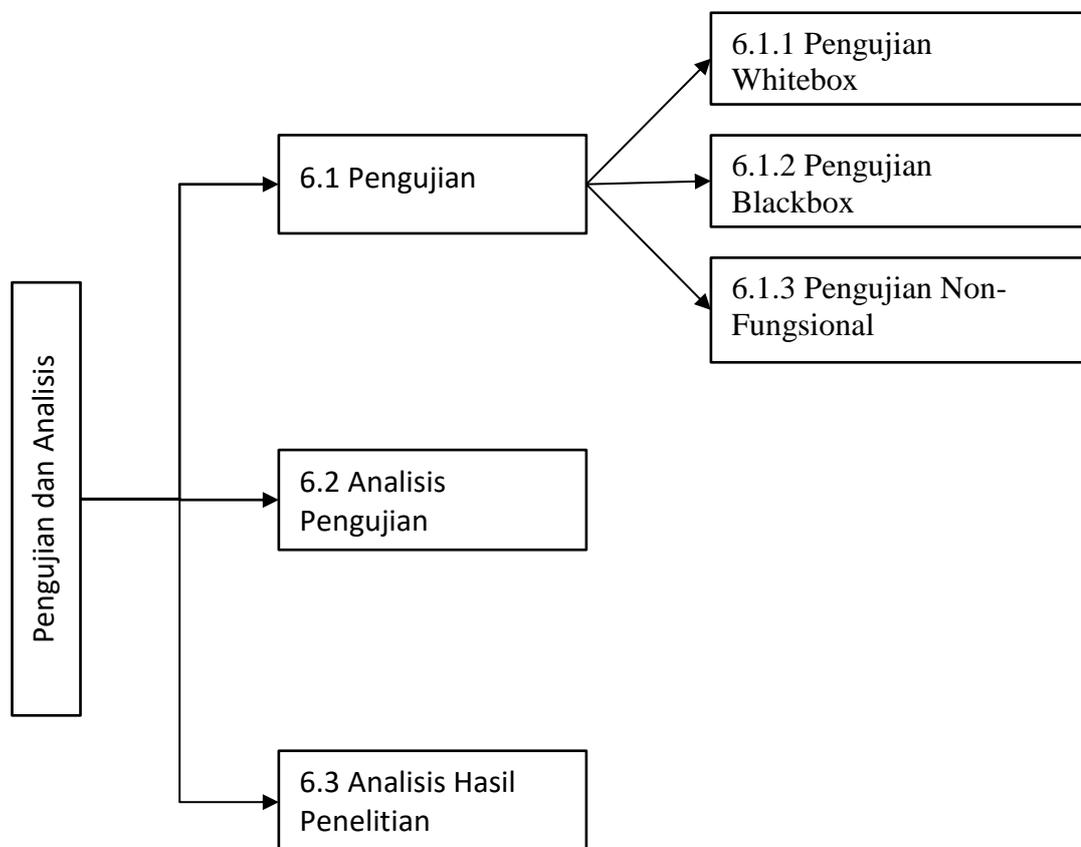


BAB 6 PENGUJIAN DAN ANALISIS

Setelah melakukan implementasi sistem maka hal selanjutnya yang dilakukan adalah menguji dan menganalisis sistem yang sudah diimplementasi. Pengujian yang dilakukan pada penelitian ini yaitu menguji kebutuhan fungsional dan kebutuhan non-fungsional. Dalam gambar 6.1 merupakan diagram pohon untuk penjelasan isi dari bab ini.



Gambar 0.1 Diagram Pohon Pengujian dan Analisis

6.1 Pengujian Fungsional

Pengujian perangkat lunak merupakan suatu elemen yang penting dan kritis untuk menjamin kualitas perangkat lunak. Pengujian dilakukan untuk mengurangi kesalahan sedini mungkin, meningkatkan kualitas dan mempertahankan kepuasan *customer*. Pada penelitian ini dilakukan pengujian fungsional dan non-fungsional. Pengujian fungsional akan dilakukan *unit testing* dan *functional testing*, sedangkan untuk pengujian non-fungsional akan dilakukan *performance testing*. Pada pengujian fungsional akan menggunakan teknik pengujian *whitebox testing* dan *blackbox testing*.

6.1.1 Pengujian *Whitebox*

Pengujian *whitebox* merupakan salah satu dari dua bagian pendekatan *box-testing*. Pada pengujian *whitebox* dilakukan dengan cara melihat ke dalam modul (pengujian internal) dengan meneliti *source code* program lalu dilakukan analisis apakah ada kesalahan atau tidak. Skenario pengujian *whitebox* yang dilakukan pada laporan penelitian ini mengambil 3 contoh yang akan diuji algoritmenya antara lain *login*, *approve saji* dan tambah transaksi.

