BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap penerapan rancangan hasil analisis dan pengembangan konsep sistem ke dalam suatu bahasa pemrograman tertentu, serta penerapan kebutuhan perangkat lunak, perangkat keras dan server.

4.1.1 Lingkungan Implementasi

Lingkungan implementasi menjelaskan kebutuhan pendukung yang dapat menunjang berjalannya sistem, seperti *hardware*, *software*, dan spesifikasi *server*.

4.1.1.1 Lingkungan Implementasi *Hardware*

Spesifikasi kebutuhan *hardware* pada pengembangan *web service* untuk aplikasi pelayanan rumah sakit pada *platform Android* adalah:

Tabel 4.1 Spesifikasi Kebutuhan *Hardware*

Sub Sistem Web	Sub Sistem Mobile
Monitor resolusi 1366 x 768 pixel	Layar WVGA resolusi 800 x 400 pixel
RAM DDR3 2GB	RAM 1GB
Processor 2.4Ghz	*Processor* 1.2Ghz
VGA 512 GB	GPU Adreno 302
Harddisk 320 GB	Memory 4GB
Koneksi Internet	Koneksi Internet
-	GPS Support

4.1.1.2 Lingkungan Implementasi *Software*

Spesifikasi kebutuhan *software* pada pengembangan *web service* untuk aplikasi pelayanan rumah sakit pada *platform Android* adalah:

Tabel 4.2 Spesifikasi Kebutuhan *Software*

Sub Sistem Web		Sub Sistem Mobile
Windows 8		Sistem Operasi *Android* 4.0 Ice Cream Sandwich
Web Browser	Mozilla Firefox 48.0	
	Google Chrome 52.0	

4.1.1.3 Lingkungan Implementasi *Server*

Spesifikasi kebutuhan *server* pada pengembangan *web service* untuk aplikasi pelayanan rumah sakit pada *platform Android* adalah:

Tabel 4.3 Spesifikasi Kebutuhan *Server*

Sub Sistem Web
Web Space 250 MB
Bandwidth 50 GB Bulan
MySQL Database (5.1 & 5.5)
PHP (5.5)
CPanel 11.x
Apache 2.x mod_deflate

4.2 Implementasi Data

Implementasi data merupakan penerapan basis data dalam pengembangan *web service* untuk aplikasi pelayanan rumah sakit pada *platform Android*. Perancangan basis data yang telah dibuat sebelumnya menggunakan MySQL. Berikut ini merupakan implementasi basis data menggunakan MySQL:

1. Pembuatan *Database*

```
CREATE DATABASE gohospit_go_hospital_app;
```

2. Pembuatan Tabel Rumah Sakit

```
CREATE TABLE tb_rs (
  id_rsINT NOT NULL,
  Nama_RS VARCHAR(30) NOT NULL,
  Alamat VARCHAR(50) NOT NULL,
  No_Tlp VARCHAR(12) NOT NULL,
  Loc_LatitudeFLOAT,
  Loc_LongitudeFLOAT,
PRIMARY KEY (ID_RS)
)
ENGINE=InnoDB;
```

3. Pembuatan Tabel Fasilitas

```
CREATE TABLE tb_fasilitas
(
  id_fasilitas INT NOT NULL,
  id_rs INT NOT NULL,
  PRIMARY KEY(id_fasilitas)
  FOREIGN KEY (id_rs) REFERENCES tb_rs(id_rs)
)
ENGINE=InnoDB;
```

4. Pembuatan Tabel Rawat Inap

```
CREATE TABLE tb_subfasilitas_rawatinap
(
  id_subfasilitas INT NOT NULL,
  tipe_ruangan VARCHAR(10) NOT NULL,
  nama_ruangan VARCHAR(25) NOT NULL,
  status ENUM('0','1') NOT NULL DEFAULT '0'
  COMMENT '0 = TERSEDIA , 1 = TERISI',
  FOREIGN KEY(id_subfasilitas)
  REFERENCES tb_fasilitas(id_subfasilitas)
)
ENGINE=InnoDB;
```

5. Pembuatan Tabel Subfasilitas

```
CREATE TABLE tb_subfasilitas
(
  id_subfasilitas CHAR(4) NOT NULL,
  nama_fasilitas VARCHAR(25) NOT NULL,
  FOREIGN KEY(id_subfasilitas)
  REFERENCES tb_fasilitas(id_subfasilitas)
```

```
)
ENGINE=InnoDB;
```

6. Pembuatan Tabel Dokter

```
CREATE TABLE tb_dokter
(
  id_dokter INT NOT NULL PRIMARY KEY,
  id_rs INT NOT NULL,
  nama VARCHAR(30) NOT NULL
  alamat VARCHAR(30) NOT NULL
  jenis_kelamin VARCHAR(6) NOT NULL
  no_tlp VARCHAR(12) NOT NULL
  FOREIGN KEY(id_rs)
  REFERENCES tb_rs(id_rs)
)
ENGINE=InnoDB;
```

7. Pembuatan Tabel Jadwal Praktek Dokter

```
CREATE TABLE tb_jadwalpraktek
(
      id_jadwal INT NOT NULL PRIMARY KEY,
      id_dokter INT NOT NULL,
      Tanggal TIMESTAMP NOT NULL,
      Waktu TIMESTAMP NOT NULL,
      FOREIGN KEY (id_dokter)
      REFERENCES tb_dokter(id_dokter)
)
ENGINE=InnoDB;
```

8. Pembuatan Tabel Pengguna

```
CREATE TABLE tb_pengguna
(
      id_pengguna INT NOT NULL PRIMARY KEY,
      nama VARCHAR(30) NOT NULL,
      alamat VARCHAR(50) NOT NULL,
      jenis_kelamin VARCHAR(6) NOT NULL,
      agama VARCHAR(7)NOT NULL,
      pendidikan VARCHAR(10)NOT NULL,
      pekerjaan VARCHAR(20)NOT NULL,
      status_pernikahan VARCHAR(12)NOT NULL,
      no_tlp VARCHAR(12),
      username VARCHAR(25)NOT NULL,
      password VARCHAR(16)NOT NULL
)
ENGINE=InnoDB;
```

9. Pembuatan Tabel *Appointment*

```
CREATE TABLE tb_appointment
(
      no_appointment INT NOT NULL PRIMARY KEY,
      id_rs INT NOT NULL,
      id_pengguna INT NOT NULL,
      id_dokter INT NOT NULL,
      tanggal_app TIMESTAMP NOT NULL,
      waktu_app TIMESTAMP NOT NULL,
      nama_pasien VARCHAR(30) NOT NULL,
      alamat_pasien VARCHAR(50) NOT NULL,
      nama_dokter VARCHAR(30) NOT NULL,
      spesialis VARCHAR(25) NOT NULL,
      FOREIGN KEY (id_rs)
      REFERENCES tb_rs(id_rs),
      FOREIGN KEY (id_pengguna)
      REFERENCES tb_pengguna(id_pengguna),
      FOREIGN KEY (id_dokter)
      REFERENCES tb_dokter(id_dokter)
)
ENGINE=InnoDB;
```

10. Pembuatan Tabel Spesialis Dokter

```
CREATE TABLE tb_spesialis
(
      id_spesialis CHAR(4) NOT NULL PRIMARY KEY,
      id_dokter INT NOT NULL,
      nama_spesialis VARCHAR(25) NOT NULL,
      FOREIGN KEY (id_dokter)
      REFERENCES tb_dokter(id_dokter)
)
ENGINE=InnoDB;
```

11. Pembuatan Tabel Admin

```
CREATE TABLE tb_admin
(
      id_admin INT NOT NULL PRIMARY KEY,
      nama VARCHAR(30) NOT NULL,
      username VARCHAR(16),
      password VARCHAR(16)
```

```
)
ENGINE=InnoDB;
```

4.3 Implementasi Antarmuka

Implementasi antarmuka berisi pemaparan setiap tampilan perangkat lunak yang dibangun beserta file xml sebagai layout dari masing-masing antarmuka. Adapun implementasi antarmuka perangkat lunak yang dibangun terdiri dari nama antarmuka beserta *class* atau *file* yang mewakilinya.

a. Implementasi antarmuka subsistem *web*

Implementasi antarmuka subsistem web dapat dilihat pada Tabel 4.4.

Tabel 4.4 Implementasi Antarmuka Subsistem Web

No	Nama Antarmuka	Nama File
1	FW01- *Login*	index.php
2	FW02- Beranda	dashboard.php
3	FW03- Pengolahan Data Rumah Sakit	rsprocess.php
4	FW04- Tambah Data Rumah Sakit	rsprocess.php
5	FW05- Ubah Data Rumah Sakit	rsprocess.php
6	FW06- Lihat Fasilitas Rumah Sakit	facilityprocess.php
7	FW07- Lihat Ruangan Rumah Sakit	roomprocess.php
8	FW08- Lihat Dokter	doctorprocess.php
9	FW09- Lihat Spesialis	specialistprocess.php

b. Implementasi Antarmuka Subsistem *Mobile*

Implementasi antarmuka subsistem *mobile* dapat dilihat pada Tabel 4.5.

Tabel 4.5 Implementasi Antarmuka Subsistem *Mobile*

No	Nama Antarmuka	Nama File
1	FM01- Menu Utama	home.xml
2	FM02–Pencarian Rumah Sakit	Searchrs.xml
3	FM05–Info Rumah Sakit	infors.xml
4	FM06 –Berita dan Kegiatan	berita.xml
5	FM07 – Pencarian Dokter	Searchdoctor.xml

No	Nama Antarmuka	Nama File
6	FM10 – Info Doctor	Infodoctor.xml
7	FM11 – Buat *Appointment*	Appointment.xml
8	FM14 - *Login*	Login.xml
9	FM15 – Profil Pribadi	profile.xml
10	FM16 – Lihat *Appointment*	LihatAppointment.xml
11	FM17 – Lihat Berita	Newsfeed.xml

4.4 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan tahapan untuk menganalisis kesesuaian perangkat lunak yang telah dirancang sehingga dapat diketahui apakah perangkat lunak tersebut memenuhi kriteria sesuai tujuan atau tidak.

4.4.1 Rencana Pengujian

Pengujian sistem yang akan dilakukan hanya berfokus pada fungsionalitas perangkat lunak yang dibangun. Pengujian ini dilakukan dengan metode pengujian *black box*.

