

## Pengembangan Aplikasi Puskesmas Untuk Pasien Umum Menggunakan Pendekatan Web Semantik (Studi Kasus UPT Puskesmas Bareng Kota Malang)

Gerry Giovanni Benson<sup>1</sup>, Bayu Priyambadha<sup>2</sup>, Djoko Pramono<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>gerrygiovanni@gmail.com, <sup>2</sup>bayu\_priyambadha@ub.ac.id, <sup>3</sup>djoko.jalin@ub.ac.id

### Abstrak

Puskesmas merupakan salah satu sarana yang bekerjasama dengan pemerintah dalam rangka pembangunan di bidang kesehatan. Pelayanan di Puskesmas Bareng yang berlokasi di Kecamatan Klojen Kelurahan Bareng Kota Malang dibagi menjadi dua jenis pasien yaitu jenis pasien BPJS dan jenis pasien Umum. Saat ini Puskesmas Bareng masih menggunakan mekanisme penyimpanan data manual untuk jenis pasien Umum. Hal ini menimbulkan berbagai kendala seperti kesulitan dalam mencari berkas pasien yang tersimpan di gudang arsip, berkas pasien dapat hilang atau rusak, terduplikasi, dan kesalahan manusia. Dari permasalahan tersebut, diperlukan suatu pengembangan sebuah aplikasi sistem informasi yang dapat menangani penyimpanan dan pengolahan data secara otomatis sehingga memudahkan kinerja dari petugas puskesmas. Penelitian ini dilakukan menggunakan metode *Waterfall System Development Life Cycle* dan pendekatan web semantik. Sistem ini dibangun berbasis website menggunakan teknologi PHP, Apache Jena Fuseki, MySQL, dan Javascript. Penelitian ini menggunakan pengujian *compatibility* dengan hasil baik di peramban *Mozilla Firefox* dan *Google Chrome* kemudian pengujian *white-box* mendapatkan hasil 100% valid dari 9 kasus uji, pengujian *black-box* didapatkan hasil 100% valid dari 37 kasus uji dan pengujian *recall* pada semantik didapatkan hasil 100% dari 5 kasus uji dan pengujian *precision* didapatkan hasil 90% dari 5 kasus uji.

**Kata kunci:** *puskesmas, sistem informasi, php, web semantik.*

### Abstract

*The Community Health Center or Health Center is one of the means which cooperate with the government in the development of the health sector. Service in Bareng Health Center located in the Bareng, Klojen district, Malang City is divided into two types of patients that are the type of patient BPJS and the type of general patient. This causes a variety of problems such as difficulty in finding patient files stored in archives, patient file which can be lost or damaged, duplicated, and human errors. Of these problems, we need a development of an information system applications that can handle data storage and processing automatically so as to facilitate the performance of Health Center. This research was conducted using Waterfall System Development Life Cycle method and using semantic web approach. The system was built based website using PHP technology, Apache Fuseki Jena, MySQL, and Javascript. This study uses a compatibility testing that showed good results in the Mozilla Firefox and Google Chrome browser and the white-box testing showed results of 100% valid from 9 test cases and black-box testing showed a 100% valid from 37 test cases and recall testing of semantics showed 100% of 5 test cases, while the precision testing showed 90% of 5 test cases.*

**Keywords:** *community health centers, information system, php, semantic web.*

### 1. PENDAHULUAN

Pusat Kesehatan Masyarakat atau Puskesmas merupakan salah satu sarana yang bekerjasama dengan pemerintah dalam rangka pembangunan di bidang kesehatan. Berdasarkan

Peraturan Menteri Kesehatan Nomor 75 Tahun 2014, Puskesmas memiliki peran melaksanakan kebijakan kesehatan untuk mencapai tujuan pembangunan kesehatan di wilayah kerjanya dalam rangka mendukung terwujudnya Kecamatan Sehat. Puskesmas memiliki beberapa

fungsi, diantaranya sebagai penggerak pembangunan berwawasan kesehatan, pemberdayaan masyarakat untuk hidup sehat dengan mandiri dan pelayanan kesehatan pertama (Kemenkes, 2015).

