

BAB 7 PENGUJIAN

Pada bab ini akan dilakukan pengujian dan analisis terhadap sistem informasi rekam medis yang telah dikembangkan. Pengujian dilakukan dengan menggunakan 2 tipe pengujian yaitu pengujian *white-box* dan *black-box* untuk menjamin keberhasilan suatu pengembangan perangkat lunak. Pengujian *black-box* merupakan metode pengujian yang digunakan untuk memvalidasi apakah perangkat lunak berfungsi dengan baik dan benar. Sedangkan pengujian *white-box* merupakan pengujian yang digunakan untuk mengetahui algoritme atau logika dari perangkat lunak.

7.1 Pengujian *White-box*

Proses pengujian *white box* dengan menggunakan *basis path* merupakan pengujian yang mengujikan algoritme yang diterapkan atau digunakan oleh sistem. Pada pengujian ini *white box* tidak semua proses yang terdapat pada sistem informasi rekam medis pada klinik Mutiara Sehat diujikan. Pengujian ini hanya menguji beberapa proses yaitu cek kepesertaan, pendaftaran pasien dan memasukan data rekam medis hasil pemeriksaan. Berikut merupakan pengujian pengujian yang dilakukan yang dijelaskan pada sub-bab selanjutnya.

7.1.1 Pengujian Cek kepesertaan

Berikut pada tabel 7.1 merupakan algoritme untuk melakukan proses cek kepesertaan pasien.

Tabel 7.1 Algoritme fungsi cek kepesertaan

Baris	Algoritme	Node
1	BEGIN cek_kepesertaan()	1
2	Input (jenis identitas, nomor identitas)	2
3	data_peserta $\leftarrow$ call service cek_kepesertaan Webservice BPJS	3
4	Output data_peserta	4
5	data_pasien $\leftarrow$ call method cek_pasien from model pasien	5
6	If (data_pasien == null)	6
7	Output form pendaftaran blank	7
8	Else	8
9	Output form pendaftaran dan data pasien	9
10	END cek_kepesertaan()	10

Berdasarkan algoritme yang telah diperoleh seperti pada Tabel 7.1, maka diperoleh *flow graph* seperti pada Gambar 7.1. Jumlah dari *flow graph* yang telah diperoleh, maka akan ditentukan dua jalur independen dan jumlah kompleksitas (*cyclomatic complexity*), seperti berikut:

Jalur 1 : 1-2-3-4-5-6-7-10

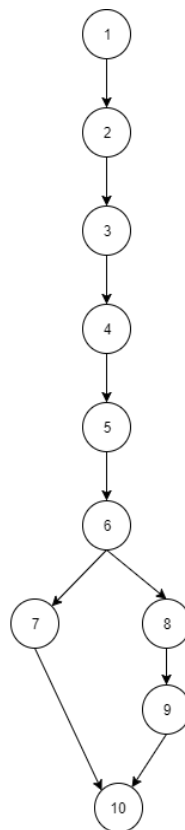
Jalur 2 : 1-2-3-4-5-6-8-9-10

Perhitungan *cyclomatic complexity*:

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 10 - 10 + 2 = 2 \end{aligned}$$

$$\begin{aligned} V(G) &= P + 1 \\ &= 1 + 1 = 2 \end{aligned}$$

$$\begin{aligned} V(G) &= R \\ &= 2 \end{aligned}$$



Gambar 7.1 *flow graph* cek kepesertaan

Berdasarkan jumlah *independent path*, maka diperoleh kasus uji sesuai pada Tabel 7.2

Tabel 7.2 Kasus Uji Melakukan Pendaftaran pasien

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Memilih jenis identitas pasien dengan NIK dan memasukan nomor NIK yang tidak belum pernah melakukan pemeriksaan sebelumnya	Menampilkan form pendaftaran blank atau dengan form inputan kosong	Menampilkan halaman form pendaftaran kosong	Valid
2	Memilih jenis identitas pasien dengan NIK dan memasukan NIK yang sudah pernah melakukan pemeriksaan sebelumnya diklinik Mutiara Sehat	Menampilkan form pendaftaran dengan data pasien yang sudah terisi oleh identitas pasien sebelumnya	Menampilkan form pendaftaran pasien dengan data pasien yang sudah terisi oleh data pendaftaran sebelumnya	Valid