4.4.2 Pengujian Black Box

Skenario pengujian ini akan memaparkan urutan dalam pengujian yang akan dilakukan pada perangkat lunak yang dibangun. Urutan setiap poin pengujian yang dilakukan berdasarkan dari maksud dan tujuan diadakannya penelitian ini, sehingga dapat menilai dan mengevaluasi maksud dan tujuan yang ingin dicapai pada penelitian ini. Pada pengujian *black box* ini dibutuhkan beberapa sistem informasi rumah sakit untuk melakukan validasi integrasi data pada setiap rumah sakit. Maka dari itu, penelitian ini menggunakan suatu *sample* aplikasi dari dua sistem informasi rumah sakit, yaitu aplikasi *sample-hospital-a* dan aplikasi *sample-hospital-b*.

Skenario pengujian perangkat lunak yang akan dibangun adalah skenario pengujian fungsionalitas, dimana tahap ini dibagi menjadi 2 pengujian yaitu pengujian subsistem *web* dan subsistem *mobile*.

1. Skenario pengujian subsistem *web*

   Pada pengujian ini semua fitur pada subsistem *web* yang dibangun akan diuji menggunakan metode *black box*. Skenario yang menerangkan fungsionalitas yang akan diuji akan diterangkan pada tabel 4.6 dibawah ini:

Tabel 4.6 Skenario Pengujian Subsistem *Web*

KELAS UJI	POIN PENGUJIAN	JENIS PENGUJIAN	METODE PENGUJIAN
Integrasi Data Spesialis Dokter	Penambahan data spesialis	*Black box*	*Equivalence Partitioning*
	Melakukan validasi integrasi pada data spesialis dokter	*Black box*	*Equivalence Partitioning*
Integrasi Data Dokter	Penambahan data dokter	*Black box*	*Equivalence Partitioning*
	Melakukan validasi integrasi pada data dokter	*Black box*	*Equivalence Partitioning*
Integrasi Data Fasilitas Rumah Sakit	Penambahan data fasilitas rumah sakit	*Black box*	*Equivalence Partitioning*
	Melakukan validasi integrasi pada data fasilitas rumah sakit	*Black box*	*Equivalence Partitioning*
Integrasi Data Ruangan Rumah Sakit	Penambahan data ruangan rumah sakit	*Black box*	*Equivalence Partitioning*
	Melakukan validasi integrasi pada data ruangan rumah sakit	*Black box*	*Equivalence Partitioning*

2. Skenario pengujian subsistem *mobile*

   Pada pengujian ini semua fitur pada subsistem *mobile* yang dibangun akan diuji menggunakan metode *black box*. Skenario yang menerangkan fungsionalitas yang akan diuji akan diterangkan pada Tabel 4.18 dibawah ini:

Tabel 4.7 Skenario Pengujian Subsistem *Mobile*

KELAS UJI	POIN PENGUJIAN	JENIS PENGUJIAN	METODE PENGUJIAN
Lihat Info Rumah Sakit	Menguji validitas informasi rumah sakit dengan membandingkan informasi pada sistem *Android* dengan informasi yang terdapat pada *server* atau rumah sakit terkait	*Black Box*	*Equivalence Partitioning*
Lihat Info Dokter	Menguji validitas informasi dokter dengan membandingkan informasi pada sistem *Android* dengan informasi yang terdapat pada *server* atau rumah sakit terkait	*Black box*	*Equivalence Partitioning*
Login	Menguji fungsionalitas apakah sistem dapat berhasil melakukan *login*	*Black Box*	*Equivalence Partitioning*
Pembuatan *Appoitnment*	Menguji validitas data *appointment* yang dikirimkan dari *mobile platform* ke aplikasi rumah sakit	*Black box*	*Equivalence Partitioning*
Lihat *Appointment*	Menampilkan daftar *appointment* yang telah dibuat sebelumnya	*Black box*	*Equivalence Partitioning*

4.4.3 Hasil Pengujian *Black Box*

Hasil pengujian menampilkan hasil dari pengujian yang dilakukan sesuai dengan rancangan dan skenario pengujian fungsionalitas. Hasil pengujian fungsionalitas terbagi menjadi dua yaitu hasil pengujian fungsionalitas sub sistem web dan hasil pengujian fungsionalitas sub sistem mobile. Berikut adalah data hasil pengujian fungsionalitas sistem.

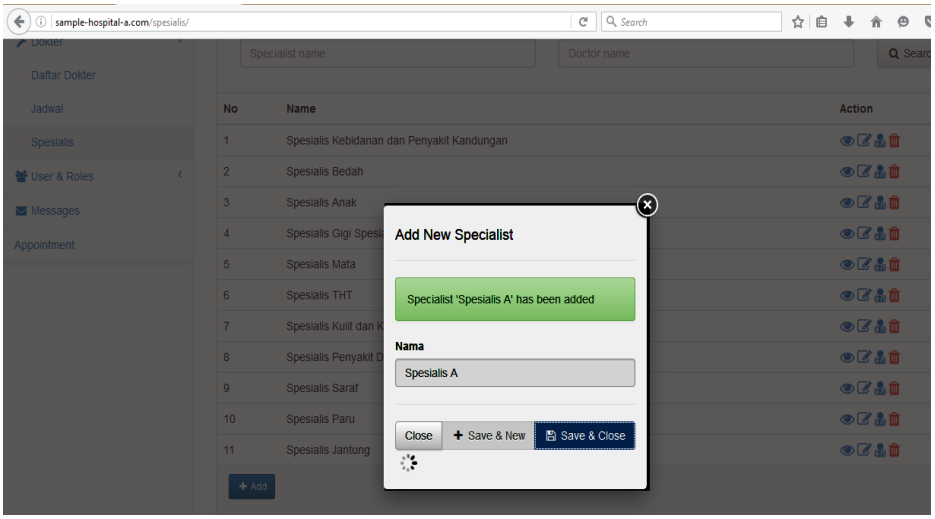
1. Hasil Pengujian Fungsionalitas Subsistem *Web*
    a. Pengujian Integrasi Data Spesialis

        Pengujian integrasi data spesialis memiliki dua poin pengujian, yaitu poin pengujian penambahan data spesialis dan poin pengujian validasi integrasi data spesialis. Berikut di bawah ini merupakan penjelasan dari setiap poin pengujian:

        i. Poin Pengujian Penambahan Data Spesialis

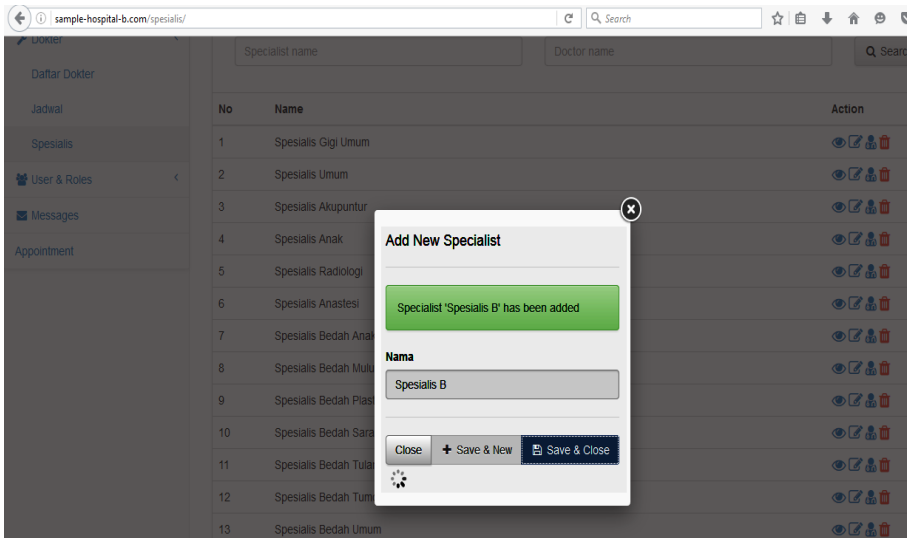
            Pada poin pengujian penambahan data spesialis, yang dilakukan adalah menambah data spesialis pada aplikasi *sample-hospital-a* dan aplikasi

sample-hospital-b. Berikut di bawah ini merupakan penambahan data spesialis dengan nama spesialis "Spesialis A" pada aplikasi *sample-hospital-a*:



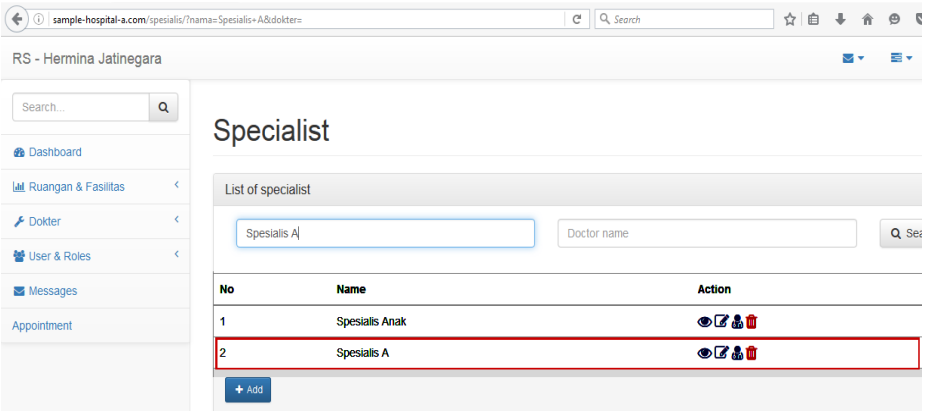
Gambar 4.1 Penambahan Data Spesialis Pada Aplikasi *Sample-Hospital-A*

Setelah melakukan penambahan data spesialis pada aplikasi *sample-hospital-a*, selanjutnya adalah melakukan penambahan data spesialis pada aplikasi *sample-hospital-b*. berikut di bawah ini merupakan gambar penambahan data spesialis dengan nama "Spesialis B" pada aplikasi *sample-hospital-b*:

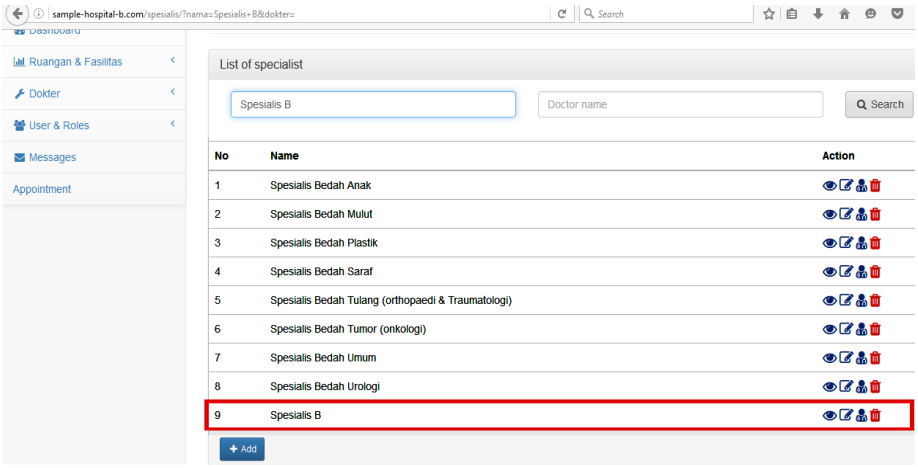


Gambar 4.2 Penambahan Data Spesialis Pada Aplikasi *Sample-Hospital-B*

Setelah menambahkan data spesialis pada aplikasi *sample-hospital-a* dan *sample-hospital-b*, hal yang perlu dilakukan adalah memeriksa apakah data spesialis yang telah ditambahkan ke masing-masing aplikasi telah tersimpan ke dalam *database*. Berikut di bawah ini merupakan gambar pencarian data spesialis yang telah ditambahkan pada *sample-hospital-a* dan *sample-hospital-b* dapat dilihat pada gambar 4.3 dan 4.4.