Puskesmas Bareng merupakan bagian dari Fasilitas Kesehatan Tingkat Satu sesuai dengan pelaksanaan program BPJS kesehatan tahun 2014 (BPJS, 2015). Masyarakat dapat menggunakan program BPJS sebagai jaminan kesehatan untuk berobat di puskesmas. Program BPJS Kesehatan di Puskesmas menggunakan aplikasi *Primary Care* atau disingkat dengan PCARE BPJS yang berbasis website sebagai penyimpanan data untuk pasien BPJS yang disimpan di dalam penyimpanan data BPJS secara daring sedangkan pelayanan untuk pasien non-BPJS atau disebut dengan pasien jenis Umum menggunakan penyimpanan kertas sebagai pencatatan data pasien dan data pemeriksaan pasien. Saat ini pelayanan untuk pasien umum di Puskesmas Bareng masih menggunakan mekanisme penyimpanan data manual. Dengan menggunakan penyimpanan yang masih manual pada pasien jenis Umum, dapat menimbulkan berbagai kendala seperti kesulitan dalam mencari berkas pasien yang tersimpan di gudang arsip, data berkas pasien yang dapat rusak karena faktor lingkungan dan waktu, data berkas pasien yang terduplikasi akibat data lama pasien yang tidak ditemukan, riwayat medis pasien yang tidak lengkap akibat faktor kesalahan manusia yaitu kehilangan berkas riwayat medis. Tenaga medis atau dokter kesulitan untuk mencari informasi riwayat medis pasien karena berkas riwayat medis pasien yang masih manual. Pencarian data anamnesa atau data diagnosa di rekam medis pasien jika menggunakan database relational akan tidak efisien karena dokter harus mengisi formulir pencarian yang berbeda ketika mencari data anamnesa atau diagnosa.

Salah satu upaya untuk mengurangi permasalahan dalam pelayanan pasien jenis Umum di Puskesmas Bareng yaitu dengan cara menerapkan aplikasi puskesmas untuk pasien jenis umum berbasis website untuk membantu dalam penyimpanan dan pengolahan data. Selain itu dengan menggunakan pendekatan web semantik, aplikasi puskesmas diharapkan dapat memberikan referensi suatu informasi riwayat dan cara pengobatan pasien yang pernah mengalami penyakit yang sama ketika tenaga medis mendiagnosa penyakit seorang pasien sehingga membantu tenaga medis dalam

memberi tindakan terhadap pasien tersebut. Berdasarkan pada permasalahan yang diuraikan pada latar belakang diatas, peneliti tertarik melakukan penelitian tentang “Pengembangan Aplikasi Puskesmas Untuk Pasien Umum Menggunakan Pendekatan Web Semantik”.

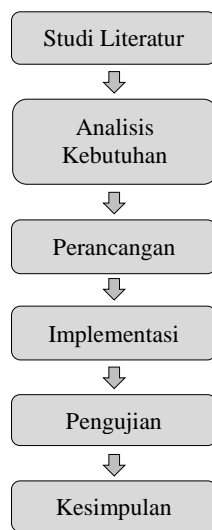
Dalam penelitian sebelumnya, pendekatan web semantik digunakan oleh Verma (2015) sebagai perhitungan peringkat halaman website dengan jenis *e-commerce* pada website – website di negara India. Penelitian tersebut, dalam proses umpan balik user secara eksplisit dan implisit, semantik digunakan untuk mengidentifikasi *user session*, dan membandingkan profil pengguna menggunakan LCS.

Model *Waterfall* dirancang oleh Winston W. Royce pada tahun 1970. Dalam model ini seluruh pekerjaan dilakukan dalam mode linear. Seluruh pekerjaan dibagi menjadi lima fase yang berbeda. Semua tahapan yang mengalir dan berhubungan satu sama lain sehingga ketika tahap kedua dimulai dan menetapkan tujuan berdasarkan hasil dari tahap pertama. *Waterfall* juga memungkinkan untuk dilakukan *review* terhadap tahapan yang telah dilakukan (Kumar et. al., 2013).

Web semantik secara sederhana merupakan sebuah website dengan data yang dideskripsikan dan saling dihubungkan dengan tujuan membentuk suatu konteks atau semantik. Web semantik merupakan salah satu dasar dari Web 3.0. Perbedaan besar antara web semantik dengan Web 2.0 adalah dari segi pemahaman suatu data atau informasi. Informasi atau konten di Web 2.0 dapat dipahami oleh manusia, karena informasi atau konten tersebut dibuat oleh manusia dan untuk manusia. Tetapi informasi tersebut tidak dapat dibaca ataupun dipahami bagi sebuah mesin (Hebeler, 2009).

## 2. METODOLOGI

Pada bagian ini menjelaskan prosedur – prosedur yang dilakukan dalam penelitian. Diawali dengan studi literatur, analisis kebutuhan, perancangan, implementasi, pengujian dan evaluasi.