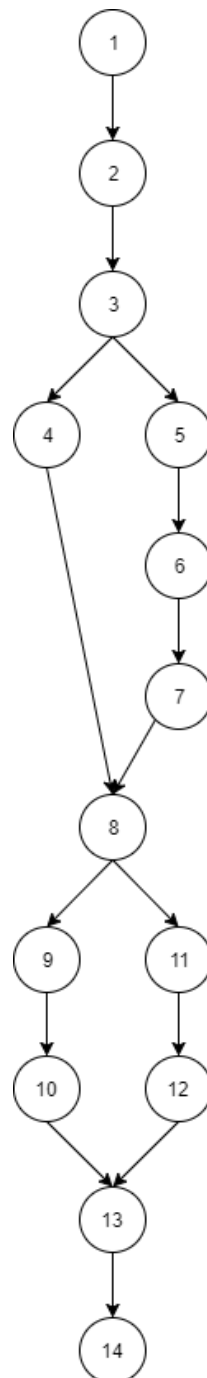
7.1.2 Pengujian Pendaftaran

Berikut ini merupakan algoritme untuk melakukan proses pendaftaran pasien pada Tabel 7.3

Tabel 7.3 Algoritme fungsi pendaftaran

Baris	Algoritme	Node
1	BEGIN pendaftaran()	1
2	Input (id pasien, tanggal pendaftaran, status peserta, nomor id, jenis id, id kelurahan, nama pasien, tanggal lahir, kelamin, nomor hp, alamat, poli tujuan, nomor rm)	2
3	If (status peserta == null premi tidak aktif)	3
4	Jenis_pasien ← UMUM	4
5	Else	5
6	jenis pasien ← BPJS	6
7	call service set_pendaftaran Webservice BPJS	7
8	If (id_pasien== null)	8
9	Call method set_pasien from model pasien	9
10	Call method set_pendaftaran from model pendaftaran	10
11	Else	11
12	Call method set_pendaftaran from model pendaftaran	12
13	Output pendaftaran	13
14	End pendaftaran()	14

Berdasarkan algoritme pendaftaran yang diperoleh dari seperti pada Tabel 7.2 maka diperoleh *flow graph* seperti pada Gambar 7.2



Gambar 7.2 *flow graph* pendaftaran

Berdasarkan jumlah *flow graph* yang telah diperoleh, maka dapat ditentukan empat jalur independen dan jumlah *cyclomatic complexity*, seperti berikut:

Jalur 1 :1-2-3-5-6-7-8-9-10-13-14

Jalur 2 :1-2-3-5-6-7-8-11-12-13-14

Jalur 3 :1-2-3-4-8-9-10-13-14

Jalur 4 :1-2-3-4-8-11-12-13-14

Perhitungan *cyclomatic complexity*:

$$\begin{aligned}V(G) &= E - N + 2 \\ &= 16 - 14 + 2 = 3\end{aligned}$$

$$\begin{aligned}V(G) &= P + 1 \\ &= 3 + 1\end{aligned}$$

$$\begin{aligned}V(R) &= R \\ &= 4\end{aligned}$$

Berdasarkan jumlah *independent path*, maka diperoleh kasus uji sesuai pada Tabel 7.4

Tabel 7.4 kasus uji pendaftaran pasien

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Pasien yang melakukan pendaftaran merupakan pasien BPJS namun belum pernah melakukan pemeriksaan diklinik Mutiara Sehat	Sistem menyimpan data pasien dan identifikasi pasien jenis BPJS	Sistem menyimpan data pasien dan mengidentifikasi pasien jenis BPJS	Valid
2	Pasien yang melakukan pendaftaran merupakan pasien BPJS namun sudah pernah melakukan pendaftaran sebelumnya	Sistem menyimpan data pendaftaran pasien dengan identifikasi pasien jenis BPJS	Sistem menyimpan data pendaftaran pasien dengan identifikasi pasien jenis BPSJ	Valid