Gambar 4.3 Pencarian Spesialis Pada *Sample-Hospital-A*



Gambar 4.4 Pencarian Spesialis Pada *Sample-Hospital-B*

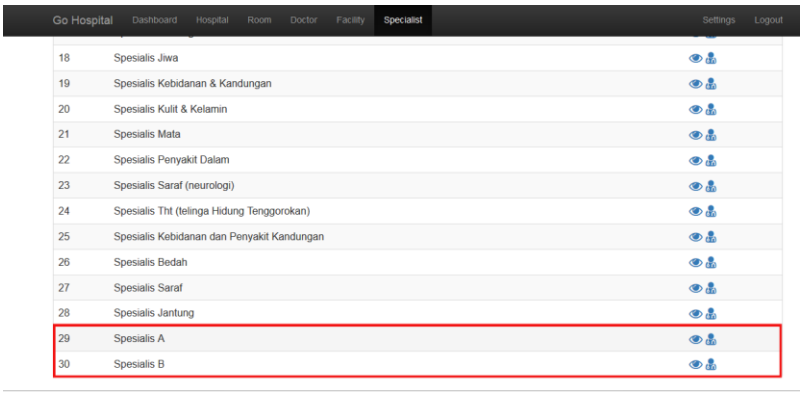
Dengan melihat gambar di atas dapat diketahui bahwa data yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah tersimpan pada masing-masing *database*. Sehingga hasil poin pengujian penambahan data spesialis dapat dilihat seperti tabel 4.8 di bawah ini:

Tabel 4.8 Tabel Hasil Poin Pengujian Penambahan Data Spesialis

POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Penambahan data spesialis	Data spesialis yang telah ditambahkan akan tersimpan ke dalam *database*	Data spesialis yang telah ditambahkan tersimpan ke dalam *database*	Valid

ii. Poin Pengujian Validasi Integrasi Data Spesialis

Untuk melakukan pengujian validasi integrasi data spesialis dapat dilakukan dengan cara memeriksa apakah data spesialis yang telah ditambahkan sebelumnya pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah tersimpan pada *database server*. Maka hal yang perlu dilakukan adalah memeriksa *list view* spesialis yang terdapat pada API. Berikut di bawah ini merupakan *list view* spesialis yang terdapat pada API dapat dilihat pada gambar 4.5.



Gambar 4.5 *List View* Spesialis Pada API

Dengan melihat *list view* spesialis pada API seperti pada gambar 4.5 di atas, maka dapat diketahui bahwa data spesialis dengan nama "Spesialis A" dan "Spesialis B" yang telah ditambahkan pada poin pengujian sebelumnya telah tersimpan pada *database server* dan dapat disimpulkan bahwa *database* pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah terintegrasi dengan *database server*. Sehingga hasil poin pengujian validasi integrasi data spesialis dapat dilihat seperti tabel 4.9 di bawah ini:

Tabel 4.9 Tabel Hasil Poin Pengujian Integrasi Data Spesialis

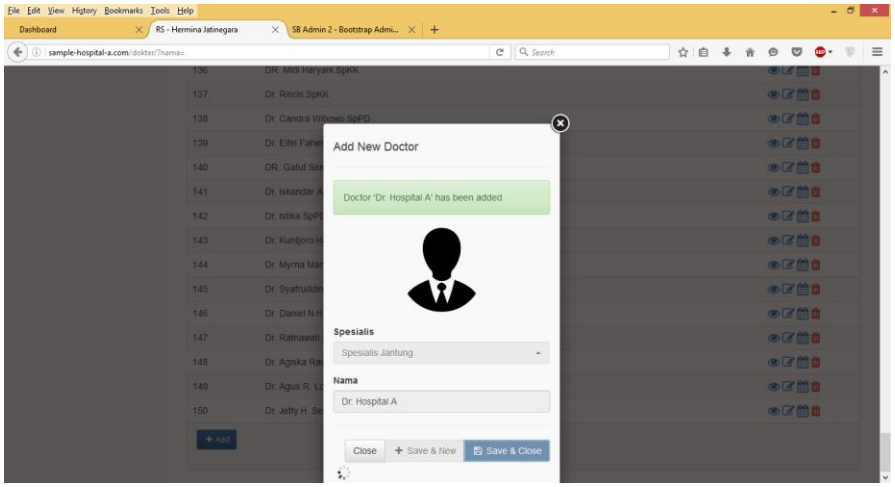
POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Validasi Integrasi Data Spesialis	Data spesialis yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* akan tersimpan pada *database server*	Data spesialis yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* tersimpan pada *database server*	Valid

b. Pengujian Integrasi Data Dokter

Pengujian integrasi data dokter memiliki dua poin pengujian, yaitu poin pengujian penambahan data dokter dan poin pengujian validasi integrasi data dokter. Berikut di bawah ini merupakan penjelasan dari setiap poin pengujian:

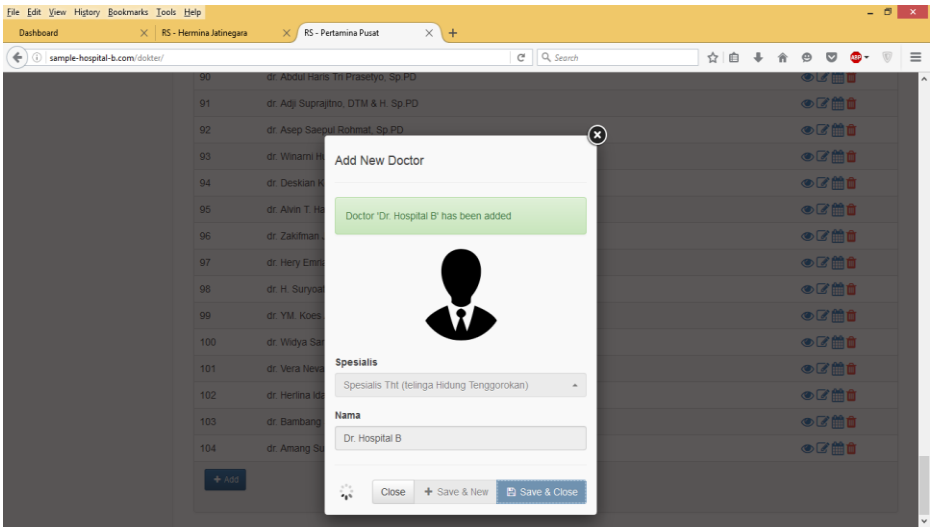
i. Poin Pengujian Penambahan Data Dokter

Pada poin pengujian penambahan data dokter, yang dilakukan adalah menambah data dokter pada aplikasi *sample-hospital-a* dan aplikasi *sample-hospital-b*. Berikut di bawah ini merupakan penambahan data dokter dengan nama dokter "Dr. Hospital A" pada aplikasi *sample-hospital-a*:



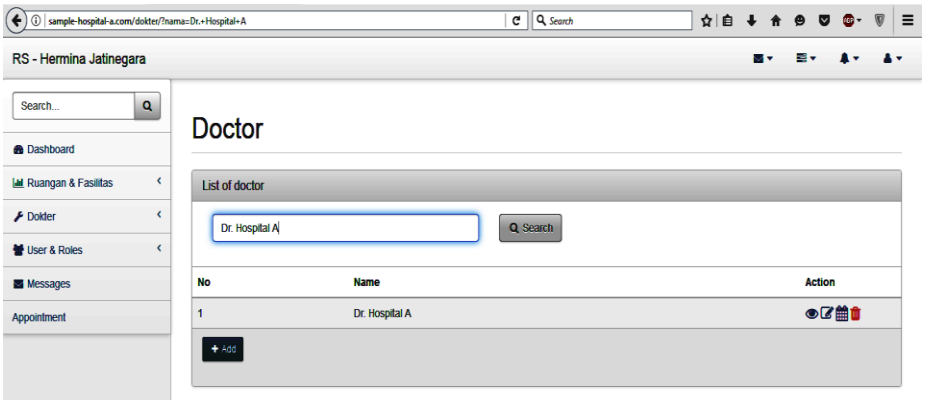
Gambar 4.6 Penambahan Data Dokter Pada Aplikasi *Sample-Hospital-A*

Setelah melakukan penambahan data dokter pada aplikasi *sample-hospital-a*, selanjutnya adalah melakukan penambahan data dokter pada aplikasi *sample-hospital-b*. berikut di bawah ini merupakan gambar penambahan data dokter dengan nama "Dr. Hospital B" pada aplikasi *sample-hospital-b*:

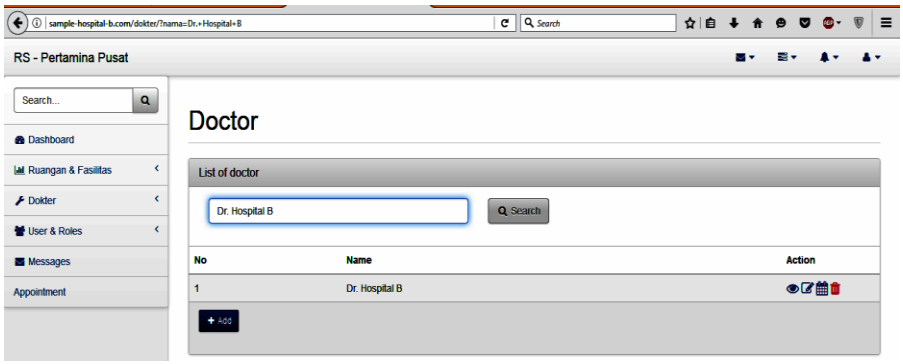


Gambar 4.7 Penambahan Data Dokter Pada Aplikasi *Sample-Hospital-B*

Setelah menambahkan data dokter pada aplikasi *sample-hospital-a* dan *sample-hospital-b*, hal yang perlu dilakukan adalah memeriksa apakah data dokter yang telah ditambahkan ke masing-masing aplikasi telah tersimpan ke dalam *database*. Berikut di bawah ini merupakan gambar pencarian data dokter yang telah ditambahkan pada *sample-hospital-a* dan *sample-hospital-b* dapat dilihat pada gambar 4.8 dan 4.9.