Gambar 1. Metodologi Penelitian

**2.1 Studi Literatur**

Pada tahap ini dilakukan studi literatur atau pemahaman terhadap berbagai hal yang berhubungan dengan analisa dan perancangan aplikasi dengan cara mengumpulkan, membaca dan memahami berbagai buku yang berkaitan dengan pengolahan dan pemrosesan data dalam komputer serta website serta program aplikasi yang dibutuhkan sehingga diharapkan mampu menyelesaikan permasalahan yang timbul. Pada penelitian ini, studi literatur yang digunakan yaitu:

- a. Konsep pencatatan dan penyimpanan data pasien jenis umum di Puskesmas Bareng
- b. Konsep web semantik
- c. *Unified modelling Language (UML)*
- d. Konsep pengujian
  - Pengujian *white-box*
  - Pengujian *black-box*
  - Pengujian *Browser Compatibility*
  - Pengujian *Recall* dan *Precision*

**2.2 Analisis Kebutuhan**

Analisis kebutuhan bertujuan untuk memperoleh semua kebutuhan yang diperlukan oleh pengguna aplikasi puskesmas untuk pasien umum. Tahap yang dilakukan dimulai dari wawancara kepada kepala UPT Puskesmas Bareng, kemudian dilanjutkan dengan melakukan observasi dan analisis terkait proses bisnis pada proses pengambilan data, pemrosesan data serta penyimpanan data di unit loket, poli, laboratorium, dan apotek dalam Puskesmas Bareng.

**2.3 Perancangan Sistem**

Tahap perancangan akan dilakukan setelah proses analisis kebutuhan telah selesai dan hasil dari analisis kebutuhan tersebut dapat digunakan dalam merancang sistem. Perancangan sistem terdiri dari:

1. Perancangan *class diagram* yang bertujuan menggambarkan peta class dan entitas yang berhubungan pada sistem.
2. Perancangan *sequence diagram* yang bertujuan menggambarkan seluruh proses dan fungsi – fungsi yang berjalan pada sistem.
3. Perancangan basis data atau *database* yang bertujuan menggambarkan entitas dan relasi penyimpanan data pada sistem.
4. Perancangan *ontology* yang bertujuan menggambarkan triple untuk menyimpan data semantik pada sistem.

**2.4 Implementasi**

Tahap implementasi sistem dilakukan setelah selesai melakukan tahap perancangan dan pengumpulan data. Pembuatan sistem ini menggunakan bahasa pemrograman PHP, menggunakan MySQL sebagai manajemen basis data dan menggunakan Apache Jena Fuseki sebagai web API semantik.

**2.5 Pengujian**

Pengujian sistem dilakukan dengan tujuan mengetahui apakah sistem yang telah dibangun dapat berjalan sesuai dengan yang diharapkan. Dengan melakukan pengujian terhadap sistem, peneliti dapat mengetahui kekurangan dan kesalahan dari sistem agar dapat dilakukan perbaikan sehingga sistem yang dihasilkan mampu memenuhi kebutuhan pengguna. Pengujian sistem ini dilakukan dengan menggunakan metode pengujian *white-box*, pengujian *black-box*, pengujian *compatibility browser*, serta pengujian *recall* dan *precision*.

**2.6 Kesimpulan**

Kesimpulan yang diperoleh dapat diambil berdasarkan hasil pengolahan dan analisis data yang telah dilakukan sebelumnya sehingga peneliti dapat menarik kesimpulan terkait perancangan sistem yang telah dibuat dan hasil pengujian dari sistem tersebut.

### 3. PERANCANGAN

#### 3.1 Identifikasi Ruang Lingkup

Deskripsi ruang lingkup merupakan tahap awal yang ada pada Puskesmas Bareng. Tahap ini juga menjadi landasan untuk tahapan selanjutnya.

##### 3.1.1 Ruang Lingkup Proyek

Ruang lingkup didasarkan pada proses pelayanan di Puskesmas Bareng, meliputi pelayanan loket, pelayanan dokter, pelayanan apotek dan pelayanan laboratorium.

#### 3.2 Analisis Kebutuhan

Proses analisis kebutuhan dimulai dengan gambaran umum aplikasi, penjelasan tentang daftar kebutuhan, kemudian digambarkan dalam bentuk model diagram. Analisis kebutuhan bertujuan untuk menjabarkan kebutuhan – kebutuhan yang disediakan sistem untuk memenuhi kebutuhan pengguna.

##### 3.2.1 Gambaran Fitur Pilihan

Dalam penelitian ini, aplikasi puskesmas untuk pasien umum yang dibuat mampu memberikan berbagai solusi yaitu sistem yang terkomputerisasi, dapat mempersingkat waktu pelayanan dan meningkatkan efisiensi waktu, memudahkan tenaga medis dalam hal memeriksa dan mendiagnosa pasien. Dalam sistem ini juga dapat mengolah keluaran data hasil pemeriksaan pasien kedalam bentuk berkas dengan menggunakan pendekatan web semantik bertujuan untuk memberikan referensi berupa data hasil pemeriksaan pasien dengan format baku agar dapat digunakan oleh penelitian atau aplikasi lain yang membutuhkan.