Tabel 7.4 kasus uji pendaftaran pasien(lanjutan)

Jalur	Kasus Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
3	Pasien yang melakukan pendaftaran merupakan pasien umum dan belum pernah melakukan pendaftaran sebelumnya	Sistem mengidentifikasi jenis pasien umum dan menyimpan data pendaftaran serta data pasien	Sistem mengidentifikasi jenis pasien umum dan menyimpan data pendaftaran serta data pasien	Valid
4	Pasien yang melakukan pendaftaran adalah pasien jenis umum dan pernah melakukan pemeriksaan sebelumnya	Sistem mengidentifikasi jenis pasien umum dan hanya menyimpan data pendaftaran pasien	Sistem mengidentifikasi jenis pasien umum dan hanya menyimpan data pendaftaran	Valid

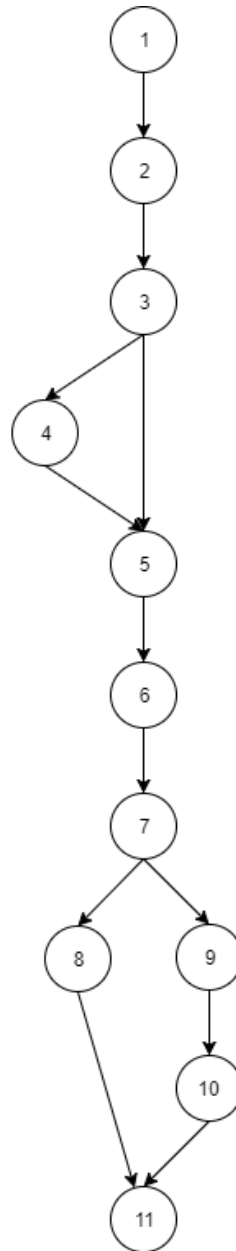
7.1.3 Pengujian Memasukan rekam medis

Berikut ini merupakan algoritme untuk proses melakukan memasukan rekam medis pada tabel 7.5

Tabel 7.5 Algoritme masukan rekam medis

Baris	Algoritme	Node
1	Begin masukan_rekam_medis()	1
2	Call (id pendaftaran, nomor identitras, diagnosa, kesadan, berat badan, tinggi badan, sistole, diastole, respiratory rate, hearth rate, dokter, terapi, keluhan, status_pulang, id bpjs)	2
3	if (id bpjs != null)	3
4	Call service set_rekam_medis Webservice BPJS	4
5	Call method set_rekam_medis from model rekam_medis	5
6	Call method set_diagnosa from diagnosa model	6
7	If (status_pulang== Rujuk lanjut)	7
8	Call method set_rujuk_lanjut from rujukan model	8
9	Elseif (status_pulang== Rujuk balik)	9
10	Call method rujuk_balik from rujukan model	10
11	End masukan_rekam_medis()	11

Berdasarkan algoritme yang telah diperoleh seperti pada tabel 7.5 maka akan diperoleh *flow graph* seperti pada Gambar 7.3



Gambar 7.3 *flow graph* memasukan rekam medis

Berdasarkan jumlah *flow graph* yang telah diperoleh, maka akan ditentukan empat jalur independen dan *cyclomatic complexity*, seperti berikut:

Jalur 1 :1-2-3-5-6-7-8-11

Jalur 2 :1-2-3-4-5-6-7-8-11

Jalur 3 :1-2-3-4-5-6-7-9-10-11

Jalur 4 :1-2-3-5-6-7-9-10-11

Perhitungan *cyclomatic complexity*:

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 13 - 11 + 2 = 4 \\ V(G) &= P + 1 \\ &= 3 + 1 = 4 \\ V(G) &= R \\ &= 4 \end{aligned}$$

Berdasarkan jumlah *independent path*, maka diperoleh kasus uji sesuai pada Tabel 7.6.