Gambar 4.8 Pencarian Dokter Pada *Sample-Hospital-A*



Gambar 4.9 Pencarian Dokter Pada *Sample-Hospital-B*

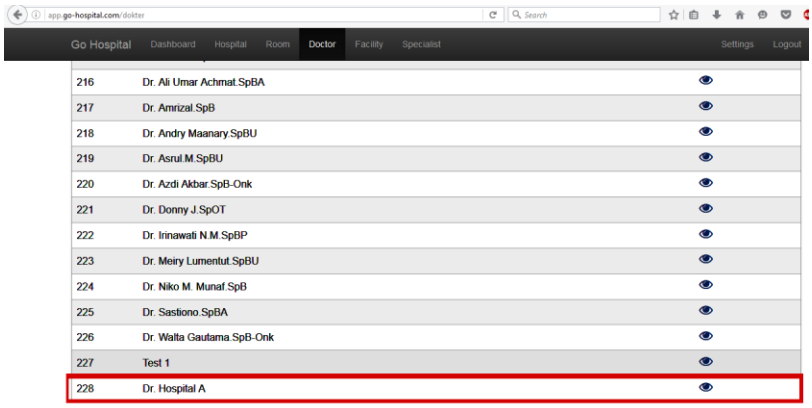
Dengan melihat gambar di atas dapat diketahui bahwa data yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah tersimpan pada masing-masing *database*. Sehingga hasil poin pengujian penambahan data dokter dapat dilihat seperti tabel 4.10 di bawah ini:

Tabel 4.10 Tabel Hasil Poin Pengujian Penambahan Data Dokter

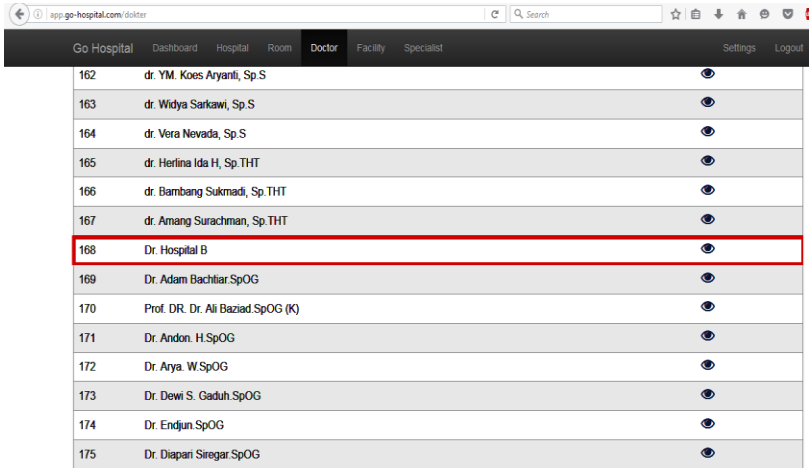
POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Penambahan data dokter	Data dokter yang telah ditambahkan akan tersimpan ke dalam *database*	Data dokter yang telah ditambahkan tersimpan ke dalam *database*	Valid

ii. Poin Pengujian Validasi Integrasi Data Dokter

Untuk melakukan pengujian validasi integrasi data dokter dapat dilakukan dengan cara memeriksa apakah data dokter yang telah ditambahkan sebelumnya pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah tersimpan pada *database server*. Maka hal yang perlu dilakukan adalah memeriksa *list view* dokter yang terdapat pada API. Berikut di bawah ini merupakan *list view* dokter yang terdapat pada API dapat dilihat pada gambar 4.10 dan gambar 4.11.



Gambar 4.10 *List View* Dokter Pada API – "Dr. Hospital A"



Gambar 4.11 *List View* Dokter Pada API – "Dr. Hospital B"

Dengan melihat *list view* dokter pada API seperti pada gambar 4.5 dan 4.6 di atas, maka dapat diketahui bahwa data dokter dengan nama "Dr. Hospital A" dan "Dr. Hospital B" yang telah ditambahkan pada poin pengujian sebelumnya telah tersimpan pada *database server* dan dapat disimpulkan bahwa *database* pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah terintegrasi dengan *database server*. Sehingga hasil poin pengujian validasi integrasi data dokter dapat dilihat seperti tabel 4.11 di bawah ini:

Tabel 4.11 Tabel Hasil Poin Pengujian Integrasi Data Dokter

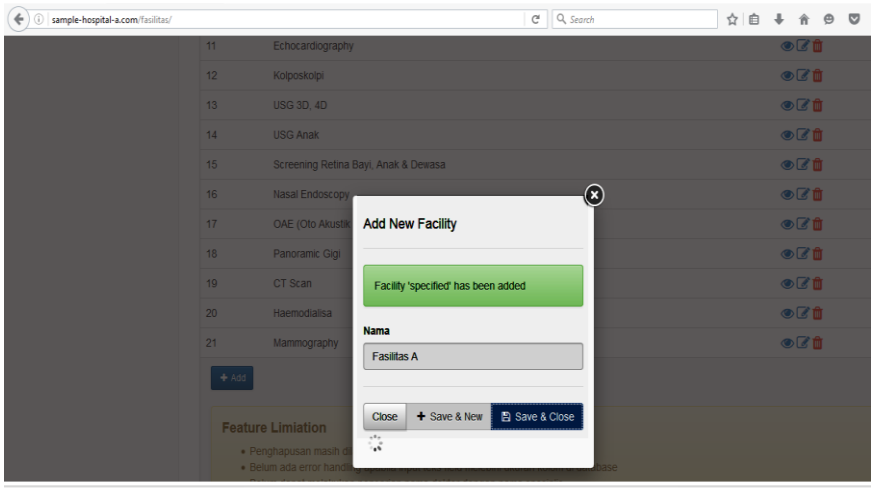
POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Validasi Integrasi Data Dokter	Data yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* akan tersimpan pada *database server*	Data yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* tersimpan pada *database server*	Valid

c. Pengujian Integrasi Data Fasilitas

Pengujian integrasi data fasilitas rumah sakit memiliki dua poin pengujian, yaitu poin pengujian penambahan data fasilitas rumah sakit dan poin pengujian validasi integrasi data fasilitas rumah sakit. Berikut di bawah ini merupakan penjelasan dari setiap poin pengujian:

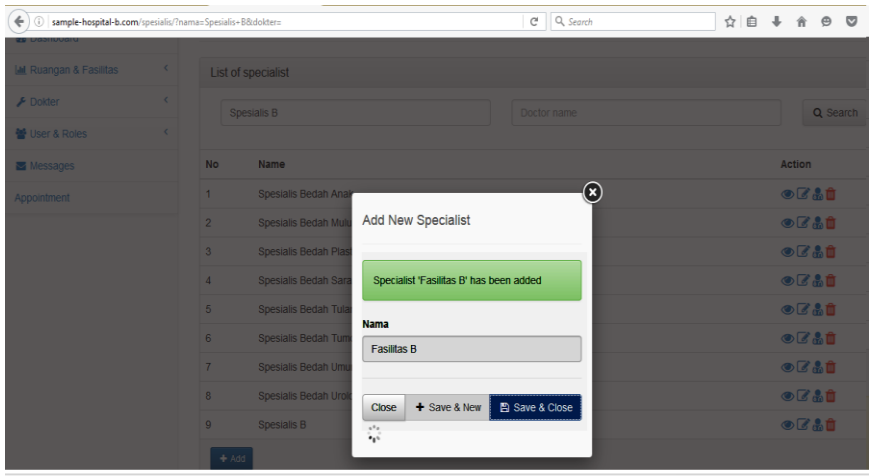
i. Poin Pengujian Penambahan Data Fasilitas

Pada poin pengujian penambahan data fasilitas rumah sakit, yang dilakukan adalah menambah data fasilitas rumah sakit pada aplikasi *sample-hospital-a* dan aplikasi *sample-hospital-b*. Berikut di bawah ini merupakan penambahan data fasilitas dengan nama fasilitas rumah sakit “Fasilitas A” pada aplikasi *sample-hospital-a*:



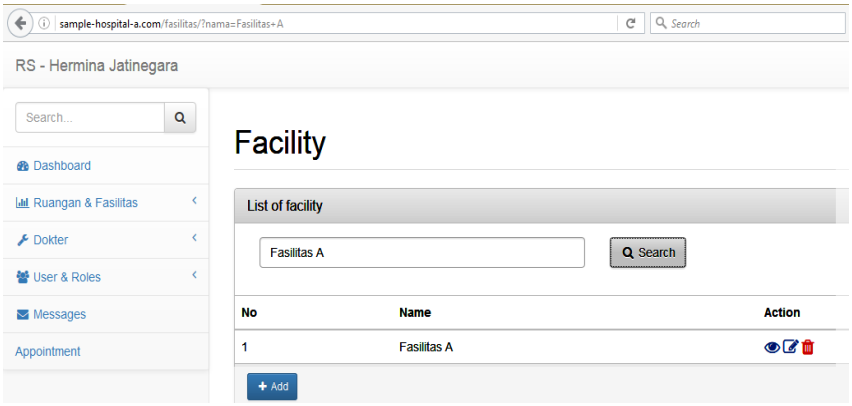
Gambar 4.12 Penambahan Data Fasilitas Pada Aplikasi *Sample-Hospital-A*

Setelah melakukan penambahan data fasilitas pada aplikasi *sample-hospital-a*, selanjutnya adalah melakukan penambahan data fasilitas pada aplikasi *sample-hospital-b*. berikut di bawah ini merupakan gambar penambahan data fasilitas dengan nama "Fasilitas B" pada aplikasi *sample-hospital-b*:

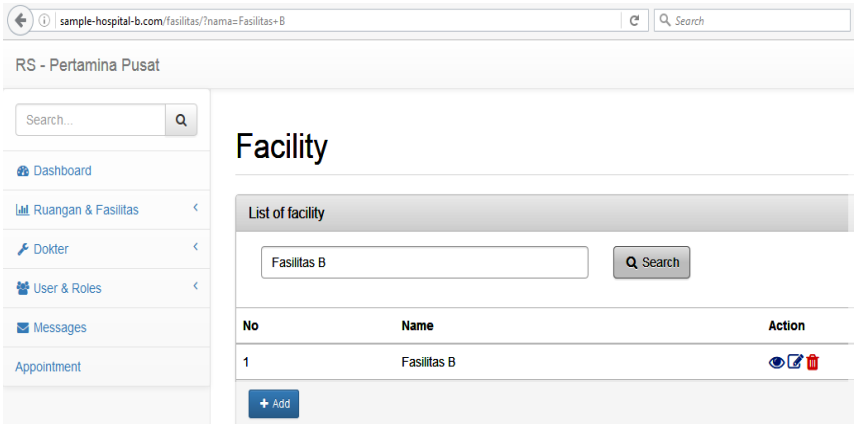


Gambar 4.13 Penambahan Data Fasilitas Pada Aplikasi *Sample-Hospital-B*

Setelah menambahkan data fasilitas rumah sakit pada aplikasi *sample-hospital-a* dan *sample-hospital-b*, hal yang perlu dilakukan adalah memeriksa apakah data fasilitas rumah sakit yang telah ditambahkan ke masing-masing aplikasi telah tersimpan ke dalam *database*. Berikut di bawah ini merupakan gambar pencarian data fasilitas rumah sakit yang telah ditambahkan pada *sample-hospital-a* dan *sample-hospital-b* dapat dilihat pada gambar 4.14 dan 4.15.