##### 3.2.2 Identifikasi Aktor

Pada sistem ini terdapat 4 (empat) pengguna, masing-masing pengguna mempunyai hak akses yang berbeda. Hak akses tersebut ditentukan berdasarkan peran pengguna tersebut, peran pengguna adalah sebagai berikut:

1. Petugas loket: Petugas yang bertugas untuk menerima pasien yang datang, melakukan pencatatan data pasien dan melayani pasien ke unit yang diinginkan.
2. Dokter: Dokter yang bertugas untuk melayani pasien, memeriksa pasien dan memberi diagnosa kepada pasien. Dokter juga dapat memberi rujukan pasien ke laboratorium.
3. Petugas Laboratorium: Petugas yang bertugas melayani pasien dengan melakukan pemeriksaan laboratorium sesuai rujukan yang

diberikan dokter.

4. Petugas Apotek: Petugas Apotek bertugas melayani pasien dengan memberikan obat sesuai dengan obat yang diberikan oleh dokter.

##### 3.2.3 Kebutuhan Fungsional

Pada aplikasi puskesmas untuk pasien umum, kebutuhan utama dijabarkan pada Tabel 1.

Tabel 1. Kebutuhan fungsional

ID	Fungsi	Kebutuhan
F02.1	Menambah pasien baru	Aplikasi dapat mencatat informasi identitas pasien yang berobat di puskesmas
F03.6	Melihat riwayat hasil tindakan medis yang relevan	Aplikasi dapat menampilkan data riwayat hasil tindakan medis dari web API semantik yang berdasarkan kata kunci yang dimasukkan oleh dokter
F03.7	Menyimpan hasil pemeriksaan dengan format RDF	Aplikasi dapat menyimpan hasil pemeriksaan pasien ke dalam aplikasi dengan format RDF (Resource Description Framework)

Tabel 1 merupakan sebagian dari keseluruhan kebutuhan fungsional. Kebutuhan fungsional merupakan kebutuhan utama yang diperlukan dalam suatu sistem.

##### 3.2.4 Kebutuhan Non Fungsional

Kebutuhan non fungsional yang digunakan pada penelitian dijelaskan pada Tabel 2.

Tabel 2. Kebutuhan non fungsional

Kode	Spesifikasi	Deskripsi
NF01	Compatibility Browser Mozilla Firefox	Sistem dapat berjalan dengan baik pada browser Mozilla Firefox dengan versi 56.0.2
NF02	Compatibility Browser Google Chrome	Sistem dapat berjalan dengan baik pada browser Google Chrome dengan versi 61.1.0

##### 3.2.5 Use Case

Use case merupakan tindakan – tindakan yang dilakukan oleh aktor. Use case dirancang berdasarkan hasil kebutuhan fungsional.

Tabel 3. Penjabaran use case berdasarkan kebutuhan fungsional

ID usecase	Nama Use Case	ID kebutuhan fungsional
U02.1	Menambah pasien baru	F02.1
U03.6	Melihat riwayat hasil tindakan medis yang relevan	F03.6

U03.7	Menyimpan pemeriksaan dengan format RDF	hasil dengan	F03.7
-------	---	--------------	-------

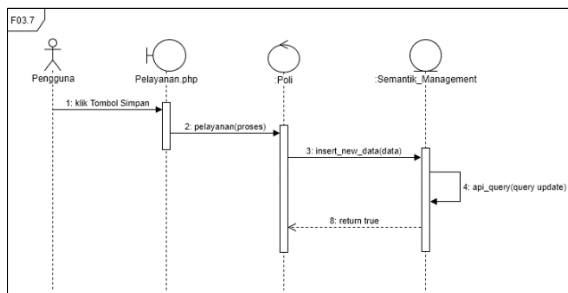
Tabel 3 merupakan sebagian *use case* yang terdapat di aplikasi.

### 3.3 Perancangan Sistem

Pada perancangan sistem dilakukan dalam beberapa tahap, yaitu identifikasi aktor – aktor yang terlibat dalam sistem, pemodelan hasil analisis kebutuhan ke dalam bentuk diagram, perancangan basis data, dan perancangan *ontology*.

#### 3.3.1 Sequence Diagram

*Sequence diagram* menggambarkan langkah – langkah yang dilakukan oleh aktor atau sistem yang merupakan respon dari *use case* tertentu.



Gambar 2. *Sequence Diagram* menyimpan hasil pemeriksaan dengan format RDF

Gambar 2 merupakan *sequence diagram* yang menggambarkan bagaimana hasil pemeriksaan dapat disimpan ke dalam dataset dengan format RDF.

