00.19
6	P6	03.11	00.15
7	P7	03.21	00.16
Total		22.23	02.02

Tabel 4.5 Hasil simulasi sistem yang dikembangkan

No	Pasien	Lama Waktu Pasien Baru (menit)	Lama Waktu Pasien Lama (menit)
1	P1	02.50	00.10
2	P2	03.00	00.11
3	P3	02.34	00.11
4	P4	02.40	00.09
5	P5	02.50	00.12
6	P6	02.51	00.11
7	P7	03.02	00.09
Total		19.47	1.13

Dari tabel 4.4 dan 4.5 dilakukan perhitungan rata-rata untuk mengetahui waktu rata-rata yang diperlukan dalam melakukan pendaftaran pasien baru dan pasien lama dengan menggunakan satuan waktu second dan persamaan menurut Arikunto (2010).

$$\bar{X} = \frac{\sum X}{\sum N}$$

Keterangan:

$\bar{X}$  = Nilai rata - rata

$\sum X$  = Total waktu

$\sum N$  = Jumlah pasien

Sehingga dari tabel 4.4 diperoleh rata – rata lama waktu pendaftaran pasien baru adalah 03.11 menit dan rata – rata lama waktu pendaftaran pasien lama adalah 00.17 menit, sedangkan dari tabel 4.5 diperoleh rata – rata lama waktu pendaftaran pasien baru pada pengujian simulasi sistem yang dikembangkan adalah 02.49 menit dan rata – rata lama waktu pasien lama adalah 00.10 menit. Berdasarkan nilai rata – rata lama waktu pendaftaran pasien baru dan pasien lama dari masing – masing tabel maka diperoleh selisih waktu antara sistem yang sedang berjalan dengan sistem yang di kembangkan selama 00.22 menit untuk pendaftaran pasien lama dan 00.07 menit untuk pendaftaran pasien lama.

## BAB V

### PENUTUP

Pada bab penutup ini akan dibahas tentang kesimpulan - kesimpulan dari hasil penelitian dan saran- saran yang diberikan untuk pengembangan sistem.

#### 5.1 Kesimpulan

Berdasarkan hasil pengamatan pada proses perancangan, implementasi dan pengujian sistem dapat diambil beberapa kesimpulan, diantaranya adalah:

1. Perancangan sistem pendaftaran pasien berbasis *biometric* dapat meningkatkan pelayanan pendaftaran pasien karena tidak ada lagi kemungkinan pasien lupa membawa kartu pasien, kartu pasien rusak atau hilang dan pencetakan kartu pasien yang mengakibatkan proses pendaftaran menjadi lebih lama dan adanya tambahan biaya pembuatan kartu pasien. Sehingga dari pengujian yang dilakukan di peroleh kesimpulan bahwa aplikasi yang di kembangkan lebih cepat 00.22 menit pada pendaftaran pasien baru dan 00.07 menit pada pendaftaran pasien lama.
2. Perancangan distribusi data dengan menggunakan teknologi *web service* dapat meningkatkan efektivitas pelayanan karena integrasi sistem antara RS. Dokter Etty dan cabang RS. Dokter Etty yang akan dikembangkan dapat dijalankan dengan baik dan dapat menekan biaya dalam pengaplikasiannya, selain itu penggunaan teknologi *web service* dapat mempermudah *maintenance* server rumah sakit karena tidak bergantung pada *hosting* tertentu.

## 5.2 Saran

Sistem pendaftaran pasien yang telah dibangun ini tentunya tidaklah luput dari kesalahan dan kekurangan. Oleh karena itu penulis menyertakan beberapa saran untuk dapat pengembangan sistem yang lebih baik dikemudian hari, diantaranya

1. Sistem bisa dikembangkan lagi dengan menggunakan flexcode 4500 SDK yang telah support dengan bahasa pemrograman PHP. Sehingga sistem dapat lebih efektif dan efisien.
2. Sistem perlu dikembangkan lagi dalam hal *interface* agar menjadi lebih menarik dan elegan.
3. Perlu dilakukan penelitian lebih lanjut untuk pengembangan sistem agar dapat menjadi sebuah sistem rumah sakit yang utuh, sehingga tak hanya menjadi alternatif solusi dalam bidang pendaftaran tetapi juga dapat menjadi alternatif solusi dalam penentuan penanganan, biaya dan lain sebagainya.
4. Sistem dapat dikembangkan lagi dengan model EAI ( Enterprise Application Integration) dalam hal mengatur pola integrasi dengan banyak layanan poli.