Gambar 4.14 Pencarian Fasilitas Pada *Sample-Hospital-A*



Gambar 4.15 Pencarian Fasilitas Pada *Sample-Hospital-B*

Dengan melihat gambar di atas dapat diketahui bahwa data yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah

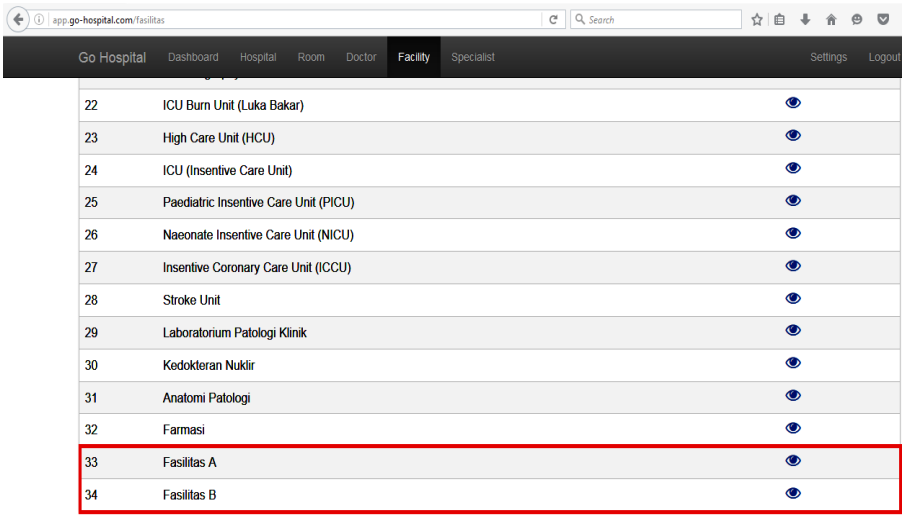
tersimpan pada masing-masing *database*. Sehingga hasil poin pengujian penambahan data fasilitas rumah sakit dapat dilihat seperti tabel 4.12 di bawah ini:

Tabel 4.12 Tabel Hasil Poin Pengujian Penambahan Data Fasilitas

POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Penambahan data fasilitas	Data fasilitas yang telah ditambahkan akan tersimpan ke dalam *database*	Data fasilitas yang telah ditambahkan tersimpan ke dalam *database*	Valid

ii. Poin Pengujian Validasi Integrasi Data Fasilitas

Untuk melakukan pengujian validasi integrasi data fasilitas rumah sakit dapat dilakukan dengan cara memeriksa apakah data fasilitas rumah sakit yang telah ditambahkan sebelumnya pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah tersimpan pada *database server*. Maka hal yang perlu dilakukan adalah memeriksa *list view* fasilitas rumah sakit yang terdapat pada API. Berikut di bawah ini merupakan *list view* fasilitas rumah sakit yang terdapat pada API dapat dilihat pada gambar 4.16.



Gambar 4.16 *List View* Fasilitas Pada API

Dengan melihat *list view* fasilitas rumah sakit pada API seperti pada gambar 4.15 dan 4.16 di atas, maka dapat diketahui bahwa data fasilitas rumah sakit dengan nama "Fasilitas A" dan "Fasilitas B" yang telah ditambahkan pada poin pengujian sebelumnya telah tersimpan pada *database server* dan dapat disimpulkan bahwa *database* pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah terintegrasi dengan *database server*. Sehingga hasil poin pengujian validasi integrasi data fasilitas rumah sakit dapat dilihat seperti tabel 4.13 di bawah ini:

Tabel 4.13 Tabel Hasil Poin Pengujian Integrasi Data Fasilitas

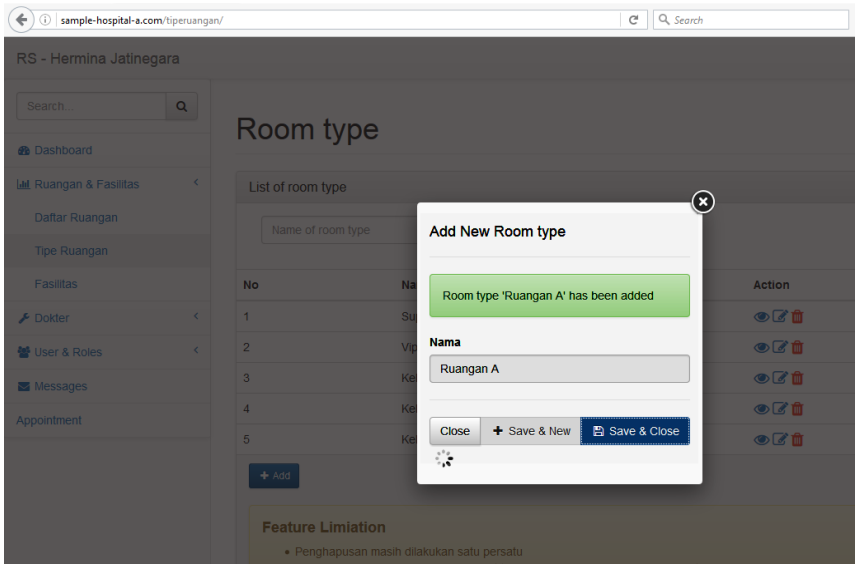
POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Validasi Integrasi Data fasilitas	Data fasilitas yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* akan tersimpan pada *database server*	Data fasilitas yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* tersimpan pada *database server*	Valid

d. Pengujian Integrasi Data Ruangan

Pengujian integrasi data ruangan memiliki dua poin pengujian, yaitu poin pengujian penambahan data ruangan dan poin pengujian validasi integrasi data ruangan. Berikut di bawah ini merupakan penjelasan dari setiap poin pengujian:

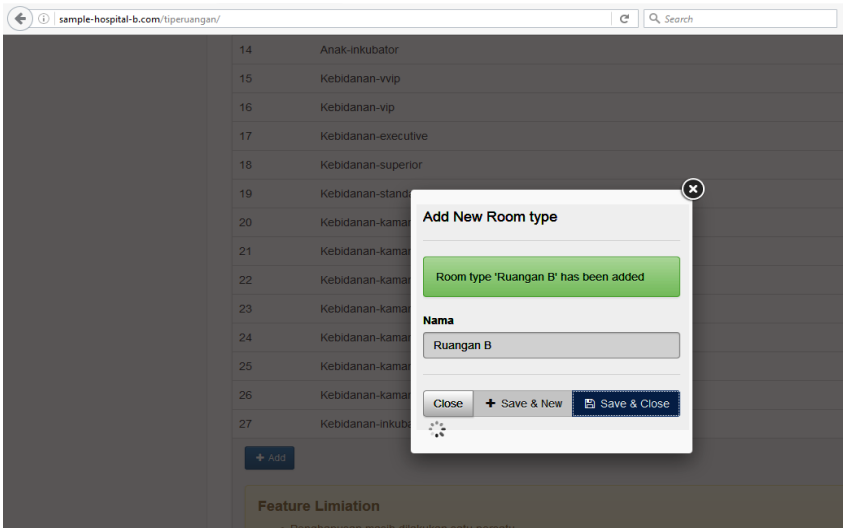
i. Poin Pengujian Penambahan Data Ruangan

Pada poin pengujian penambahan data ruangan, yang dilakukan adalah menambah data ruangan pada aplikasi *sample-hospital-a* dan aplikasi *sample-hospital-b*. Berikut di bawah ini merupakan penambahan data ruangan dengan nama ruangan "Ruangan A" pada aplikasi *sample-hospital-a*:



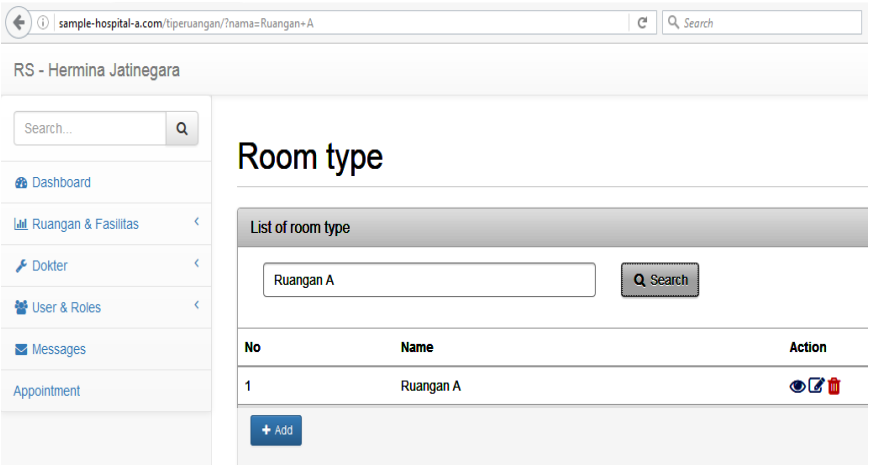
Gambar 4.17 Penambahan Data Ruangan Pada Aplikasi *Sample-Hospital-A*

Setelah melakukan penambahan data ruangan pada aplikasi *sample-hospital-a*, selanjutnya adalah melakukan penambahan data ruangan pada aplikasi *sample-hospital-b*. berikut di bawah ini merupakan gambar penambahan data ruangan dengan nama “Ruangan B” pada aplikasi *sample-hospital-b*:

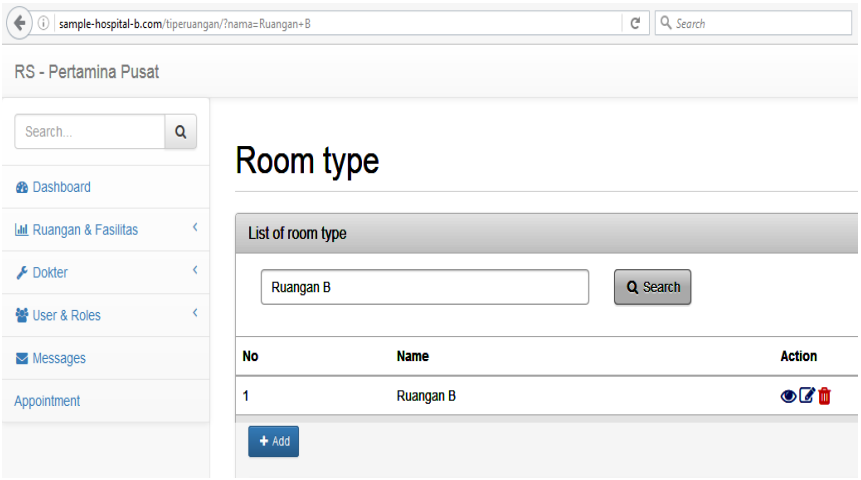


Gambar 4.18 Penambahan Data Ruangan Pada Aplikasi *Sample-Hospital-B*

Setelah menambahkan data ruangan pada aplikasi *sample-hospital-a* dan *sample-hospital-b*, hal yang perlu dilakukan adalah memeriksa apakah data ruangan yang telah ditambahkan ke masing-masing aplikasi telah tersimpan ke dalam *database*. Berikut di bawah ini merupakan gambar pencarian data ruangan yang telah ditambahkan pada *sample-hospital-a* dan *sample-hospital-b* dapat dilihat pada gambar 4.19 dan 4.20.



Gambar 4.19 Pencarian Ruangan Pada *Sample-Hospital-A*



Gambar 4.20 Pencarian Ruangan Pada *Sample-Hospital-B*

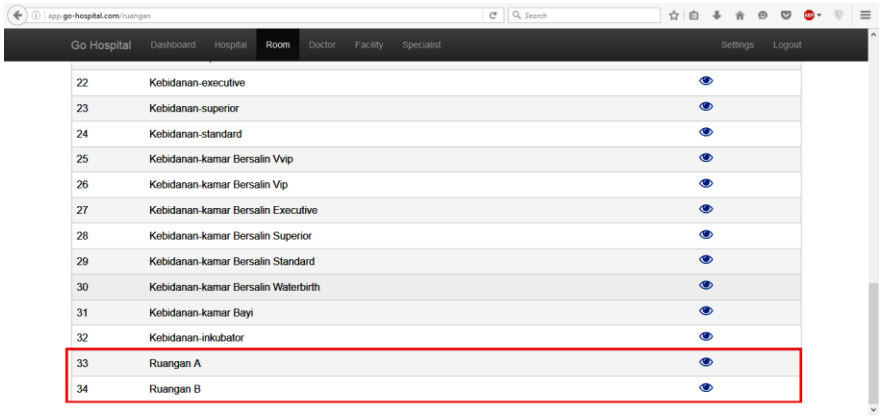
Dengan melihat gambar di atas dapat diketahui bahwa data yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah tersimpan pada masing-masing *database*. Sehingga hasil poin pengujian penambahan data ruangan dapat dilihat seperti tabel 4.14 di bawah ini:

Tabel 4.14 Tabel Hasil Poin Pengujian Penambahan Data Ruangan

POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Penambahan data ruangan	Data ruangan yang telah ditambahkan akan tersimpan ke dalam *database*	Data ruangan yang telah ditambahkan tersimpan ke dalam *database*	Diterima

ii. Poin Pengujian Validasi Integrasi Data Ruangan

Untuk melakukan pengujian validasi integrasi data ruangan dapat dilakukan dengan cara memeriksa apakah data ruangan yang telah ditambahkan sebelumnya pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah tersimpan pada *database server*. Maka hal yang perlu dilakukan adalah memeriksa *list view* ruangan yang terdapat pada API. Berikut di bawah ini merupakan *list view* ruangan yang terdapat pada API dapat dilihat pada gambar 4.22.



Gambar 4.21 *List View* Ruangan Pada API

Dengan melihat *list view* ruangan pada API seperti pada gambar 4.5 di atas, maka dapat diketahui bahwa data ruangan dengan nama "Ruangan A" dan "Ruangan B" yang telah ditambahkan pada poin pengujian sebelumnya telah tersimpan pada *database server* dan dapat disimpulkan bahwa *database* pada aplikasi *sample-hospital-a* dan *sample-hospital-b* telah terintegrasi dengan *database server*. Sehingga hasil poin pengujian validasi integrasi data ruangan dapat dilihat seperti tabel 4.15 di bawah ini:

Tabel 4.15 Tabel Hasil Poin Pengujian Integrasi Data Ruangan

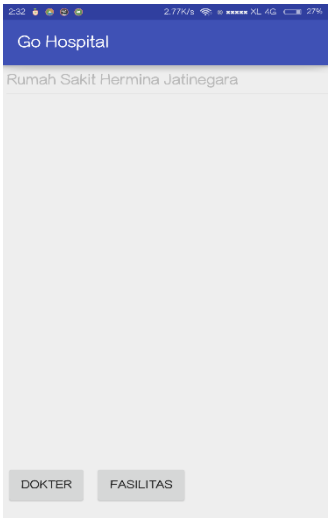
POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Validasi Integrasi Data ruangan	Data yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* akan tersimpan pada *database server*	Data yang ditambahkan pada aplikasi *sample-hospital-a* dan *sample-hospital-b* tersimpan pada *database server*	Valid

2. Hasil Pengujian Fungsionalitas Subsistem *Mobile*

   a. Pengujian Lihat Info Rumah Sakit

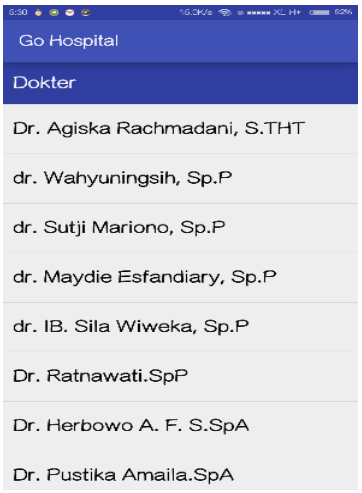
   Pengujian lihat info rumah sakit dilakukan dengan cara membandingkan informasi rumah sakit yang ada pada *platform Android* dengan informasi yang terdapat pada *server* ataupun dengan rumah sakit terkait. Pengujian lihat info

rumah sakit akan valid jika memiliki informasi yang sesuai dengan informasi yang terdapat pada *server* atau rumah sakit yang bersangkutan. Berikut di bawah ini merupakan gambar dari informasi rumah sakit yang dipilih oleh pengguna:

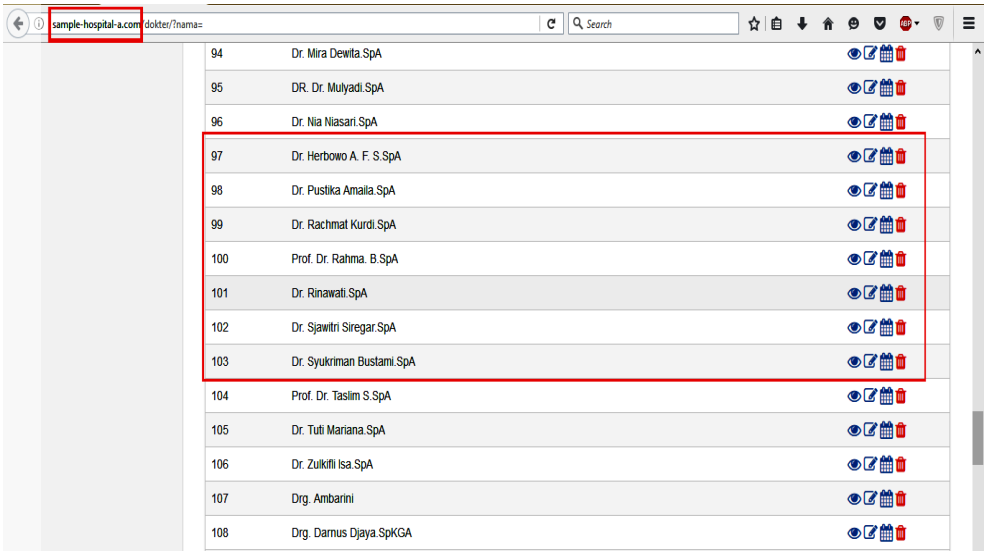


Gambar 4.22 *View* Rumah Sakit Pada *Platform Android*

Dari gambar di atas dapat diketahui terdapat 2 *button menu*, yaitu 2 dokter dan fasilitas yang jika kita *tap* pada *button menu* tersebut, maka sistem akan menampilkan daftar dokter dan daftar fasilitas yang terdapat pada rumah sakit yang dipilih sebelumnya, yaitu Rumah Sakit Hermina Jatinegara. Berikut di bawah ini merupakan contoh gambar perbandingan beberapa daftar dokter yang terdapat pada *platform Android* dan API dapat dilihat pada gambar 4.23 dan gambar 4.24.