#### 3.3.2 Class Diagram

Kelas – kelas yang digunakan pada aplikasi menggunakan kelas turunan *Controller* dan entitas. Pada penelitian ini, kelas dari kelas turunan *Controller* hanya menghubungkan antara tampilan dan entitas dengan total *controller* sebanyak 6 kelas dan total *entity* sebanyak 15 kelas.

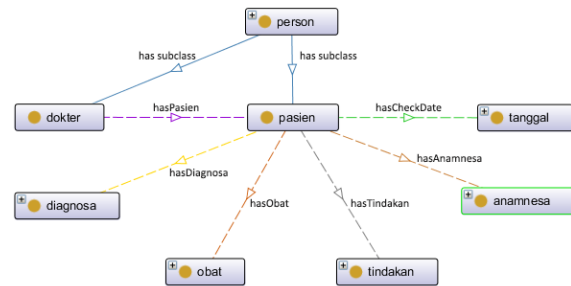
#### 3.3.3 Entity Relationship Diagram

*Entity Relationship Diagram* merupakan perancangan struktur database aplikasi puskesmas. Beberapa istilah yang digunakan di tabel database adalah *primary key* dan *foreign key*.

#### 3.3.4 Perancangan Ontology

Perancangan *ontology* dibutuhkan untuk merepresentasikan pengetahuan apa saja yang diperlukan dalam membangun aplikasi puskesmas. Dalam penelitian ini, *ontology* diberi nama “puskesmas”. *Directed graph* dari

*ontology* puskesmas akan digambarkan pada Gambar 3.



Gambar 3. *Ontology* Puskesmas

Setiap kelas dan *properties* memiliki relasi domain dan range yang dijelaskan pada Tabel 4.

Tabel 4. *Domain Range Ontology* Puskesmas

Domain	Property	Range
Pasien	hasAnamnesa	Anamnesa
Pasien	hasCheckDate	Tanggal
Pasien	hasDiagnosa	Diagnosa
Pasien	hasObat	Obat
Pasien	hasTindakan	Tindakan
Dokter	hasPasien	Pasien
Diagnosa	diagnosa_kode	xsd:string
Diagnosa	keterangan	xsd:string
Tindakan	keterangan	xsd:string
Anamnesa	keterangan	xsd:string
Obat	keterangan	xsd:string
Obat	obat_dosis	xsd:string
Obat	obat_jumlah	xsd:int
Person	person_nama_lengkap	xsd:string

Setiap *class* memiliki *triple* dengan masing – masing *property* baik *object property* maupun *data property*.

#### 3.3.5 Perancangan Algoritme

Perancangan algoritme dilakukan berdasarkan perancangan *class diagram*. Setiap *class* memiliki fungsi algoritme. Perancangan algoritme bertujuan untuk menjabarkan algoritme dari fungsi *class*.

## 4. IMPLEMENTASI

### 4.1 Implementasi Database

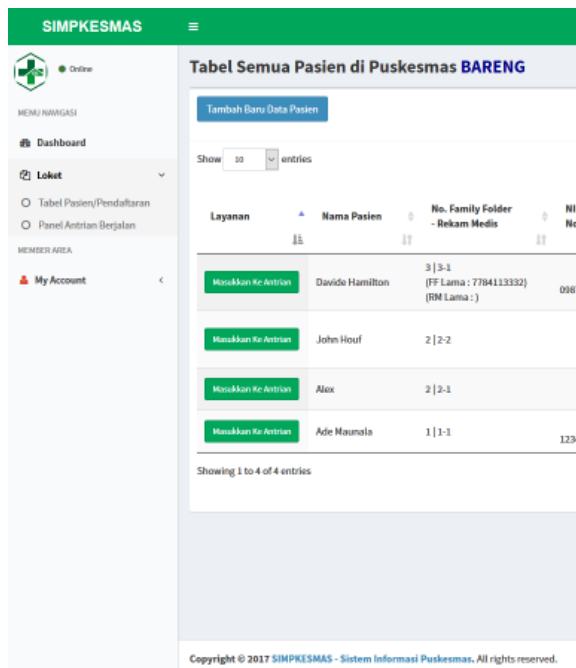
Hasil perancangan *database* pada bab perancangan bagian perancangan *database* diimplementasikan kedalam *database* yang digunakan oleh aplikasi puskesmas.