Tabel 7.6 Kasus Uji Memasukan rekam medis

Jalur	Kasus uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Rekam medis yang dimasukan dari pasien BPJS dan dalam hasil pemeriksaan memerlukan rujukan lanjut	Sistem menyimpan rekam medis ke service BPJS kesehatan dan menyimpan rujukan lanjut	Sistem menyimpan rekam medis ke service BPJS kesehatan dan menyimpan rujukan lanjut	Valid
2	Rekam medis yang dimasukan merupakan pasien umum dan dalam pemeriksaan memerlukan rujukan lanjut	Sistem tidak menyimpan rekam medis ke service BPJS dan menyimpan rujukan lanjut	Sistem tidak menyimpan rekam medis ke service BPJS dan menyimpan rujukan lanjut	Valid
3	Rekam medis yang dimasukan merupakan pasien BPJS dan memerlukan rujukan balik	Sistem rekam medis ke service BPJS kesehatan dan menyimpan rujukan balik	Sistem rekam medis ke service BPJS kesehatan dan menyimpan rujukan balik	Valid
4	Rekam medis yang dimasukan merupakan pasien umum dan dalam pemeriksaan memerlukan rujukan balik	Sistem tidak menyimpan rekam medis ke service BPJS dan menyimpan rujukan balik	Sistem tidak menyimpan rekam medis ke service BPJS dan menyimpan rujukan balik	Valid

7.2 Pengujian *Black-Box*

Pengujian black-bos bertujuan untuk memastikan setiap fungsional yang telah didefinisikan mampu dipenuhi oleh sistem yang dibangun. Setiap persyaratan fungsional yang telah didefinisikan pada bagian analisis persyaratan akan diuji pada pengujian ini. Pengujian akan dilakukan dengan mendefinisikan kasus uji (*test case*) yang didasarkan pada skenario use case yang telah didefinisikan pada bagian analisis kebutuhan. Kemudian setiap hasil pengujian yang dilakukan akan dibandingkan dengan hasil telah didefinisikan sebelumnya pada *Use case scenario*.

7.2.1 Pengujian cek kepesertaan pasien

Tabel 7.7, 7.8, 7.9 menjelaskan kasus uji dari pengujian validasi yang merujuk pada *use case* cek kepesertaan pasien.

Tabel 7.7 Kasus uji cek kepesertaan pasien

Nomor Kasus Uji	UJ-01
Nama Kasus Uji	Kasus uji cek kepesertaan pasien
Kode Kebutuhan	SKPL-F-06
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan cek kepesertaan pasien
Test Case	Pengujian dengan memasukan nomor identitas pasien dan jenis identitas pasien
Prosedur Uji	1. Penguji memilih menu pendaftaran 2. Memilih jenis identitas pasien 3. Memasukan nomor identitas pasien 4. Menekan tombol cek kepesertaan
Hasil yang diharapkan	Menampilkan form pendaftaran pasien, form identitas pasien dan data hasil cek kepesertaan pasien
Hasil Pengujian	Menampilkan form pendaftaran pasien, form pendaftaran pasien yang sudah terisi oleh data pasien dan data hasil cek kepesertaan pasien
Status Validasi	Valid

Tabel 7.8 Kasus uji cek kepesertaan pasien: *expection 3a*

Nomor Kasus Uji	UJ-02
Nama Kasus Uji	Kasus uji cek kepesertaan pasien <i>expection 3a</i>
Kode Kebutuhan	SKPL-F-06

Tabel 7.8 Kasus uji cek kepesertaan pasien: *exception 3a*(lanjutan)