Gambar 4.23 *List View* Dokter Pada *Platform Android*



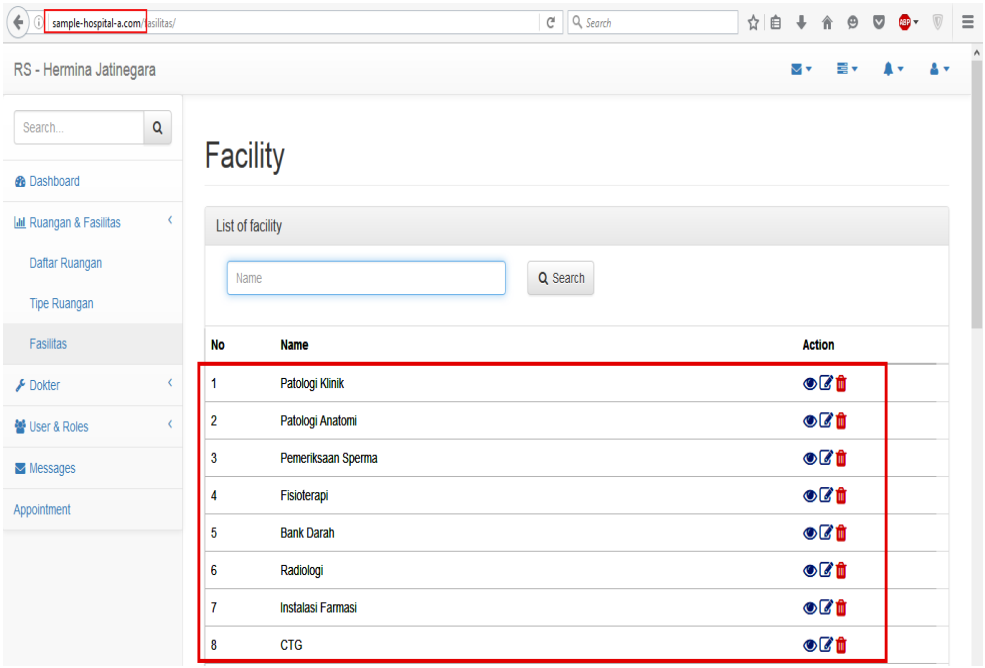
Gambar 4.24 *List View* Dokter Pada API

Dari gambar di atas dapat dilihat bahwa sistem aplikasi pelayanan rumah sakit pada *platform Android* memiliki informasi daftar dokter yang sesuai dengan informasi daftar dokter yang terdapat di *server* atau rumah sakit yang bersangkutan. Kemudian, berikut di bawah ini merupakan contoh gambar

perbandingan beberapa daftar fasilitas yang terdapat pada *platform Android* dan API dapat dilihat pada gambar 4.25 dan gambar 4.26.



Gambar 4.25 *List View* Fasilitas Pada *Platform Android*



Gambar 4.26 *List View* Fasilitas Rumah Sakit Pada API

Dari gambar di atas dapat dilihat bahwa sistem aplikasi pelayanan rumah sakit ada *platform Android* memiliki informasi daftar fasilitas yang sesuai dengan informasi daftar fasilitas yang terdapat di *server* atau rumah sakit yang

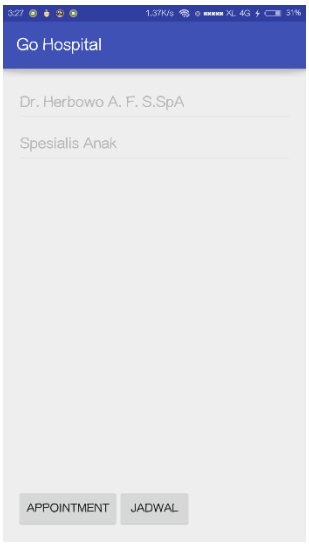
bersangkutan. Maka, dari uraian di atas dapat dilihat bahwa sistem aplikasi pelayanan rumah sakit pada *platform Android* memiliki informasi yang sesuai dengan informasi yang terdapat di *server* atau rumah sakit yang bersangkutan, dalam kasus di atas adalah dengan rumah sakit Hermina Jatinegara (*sample-hospital-a*). Sehingga dapat diberi kesimpulan bahwa fungsi lihat info rumah sakit memiliki informasi yang valid. Berikut di bawah ini adalah tabel hasil pengujian lihat info rumah sakit yang dapat dilihat pada Tabel 4.16.

Tabel 4.16 Hasil Pengujian Lihat Info Rumah Sakit

POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Menguji validitas informasi rumah sakit dengan membandingkan informasi pada sistem *Android* dengan informasi yang terdapat pada *server* atau rumah sakit terkait	Sistem *mobile android* akan menampilkan informasi rumah sakit yang sesuai dengan informasi yang terdapat pada *server* atau rumah sakit bersangkutan	Sistem *mobile android* menampilkan informasi rumah sakit yang sesuai dengan informasi yang terdapat pada *server* atau rumah sakit bersangkutan	Valid

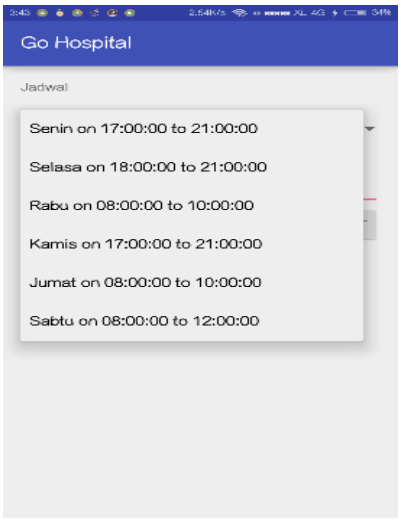
b. Pengujian Lihat Info Dokter

Pengujian lihat info dokter dilakukan dengan cara membandingkan informasi dokter yang ada pada *platform Android* dengan informasi yang terdapat pada *server* ataupun dengan rumah sakit terkait. Pengujian lihat info dokter akan valid jika memiliki informasi yang sesuai dengan informasi yang terdapat pada *server* atau rumah sakit bersangkutan. Berikut di bawah ini merupakan gambar dari informasi dokter yang dipilih oleh pengguna:

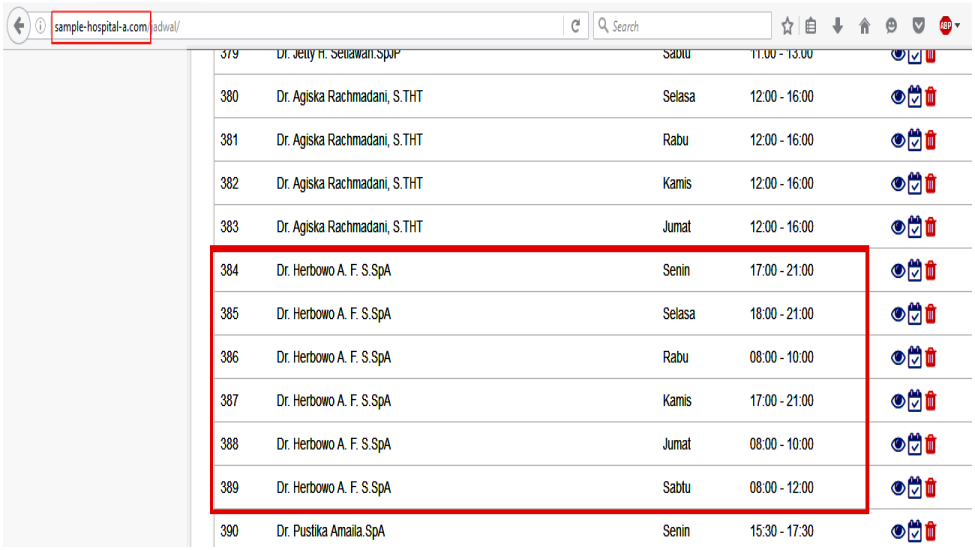


Gambar 4.27 *View* Dokter

Dari penjelasan gambar di atas, dapat dilihat bahwa setelah memilih dokter, terdapat 2 *button menu*, yaitu *appointment* dan jadwal dimana *button appointment* digunakan ketika pengguna akan melakukan *appointment* dengan dokter dan jadwal digunakan ketika pengguna ingin mengetahui jadwal praktek dokter yang dipilih. Pada pengujian ini dilakukan validitas data apakah jadwal praktek dokter yang terdapat di menu *appointment* pada *platform Android* sesuai dengan jadwal yang terdapat pada sistem informasi rumah sakit bersangkutan, dalam kasus ini merupakan Rumah Sakit Hermina Jatinegara (*sample-hospital-a*). Berikut di bawah ini merupakan perbandingan informasi jadwal praktek dari "Dr. Herbowo A. F. S.SpA" yang terdapat pada *platform Android* dengan yang terdapat pada aplikasi *sample-hospital-a* dapat dilihat pada gambar 4.28 dan 4.29.



Gambar 4.28 Informasi Jadwal Praktek Dokter Pada *Platform Android*



Gambar 4.29 Informasi Jadwal Praktek Dokter Pada API

Dilihat dari penjelasan gambar di atas dapat diketahui jika informasi jadwal praktek dokter yang terdapat pada *platform Android* dengan informasi jadwal praktek dokter yang terdapat pada aplikasi *sample-hospital-a* sudah sesuai. Sehingga dapat diberi kesimpulan bahwa fungsi lihat info dokter memiliki

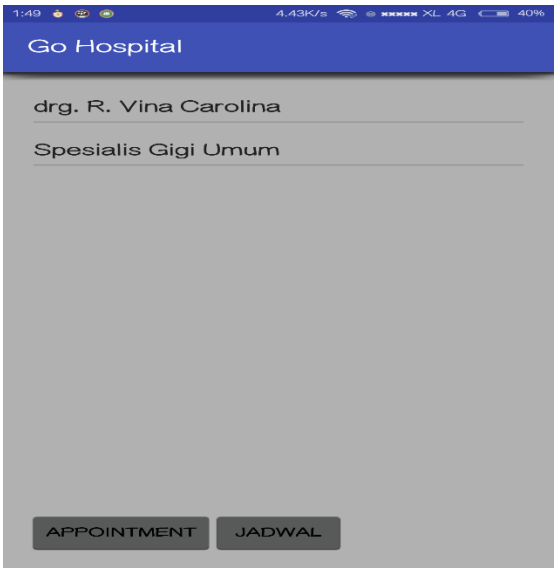
informasi yang valid. Berikut di bawah ini adalah tabel hasil pengujian lihat info dokter yang dapat dilihat pada Tabel 4.17.

Tabel 4.17 Hasil Pengujian Lihat Info Dokter

POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Menguji validitas informasi dokter dengan membandingkan informasi pada sistem *Android* dengan informasi yang terdapat pada *server* atau rumah sakit terkait	Sistem *mobile android* akan menampilkan informasi dokter yang sesuai dengan informasi yang terdapat pada *server* atau rumah sakit bersangkutan	Sistem *mobile android* menampilkan informasi rumah sakit yang sesuai dengan informasi yang terdapat pada *server* atau rumah sakit bersangkutan	Valid

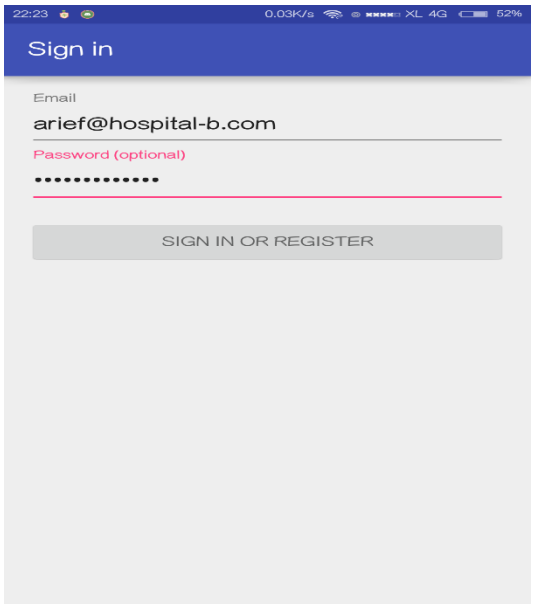
c. Pengujian *Login*

Pengujian *login* dilakukan dengan menguji fungsionalitas *login* apakah telah dapat digunakan sesuai dengan fungsinya. Proses *login* akan dilakukan ketika pengguna akan melakukan akses data ke sistem informasi rumah sakit, misalnya mencari dokter yang melakukan praktek pada suatu rumah sakit, mencari fasilitas yang ada pada suatu rumah sakit, dan membuat *appointment* dengan dokter melakukan yang praktek pada suatu rumah sakit. Berikut di bawah ini akan diuraikan contoh proses penggunaan fungsi *login* ketika pengguna akan melakukan *appointment* dengan dokter yang terdapat pada rumah sakit Rumah Sakit Pertamina Pusat (sample-hospital-b) dengan "Drg. R. Vina Carolina". Proses pertama yang dilakukan adalah pengguna memilih daftar dokter yang terdapat pada Rumah Sakit Pertamina Pusat, yaitu "Drg. R. Vina Carolina", berikut di bawah ini merupakan gambar dari lihat info dokter "Drg. R. Vina Carolina" dapat dilihat pada gambar 4.30.