<input type="checkbox"/> aauth_groups	<input type="checkbox"/> jenis_rawat
<input type="checkbox"/> aauth_group_to_group	<input type="checkbox"/> kabupaten
<input type="checkbox"/> aauth_login_attempts	<input type="checkbox"/> kasir
<input type="checkbox"/> aauth_perms	<input type="checkbox"/> kecamatan
<input type="checkbox"/> aauth_perm_to_group	<input type="checkbox"/> kelurahan
<input type="checkbox"/> aauth_perm_to_user	<input type="checkbox"/> laboratorium_pasien
<input type="checkbox"/> aauth_pms	<input type="checkbox"/> obat
<input type="checkbox"/> aauth_users	<input type="checkbox"/> obat_bentuk
<input type="checkbox"/> aauth_user_to_group	<input type="checkbox"/> obat_ket_cara_pemakaian
<input type="checkbox"/> aauth_user_variables	<input type="checkbox"/> obat_ket_waktu_pemakaian
<input type="checkbox"/> agama	<input type="checkbox"/> obat_pcare
<input type="checkbox"/> antrian	<input type="checkbox"/> obat_prolanis_prb
<input type="checkbox"/> apotek_gudang_obat	<input type="checkbox"/> pasien
<input type="checkbox"/> bagian_puskesmas	<input type="checkbox"/> pegawai
<input type="checkbox"/> biaya	<input type="checkbox"/> pekerjaan
<input type="checkbox"/> configuration	<input type="checkbox"/> person
<input type="checkbox"/> diagnosa	<input type="checkbox"/> poli_diagnosa
<input type="checkbox"/> diagnosa_pcare	<input type="checkbox"/> poli_obat
<input type="checkbox"/> dokter	<input type="checkbox"/> poli_pelayanan
<input type="checkbox"/> jenis_diagnosa	<input type="checkbox"/> poli_tindakan
<input type="checkbox"/> jenis_kasus_diagnosa	<input type="checkbox"/> provinsi
<input type="checkbox"/> jenis_kesadaran	<input type="checkbox"/> status_kawin
<input type="checkbox"/> jenis_kunjungan	<input type="checkbox"/> status_keluar_pasien
<input type="checkbox"/> jenis_pasien	<input type="checkbox"/> status_wilayah
<input type="checkbox"/> jenis_racikan_obat	<input type="checkbox"/> tindakan
	50 tables

Gambar 4. Implementasi Database Aplikasi

### 4.2 Implementasi Antarmuka

Berikut ini adalah beberapa bagian implementasi antarmuka dari aplikasi puskesmas.



Gambar 5. Antarmuka Layanan Loket

## 5. PENGUJIAN

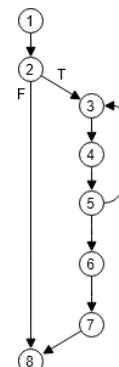
Terdapat 4 (empat) buah pengujian yang akan dilakukan, yang dilakukan antara lain *compatibility testing*, *blackbox testing*, *whitebox testing*, *recall and precision testing*. Pada *whitebox testing* akan menggunakan *basis path testing*.

### 5.1 Compatibility Testing (Non-Fungsional)

Gambar 6. Hasil Pengujian Compatibility

Berdasarkan Gambar 6, hasil pengujian menunjukkan bahwa sistem dapat berjalan di peramban *Google Chrome* dengan versi 61 sebagai versi tertinggi dan peramban *Mozilla Firefox* versi 56 sebagai versi tertinggi.

### 5.2 Basis Path Testing



Gambar 7. Flow Graph fungsi getRelevan

#### Jalur Independen getRelevan

- Jalur 1: 1-2-8
- Jalur 2: 1-2-3-4-5-6-7-8
- Jalur 3: 1-2-3-4-5-3-4-5-6-7-8

#### Cyclomatic Complexity getRelevan

$$V(G) = 3 \text{ Regions}$$

$$V(G) = 9 \text{ Edge} - 8 \text{ Node} + 2 = 3$$

$$V(G) = 2 \text{ Predicate Node} + 1 = 3$$

Dengan nilai  $V(G) = 3$  memperlihatkan bahwa aplikasi masih tergolong tidak terlalu rumit dalam pemeliharaan kode program.

Tabel 5. Kasus Uji Jalur Independen Fungsi getRelevan

No	Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang didapat	Status
1	1-2-8	hasil['isEmpty'] = true	return feedback['isEmpty'] = true	return feedback['isEmpty'] = true	valid
2	1-2-3-4-5-6-7-8	hasil['isEmpty'] = false	return feedback['isEmpty'] = false	return feedback['isEmpty'] = false	valid
3	1-2-3-4-5-3-4-5-6-7-8	hasil['isEmpty'] = true	return feedback['isEmpty'] = true	return feedback['isEmpty'] = true	valid

Fungsi getRelevan merupakan salah satu dari total 3 pengujian *basis path*. Dari 3 kasus uji fungsi getRelevan didapatkan hasil sesuai dengan yang diharapkan.