Tujuan Pengujian	Pengujian dilakukan untuk memastikan jika nomor identitas yang dilakukan pengujian tidak terdaftar kepesertaan BPJS maka sistem tidak menampilkan hasil cek kepesertaan BPJS
Test Case	Pengujian dengan memasukan nomor identitas pasien dan jenis identitas pasien yang tidak terdaftar kepesertaan BPJS
Prosedur Uji	1. Penguji memilih menu pendaftaran 2. Memilih jenis identitas pasien 3. Memasukan nomor identitas pasien 4. Menekan tombol cek kepesertaan
Hasil yang diharapkan	Sistem dapat menampilkan form pendaftaran pasien, form identitas pasien dan menampilkan hasil cek kepesertaan BPJS dengan pesan pasien tidak terdaftar
Hasil Pengujian	Menampilkan form pendaftaran, form identitas pasien dan menampilkan hasil cek kepesertaan BPJS dengan pesan pasien tidak terdaftar
Status Validasi	Valid

Tabel 7.9 Kasus uji cek kepesertaan pasien: *exception 3b*

Nomor Kasus Uji	UJ-03
Nama Kasus Uji	Kasus uji cek kepesertaan pasien <i>exception 3b</i>
Kode Kebutuhan	SKPL-F-03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan jika nomor identitas yang dilakukan pengujian tidak terdapat pada sistem maka sistem menampilkan form identitas pasien kosong
Test Case	Pengujian dengan memasukan nomor identitas pasien yang tidak terdapat pada sistem
Prosedur Uji	1. Penguji memilih menu pendaftaran 2. Memilih jenis identitas pasien 3. Memasukan nomor identitas pasien 4. Menekan tombol cek kepesertaan

Tabel 7.9 Kasus uji cek kepesertaan pasien: *expectation 3b*(lanjutan)

Hasil yang diharapkan	Sistem dapat menampilkan form pendaftaran pasien, form identitas pasien kosong dan hasil cek kepesertaan BPJS kesehatan
Hasil Pengujian	Menampilkan form pendaftaran pasien, form identitas pasien kosong dan hasil cek kepesertaan BPJS kesehatan
Status Validasi	Valid

7.2.2 Pengujian pendaftaran pasien

Tabel 7.10, 7.11, 7.12 menjelaskan kasus uji dari pengujian validasi yang merujuk pada use case pendaftaran pasien.

Tabel 7.10 Kasus uji pendaftaran pasien

Nomor Kasus Uji	UJ-04
Nama Kasus Uji	Kasus uji pendaftaran pasien
Kode Kebutuhan	SKPL-F-03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat melakukan pendaftaran pasien
Test Case	Pengujian dengan memasukan data pasien dan data pendaftaran pasien sesuai dengan format yang ditentukan
Prosedur Uji	1. Penguji memilih menu pendaftaran 2. Penguji melakukan cek kepesertaan 3. Penguji mengisi <i>form</i> pendaftaran pasien 4. Penguji menekan tombol simpen
Hasil yang diharapkan	Pendaftaran pasien tersimpan di <i>database</i> dan dapat dilihat pada tabel pendaftaran pasien
Hasil Pengujian	Pendaftaran pasien tersimpan dalam <i>database</i> dan data pasien yang mendaftar dapat dilihat pada tabel pendaftaran pasien dengan jenis pasien umum
Status Validasi	Valid

Tabel 7.11 Kasus uji pendaftaran pasien: *expectation 6a*

Nomor Kasus Uji	UJ-05
Nama Kasus Uji	Kasus uji pendaftaran pasien <i>expectation 6a</i>
Kode Kebutuhan	SKPL-F-03

Tabel 7.11 Kasus uji pendaftaran pasien: *expectation 6a*(lanjutan)