## DAFTAR PUSTAKA

- Abu Yahya Marwan bin Musa. Tafsir Al – Qur'an Hidayatul Insan.  
www.tafsir.web.id. Diakses pada sabtu 07 Oktober 2017 jam 08.41 PM
- Adi Irawan, Indah Fitri Astuti, M.Cs, Dedy Cahyadi, M.Eng. 2015. Presensi Sidik Jari (*Fingerprin*) Berbasis *Web Service*. Universitas Mulawarman
- Adinoto, Trio Suryaning. 2013. Perancangan Absensi Karyawan SMP Negeri 1 Kramat Tegal. Universitas Dian Nuswantoro
- Aditama, Roki. 2017. *Web Service* Pembayaran Uang Kuliah *Online* dengan PHP dan SOAP WSDL. Penerbit: CV. Lokomedia
- Arif Adi, Riyanto. 2013. Pemanfaatan *Web Service* Sebagai Integrasi Data Farmasi di RSUD Banyumas. STMIK AMIKOM Purwokerto.
- Arikunto. 2010. *Prosedur Penelitian: Suatu Pendekatan Praktek*. Jakarta: Rineka Cipta
- Chairunnisa, Maya Puspita, 2017. Gambaran Kepuasan Pasien Rawat Jalan Terhadap Pelayanan di Rumah Sakit Islam Jakarta Sukapura (RSIJS) Tahun 2015. Penerbit: Universitas Muhammadiyah Jakarta
- Dahlan Abdullah, Iswandi, 2015. Perancangan Sistem Pendaftaran *Online* Pasien pada Klinik dengan *Metode* FIFO Berbasis *Web Service*. Penerbit: Universitas Malikussaleh
- Danny Indrawan, Arif Kurniadi, Retno Astuti S. 2013. Rancangan Sistem Informasi Pelayanan Pendaftaran Pasien Rawat Jalan Berbasis Web di Puskesmas Mranggen III Demak Tahun 2013. Penerbit: Universitas Dian Nuswantoro

- Deviana, Hartati. 2011. Penerapan XML *Web Service* pada Sistem Distribusi Barang. Politeknik Negeri Sriwijaya
- Devy Haryadi, Solikhah. 2013. Evaluasi Sistem Informasi Pendaftaran Pasien Rawat Jalan di Rumah Sakit Umum PKU Muhammadiyah Bantul. Universitas Ahmad Dahlan Yogyakarta
- Edhy Sutanta, Khabib Mustofa. 2012. Kebutuhan *Web Service* untuk Sinkronisasi Data antar Sistem Informasi dalam E-Gov di Pemkab Bantul Yogyakarta. Universitas Gajah Mada
- Frieyadie. 2010. Mudah Belajar Pemrograman *Database* MYSQL dengan *Microsoft Visual Basic* 6.0. Penerbit: Andi
- Iqbal Firdaus, Dhanang Sukmana Adi, Fredi Aji Noorhadi. 2014. Rancang Bangun E-Resto Menggunakan *Web Service* untuk Restoran *Family* di Kotamadya Surakarta. Politeknik Indonusa Surakarta
- Komputer, Wahana. 2010. Membuat Aplikasi *Client Server* dengan *Visual Basic* 2008. Penerbit: Andi
- Kurniawan, Erick. 2014. Implementasi *Rest Web Service* untuk *Sales Order* dan *Sales Tracking* Berbasis Mobile. Jurnal EKSIS
- M. Abdul Ghoftar E.M, Abdurrahim Mu'thi. Abu Ihsan Al-Atsari. 2004. Tafsir Ibnu Katsir. Penerbit: Pustaka Imam Asy-Syafi'i.
- M. Sidi Mustaqobal, Roeri Fajri Firdaus, Hendra Rahmadi. 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. Penerbit: Universitas Widyatama
- Madiun, Madcoms. 2008 PHP dan MySQL untuk Pemula. Penerbit: Andi

- Muhammad Aminudin Rahman, Imam Kuswardayan, Ridho Rahman Hariadi. 2013. Perancangan dan Implementasi RESTful *Web Service* untuk Game Sosial *Food Merchant Saga* pada Perangkat Android. Institut Teknologi Sepuluh November
- Neti M. Bustani, A. Joy Rattu, Josephine S. M. Saerang. 2015. Analisis Lama Waktu Tunggu Pelayanan Pasien Rawat Jalan di Balai Kesehatan Mata Masyarakat Provinsi Sulawesi Utara. Penerbit: Pasca Sarjana Universitas Sam Ratulangi
- Presman, Roger S. 2001. *Software Engineering*. <http://www.mhhe.com>
- Rangkuti, Atika Sahroni. 2016. Gambaran Pelaksanaan Pelayanan Penerimaan Pasien Rawat Jalan Terhadap Kelengkapan Pengisian Berkas Rekam Medis di RSUD Imelda Pekerja Indonesia Medan Pada Tahun 2016. Akademi Perkam Medis dan Informasi Kesehatan Imelda Medan
- Riyanto. 2003. Mengakses database menggunakan ODBC. Penerbit: PT Elex Media Komputindo
- Ryab Wahyudi, Oni Soesanto, Muliadi. 2015. Rancang Bangun Aplikasi Pengenalan Pola Sidik Jari. Prodi Ilmu Komputer FMIPA UNLAM
- Santosa, Budi. 2008. Analisa dan Perancangan *Web Service* untuk Sistem Informasi Universitas. Jurusan Teknik Informatika UPN Veteran Yogyakarta
- Sigit Surendra, Martinus Raditia. 2014. Implementasi PHP *Web Service* Sebagai Penyedia Data Aplikasi Mobile. Universitas Multimedia Nusantara
- Siswotomo, Wiwit. 2004. Membangun *Web Service Open Source* Menggunakan PHP. Penerbit: PT Elex Media Komputindo

Sugiyono, 2008. Metode Penelitian Kuantitatif Kualitatif dan R&D. Penerbit:  
Alfabeta, CV Bandung