### 5.3 Black-box Testing

Tabel 6. Pengujian Blackbox Testing

Kode Fungsional	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
F02.1	(1) Petugas Loket masuk ke menu tabel pasien kemudian memilih tombol Tambah Baru Data Pasien  (2) Petugas Loket mengisi semua data di formulir pendaftaran pasien yang telah disediakan kemudian memilih tombol Simpan	Sistem menyimpan semua data yang dimasukkan oleh petugas	Sistem berhasil menyimpan semua data yang dimasukkan oleh petugas	Valid
F03.6	(1) Dokter memilih menu Poli kemudian memilih tombol Layani di baris pasien pada tabel antrian  (2) Dokter mengisi data kata kunci anamnesa di bagian label Cari Riwayat Medis Yang Relevan dan memilih tombol Cari	Sistem menampilkan hasil pencarian yang berhubungan dengan kata kunci dalam bentuk tabel	Sistem berhasil menampilkan hasil pencarian yang berhubungan dengan kata kunci dalam bentuk tabel	Valid

Tabel 6 adalah beberapa pengujian *black-box* dari total 37 pengujian yang dilakukan. Pada total 37 kasus uji didapatkan 37 hasil yang didapat sesuai dengan yang diharapkan maka kebutuhan fungsional sistem valid 100%.

### 5.4 Pengujian Recall dan Precision

Kasus uji yang digunakan diambil berdasarkan gejala umum yang sering terjadi pada pasien. Pengujian menggunakan 5 kasus uji dengan kata kunci gejala atau keluhan.

Tabel 7. Pengujian Nilai Recall

No	Keyword	Relevant items retrieved	Total relevant item in ontology	R
1	Batuk	2	2	1
2	Lelah	3	3	1
3	Nafas memendek	2	2	1
4	Demam	4	4	1
5	Lemas	2	2	1
Nilai rerata R				1

Hasil pengujian nilai *recall* (R) menunjukkan bahwa nilai R yang diperoleh adalah 1. Hal ini menunjukkan bahwa sistem sudah mampu menemukan hampir semua jumlah informasi relevan sesuai dengan jumlah informasi relevan yang disimpan di dalam *ontology*.

Tabel 8. Pengujian Nilai Precision

No	Keyword	Relevant items retrieved	Total item retrieved	P
1	Batuk	2	2	1
2	Lelah	3	3	1
3	Nafas memendek	2	2	1
4	Demam	4	4	1
5	Lemas	1	2	0.5
Nilai rerata P				0.9

Hasil pengujian nilai *precision* (P) menunjukkan bahwa nilai P yang diperoleh adalah 0.9. Nilai *precision* yang diperoleh menyatakan bahwa hampir semua informasi telah relevan atau sesuai dari total keseluruhan informasi yang dikembalikan oleh sistem.

## 6. PENUTUP

Dari hasil pengamatan pada proses analisis kebutuhan, perancangan, implementasi hingga tahap pengujian maka penulis dapat mengambil kesimpulan:

1. Sistem dapat dispesifikasi kebutuhannya dengan melakukan wawancara kepada kepala UPT Puskesmas Bareng dan observasi dan analisis terkait proses bisnis pada proses pengambilan data, pemrosesan data serta penyimpanan data di unit loket, poli,