Tujuan Pengujian	Pengujian dilakukan untuk memastikan jika pasien yang melakukan pendaftaran merupakan pasien BPJS aktif maka, sistem mengirimkan data pendaftaran pasien ke <i>web service</i> BPJS
Test Case	Pengujian dengan memasukan data pendaftaran pasien BPJS kesehatan aktif
Prosedur Uji	1. Penguji memilih menu pendaftaran 2. Penguji melakukan cek kepesertaan 3. Penguji mengisi <i>form</i> pendaftaran pasien 4. Penguji menekan tombol simpen
Hasil yang diharapkan	Data pendaftaran pasien tersimpan di <i>database</i> dan terkirim ke <i>web service</i> BPJS serta menampilkan tabel pendaftaran.
Hasil Pengujian	Data pendaftaran pasien tersimpan dalam <i>database</i> dan terkirim ke <i>web service</i> BPJS serta data pasien yang mendaftar dapat dilihat pada tabel pendaftaran pasien dengan jenis pasien BPJS
Status Validasi	Valid

Tabel 7.12 Kasus uji pendaftaran pasien: *expectation 6b*

Nomor Kasus Uji	UJ-06
Nama Kasus Uji	Kasus uji pendaftaran pasien <i>expectation 6b</i>
Kode Kebutuhan	SKPL-F-03
Tujuan Pengujian	Pengujian dilakukan untuk memastikan jika data identitas pasien yang melakukan pendaftaran tidak terdapat pada sistem maka sistem menyimpan data identitas pasien
Test Case	Pengujian dengan memasukan data pendaftaran pasien yang belum pernah datang
Prosedur Uji	1. Penguji memilih menu pendaftaran 2. Penguji melakukan cek kepesertaan 3. Penguji mengisi <i>form</i> data identitas pasien 4. Penguji mengisi <i>form</i> pendaftaran pasien 5. Penguji menekan tombol simpen

Tabel 7.12 Kasus uji pendaftaran pasien: *expectation 6b*(lanjutan)

Hasil yang diharapkan	Pendaftaran pasien tersimpan dalam <i>database</i> dan menyimpan data pasien serta menampilkan tabel pendaftaran.
Hasil Pengujian	Data pendaftaran pasien dan data identitas pasien tersimpan dalam <i>database</i> serta data pasien yang mendaftar dapat dilihat pada tabel pendaftaran pasien dengan jenis pasien BPJS
Status Validasi	Valid

7.2.3 Pengujian memasukan rekam medis

Tabel 7.13, 7.14, 15 menjelaskan kasus uji dari pengujian validasi yang merujuk pada *use case* memasukan data rekam medis.

Tabel 7.13 Kasus uji memasukan data rekam medis

Nomor Kasus Uji	UJ-07
Nama Kasus Uji	Kasus uji memasukan data rekam medis
Kode Kebutuhan	SKPL-F-07
Tujuan Pengujian	Pengujian dilakukan untuk memastikan data rekam medis dapat berhasil disimpan
Test Case	Pengujian dengan memasukan data rekam medis hasil pemeriksaan salah satu pasien
Prosedur Uji	1. Penguji memilih menu rekam medis 2. Penguji memilih pasien 3. Penguji menekan tombol tampilkan 4. Penguji mengisi <i>form</i> rekam medis hasil pemeriksaan pasien 5. Penguji menekan tombol simpan
Hasil yang diharapkan	Data rekam medis berhasil disimpan pada <i>database</i>
Hasil Pengujian	Data rekam medis berhasil disimpan pada <i>database</i>
Status Validasi	Valid

Tabel 7.14 Kasus uji memasukan data rekam medis: *expectation 3b*

Nomor Kasus Uji	UJ-08
Nama Kasus Uji	Kasus uji memasukan data rekam medis <i>expectation 3b</i>

Tabel 7.14 Kasus uji memasukkan data rekam medis: *expectation 3b*(lanjutan)

Kode Kebutuhan	SKPL-F-07
Tujuan Pengujian	Pengujian dilakukan untuk memastikan pada saat pengujian memilih pasien maka form data primer pasien sudah terisi jika pasien sudah pernah berkunjung sebelumnya
Test Case	Pengujian dengan memilih pasien yang sudah pernah berkunjung sebelumnya
Prosedur Uji	1. Penguji memilih menu rekam medis 2. Penguji memilih pasien 3. Penguji menekan tombol tampilkan
Hasil yang diharapkan	Menampilkan form rekam medis dengan form data primer sudah terisi
Hasil Pengujian	Sistem menampilkan form rekam medis data primer pasien sudah terisi
Status Validasi	Valid