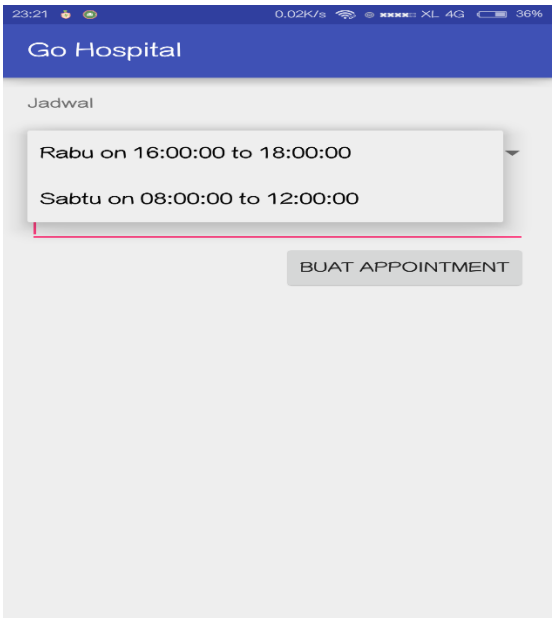
Gambar 4.30 Info Dokter "Drg. R. Vina Carolina"

Setelah melihat info dokter tersebut, kemudian pengguna memilih button *appointment* yang terdapat pada info dokter tersebut. Maka, sistem akan meminta *user* untuk melakukan *login*. Berikut di bawah ini merupakan gambar halaman *login* yang harus dilakukan oleh pengguna dapat dilihat pada gambar 4.31.



Gambar 4.31 *Login* Pengguna Pada *Platform Android*

Setelah pengguna berhasil melakukan *login*, maka akan tampil kembali halaman info dokter seperti pada gambar 4.30, kemudian ketika kita dapat melakukan pembuatan *appointment* dengan memilih tombol *appointment* yang ada pada halaman info dokter tersebut. Berikut di bawah ini merupakan gambar dari halaman pembuatan *appointment* yang dapat diakses setelah pengguna berhasil melakukan *login* dapat dilihat pada gambar 4.32.



Gambar 4.32 Halaman Pembuatan *Appointment*

Dari uraian di atas dapat dilihat bahwa fungsionalitas *login* telah berjalan dengan baik dan sesuai dengan fungsinya. Maka, dari uraian di atas pula dapat diberi kesimpulan bahwa pengujian proses *login* ini valid. Berikut di bawah ini merupakan tabel pengujian *login* yang dapat dilihat pada tabel 4.18.

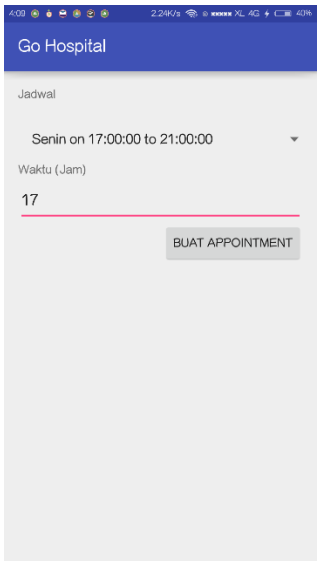
Tabel 4.18 Pengujian *Login*

POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Menguji fungsionalitas apakah sistem dapat berhasil melakukan *login*	Pengguna dapat melakukan *login* dan dapat mengakses fungsionalitas sistem yang membutuhkan *login*	Pengguna berhasil melakukan *login* dan dapat mengakses fungsionalitas pembuatan *appoinment*	Valid

d. Pengujian Pembuatan *Appointment*

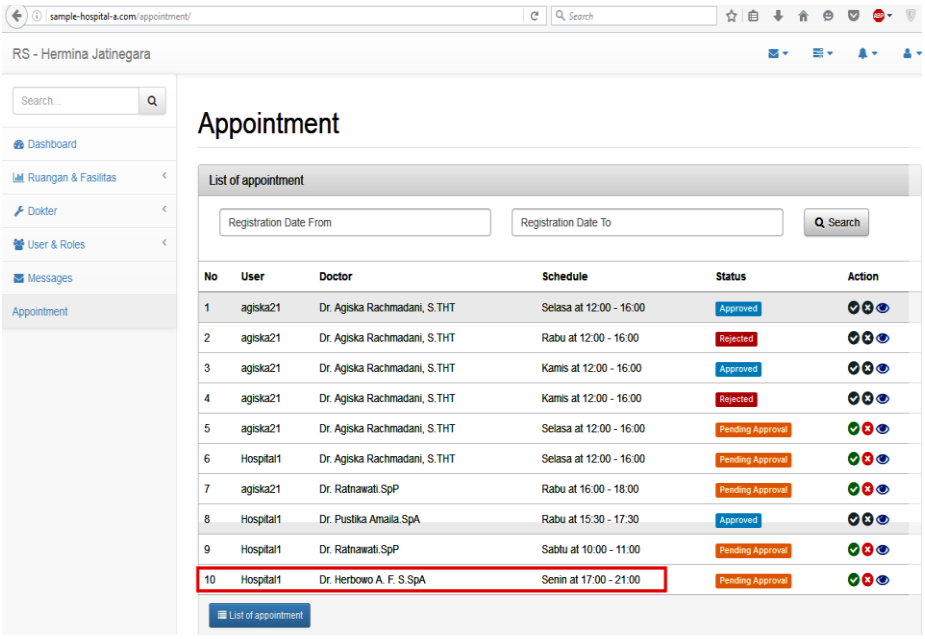
Pengujian pembuatan *appoinment* dilakukan dengan menguji validitas data *appointment* yang dikirimkan dari *mobile platform Android* ke Aplikasi rumah sakit. Pada pengujian ini akan dilakukan contoh pembuatan *appointment* dengan "Dr. Herbowo A. F. S.SpA" sebagai dokter yang terdaftar pada Rumah

Sakit Hermina Jatinegara (*sample-hospital-a*). Berikut di bawah ini merupakan pembuatan *appointment* dengan "Dr. Herbowo A. F. S.SpA" dapat dilihat pada gambar 4.33.



Gambar 4.33 Pembuatan *Appointment* Dengan Dokter

Setelah melakukan pembuatan *Appointment*, data *appointment* akan dikirimkan dari *platform Android* ke database *sample-hospital-a*. Maka, pengujian validitas data yang dilakukan adalah dengan memeriksa informasi *appointment* yang terdapat pada aplikasi *sample-hospital-a*. berikut di bawah ini merupakan *list appointment* yang terdapat pada Rumah Sakit Hermina Jatinegara dapat dilihat pada gambar 4.34.



Gambar 4.34 *List Appointment* Aplikasi *Sample-Hospital-A*

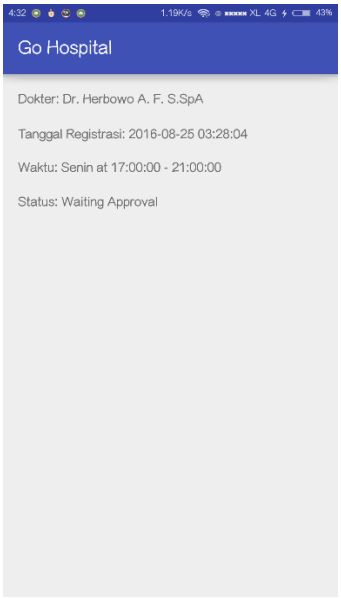
Dengan melihat gambar di atas, dapat diketahui bahwa informasi *appointment* yang terdapat pada *sample-hospital-a* sudah sesuai dengan data *appointment* yang dikirim dari *platform Android*. Maka, dapat diberi kesimpulan jika fungsi pembuatan *appointment* pada *platform Android* ini telah sesuai dan mengirimkan data yang valid. Berikut adalah hasil pengujian pembuatan *appointment* dapat dilihat pada tabel 4.19 di bawah ini:

Tabel 4.19 Pengujian Pembuatan *Appointment*

POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Menguji validitas data *appointment* yang dikirimkan dari *mobile platform* ke aplikasi rumah sakit	Aplikasi rumah sakit akan menerima data *appointment* dari sistem *platform Android*	Aplikasi rumah sakit menerima data *appointment* dari sistem *platform Android*	Valid

e. Pengujian Lihat *Appointment*

Pada pengujian lihat *appointment* dilakukan validitas data apakah *list appointment* dapat tampil sesuai dengan data *appointment* yang dibuat sebelumnya. Berikut ini merupakan fungsi lihat *appointment* yang dapat dilihat pada gambar 4.35.



Gambar 4.35 Fungsi Lihat *Appointment* Pada Sistem *Platform Android*

Berdasarkan gambar di atas, maka dapat disimpulkan bahwa data *appointment* dapat ditampilkan dan sesuai dengan data *appointment* yang telah dibuat sebelumnya. Berikut adalah hasil pengujian lihat *appointment* dapat dilihat pada tabel 4.20 di bawah ini:

Tabel 4.20 Pengujian Lihat *Appointment*

POIN PENGUJIAN	HASIL YANG DIHARAPKAN	PENGAMATAN	KESIMPULAN
Menampilkan daftar *appointment* yang telah dibuat sebelumnya	Sistem akan menampilkan informasi daftar *appointment* yang telah dibuat sebelumnya	Sistem menampilkan informasi daftar *appointment* yang telah dibuat sebelumnya	Valid