- laboratorium, dan apotek dalam Puskesmas Bareng.
2. Dari hasil spesifikasi kebutuhan, didapat kebutuhan sistem yang lalu dirancang kedalam use case, use case scenario, sequence diagram, class diagram, entity relationship diagram, perancangan *ontology* dan perancangan antarmuka.
  3. Hasil pengujian terhadap sistem menggunakan pengujian *compatibility*, pengujian *white-box*, pengujian *black-box*, serta pengujian *recall* dan *precision*. Pada *compatibility testing* sistem dapat berjalan dengan baik di peramban *Mozilla Firefox* dan *Google Chrome*. Pada metode *white-box* menggunakan *basis path testing* didapatkan hasil 100% valid dari 9 kasus uji. Dari pengujian *black-box* didapatkan hasil 100% valid dari 37 kasus uji. Pada pengujian *recall* dan *precision* didapatkan nilai *recall* 100% dan nilai *precision* 90% yang membuktikan bahwa pencarian informasi yang terkaitt atau semantik memiliki efektifitas yang tinggi.
- #### DAFTAR PUSTAKA
- Albukhitan, Saeed, 2016. Semantic Annotation of Arabic Web Resources using Semantic Web Services. Saudi Arabia: Elsevier.
- Antoniou, Grigoris, 2004. A Semantic Web Primer. England: MIT Press.
- Bassil, Youssef, 2011. A Simulation Model for the Waterfall Software Development Life Cycle. International Journal of Engineering & Technology. Vol. 2, No. 5, ISSN: 2049-3444.
- Booch, Grady & et all., 2005. The Unified Modeling Language User Guide Second Edition. United States: Addison-Wesley.
- BPJS, 2015. Pahami Lebih Dalam tentang Sistem Rujukan Berjenjang dan Pola Pembayaran BPJS Kesehatan ke Faskes. [online] Tersedia di: <<http://bpjs-kesehatan.go.id/bpjs/index.php/unduh/index/269>> [Diakses 07 Maret 2017].
- BPJS, 2015. Fasilitas Kesehatan. [online] Tersedia di: <<http://www.bpjs-kesehatan.go.id/bpjs/index.php/pages/detail/2014/14/Fasilitas-Kesehatan>> [Diakses 07 Maret 2017].
- Chopra, Rajiv., 2012. Reusing Black Box Test Paths For White Box Testing of Websites. IEEE International Advance Computing Conference (IACC). ISBN: 978-1-4673-4529-3.
- Deitel, P., 2004. Internet & World Wide Web How To Program. 3rd ed. New Jersey.
- Ducharme, Bob., 2011. Learning SPARQL. United States: O'Reilly.
- Ferru, Daniele, 2016. Write Once Run Anywhere Custom SPARQL Functions. Jurnal IEEE.
- Fobel, Andrew dan Subramanian, Nary, 2016. Comparison of the Performance of Drools and Jena Rule-Based Systems for Event Processing on the Semantic Web. USA: IEEE.
- Hebeler, John & et all., 2009. Semantic Web Programming. Indianapolis: Wiley.
- Kemenkes, 2016. Data Dasar Puskesmas Provinsi Jawa Timur. Jakarta: Kemenkes RI.
- Kemenkes, 2015. Profil Kesehatan Indonesia. Jakarta: Kemenkes RI.
- Kim, Il-Woong, 2009. A Model-Driven Approach for Describing Semantic Web Services: From UML to OWL-S. Jurnal IEEE vol. 39, no. 6.
- Kumar, Naresh., 2013. Evolving a New Software Development Life Cycle Model SDLC-2013 with Client Satisfaction. International Journal of Soft Computing and Engineering (IJSCE). Volume-3, Issue-1, March 2013, ISSN: 2231-2307.
- Lestari, Nisaa Putri, 2016. Uji Recall and Precision Sistem Temu Kembali Informasi OPAC Perpustakaan ITS Surabaya. Universitas Airlangga.
- Ljubuncic, Igor., 2011. Apache Web Server Complete Guide.
- Medidata, 2015. MIMS Petunjuk Konsultasi Indonesia. Edisi 15. Jakarta: Bhuna Ilmu Komputer.
- Mohebbi, Keyvan., 2012. Contemporary Semantic Web Service Frameworks: An Overview And Comparisons. International Journal on Web Service Computing (IJWSC). DOI: 10.5121/ijwsc.2012.3306.
- Nurwahdah, Aeny, 2014. Rancang Bangun Aplikasi Tanya Jawab Menggunakan Semantic Web Dan Algoritma Nazief & Adriani Sebagai Metode Stemmer



- Kalimat Tanya. Universitas Islam Negeri Maulana Malik Ibrahim Malang.
- Pressman, Roger S., 2001 . Software Engineering: A Practitioner's Approach, Fifth Edition.
- Purewal, Semmy. 2014. Learning Web App Development. Sebastopol: O'Reilly.
- Ramadhan, Mukhis. 2009. Desain Web Dengan PHP. Jurnal SAINTIKOM.
- Rovan, Lidia dan Nižetic, Ivana, 2008. Extending Java Web Applications for Semantic Web. Croatia: ITI.
- Rumbaugh, James & et all., 2004. The Unified Modeling Language Reference Manual. Boston: Addison-Wesley.
- Suryanarayana, D., 2011. 'Stepping Towards a Semantic Web Search Engine for Accurate Outcomes in Favor of User Queries Using RDF and Ontology Technologies'. Jurnal IEEE.
- Verma, Neha & et all., 2015. E-commerce website ranking using semantic web mining and neural computing. India: Elsevier.
- Wahyudi, A. T., 2013. Semantic Search Pada Digital Library Online Public Access Catalog. STMIK EL RAHMA.
- W3C, 2009. OWL 2 Web Ontology Language Primer. [online] Tersedia di: <<https://www.w3.org/TR/2009/REC-owl2-primer-20091027/>> [Diakses 01 Desember 2017].
- W3C, 2004. RDF Primer. [online] Tersedia di: <<https://www.w3.org/TR/2004/REC-rdf-primer-20040210/>> [Diakses 01 Desember 2017].