Tabel 7.15 Kasus uji memasukkan data rekam medis: *expectation 5a*

Nomor Kasus Uji	UJ-09
Nama Kasus Uji	Kasus uji memasukkan data rekam medis <i>expectio 5a</i>
Kode Kebutuhan	SKPL-F-07
Tujuan Pengujian	Pengujian dilakukan untuk memastikan jika data rekam medis hasil pemeriksaan merupakan pasien BPJS maka sistem mengirim data rekam medis kedalam sistem
Test Case	Pengujian dengan memasukkan data rekam medis hasil pemeriksaan pasien yang merupakan pasien kepesertaan BPJS kesehatan
Prosedur Uji	1. Penguji memilih menu rekam medis 2. Penguji memilih pasien 3. Penguji menekan tombol tampilkan 4. Penguji mengisi <i>form</i> rekam medis hasil pemeriksaan pasien 5. Penguji menekan tombol simpan
Hasil yang diharapkan	Data rekam medis berhasil disimpan pada <i>database</i> dan dikirim ke <i>web service</i> BPJS Kesehatan

Tabel 7.15 Kasus uji memasukan data rekam medis: *expection 5a*(lanjutan)

Hasil Pengujian	Data rekam medis berhasil disimpan pada <i>database</i> serta data tersimpan di BPJS Kesehatan melalui <i>web service</i> BPJS kesehatan
Status Validasi	Valid

7.3 Pengujian *Compatibility*

Pengujian kompabilitas dilakukan dengan menggunakan SortSite versi 5.22.1766. Dari pengujian ini dihasilkan versi *browser* apa saja yang mendukung berjalannya sistem informasi Rekem Medis dengan baik. Pada Tabel 7.20 dapat dilihat *browser* beserta versinya yang digunakan dalam uji kompabilitas.

Tabel 7.1 *Browser* yang digunakan pada *compatibility testing*

No	Nama <i>Browser</i>	Versi <i>Browser</i>
1	Microsoft Edge	14
2	Firefox	51
3	Safari	<= 9.0, dan 10
4	Opera	41
5	Chrome	55
6	iOS	<= 8.0 , 9, 10
7	Android	<= 3.0 dan 4.0
8	Internet Explorer	8, 9, 10, 11

Browser	Internet Explorer				Edge	Firefox	Safari		Opera	Chrome	iOS			Android	
Version	8	9	10	11	14	51	≤ 9	10	41	55	≤ 8	9	10	≤ 3	4*
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	⚠	⚠	⚠	⚠	✓	⚠	⚠	⚠	✓	⚠	✓	✓	✓	✓	✓

Gambar 7.4 Hasil *compatibility testing*

Terdapat 3 kategori masalah yang dapat diketahui dari hasil pengujian kompabilitas menggunakan SortSite, yakni *critical Issues*, *major issues*, dan *minor issues*. Penejelasan dari ketiga kategori masalah dapat dilihat pada Gambar 7.5.

✓	Tidak ada kendala pada browser
✗	Critical Issues : Terdapat konten atau fungsi yang hilang pada beberapa browser
⚠	Major Issues : Terdapat masalah utama pada konten atau performa pada beberapa browser
⚠	Minor Issues : Terdapat masalah minor pada konten atau performa pada beberapa browser

Gambar 7.5 Kategori *compatibility issues* pada SortSite

Sumber : Diadaptasi Dari Power Mapper (2016)

Dari hasil pengujian kompatibilitas pada Gambar 7.4 didapatkan bahwa *website* Sistem Informasi Rekam Medis yang dibuat dapat berjalan hampir pada semua *browser* yang di uji kecuali di internet *explorer* versi 8 yang didapatkan masalah utama.