6.1.1.1 Pengujian *unit login*

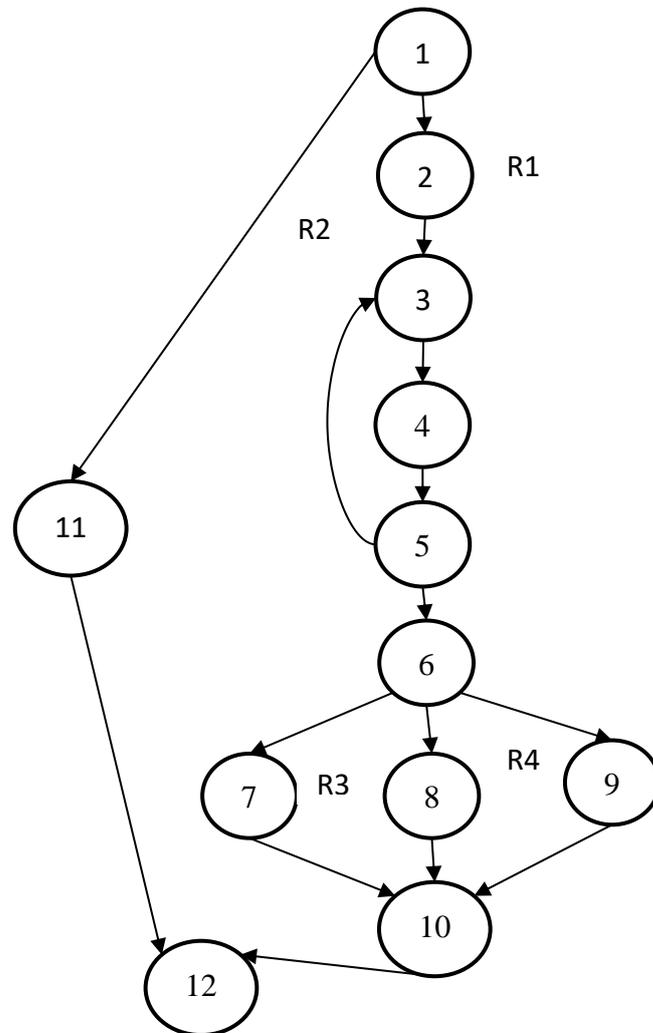
Pada fungsi *login* merupakan fungsi yang dapat masuk kedalam sistem yang dapat dilakukan oleh *user*. Pada Tabel 6.1 menampilkan algoritme dari fungsi *login*.

Kode: POS-KF-001

Tabel 0.1 Algoritme *login*

1	FUNCTION auth()	
2	\$username <- POST('username')	
3	\$password <- POST('password')	
4	\$where <- ARRAY('username','password')	1
5	\$ccek <- LOAD model->cek_login("pengguna",	
6	\$where)	
7	\$akses<-LOAD model->cek_akses("pengguna",	
8	\$where)	
8	IF \$cek > 0	2
9	FOREACH(\$akses as \$row)	3
10	\$data_session <- ARRAY('id','nama',	4
11	'status','level','outlet')	
12	\$this <- set_userdata	5
13	IF userdata('level'= 'admin')	6

	Redirect "admin"	7
14	ELSEIF userdata('level' = 'kasir')	8
15	Redirect "kasir"	
16	ELSEIF userdata('level' = 'dapur')	9
17	Redirect "dapur"	
18	END IF	10
19	ELSE	
20	Load view	11
21	ECHO "Maaf, kombinasi username dengan	
22	password salah."	
23	END IF	12
24		



Gambar 0.2 Flow graph algoritme login

Dalam tabel yang ditunjukkan pada Tabel 6.1 merupakan pembahasan algoritme *login*. Pada algoritme yang tertampil terdapat angka-angka yang menunjukkan bahwa algoritme tersebut memiliki node-node sebanyak angka yang ditunjukkan, pada Tabel 6.1 terdapat 12 node yang akan diubah ke bentuk *flow graph*. Dalam gambar yang diperlihatkan pada Gambar 6.2 merupakan *flow graph* algoritme untuk *login* yang akan dilakukan perhitungan *cyclometric complexity* yang sudah dibahas pada landasan kepastakaan. Berikut *perhitungan cyclometric complexity* pada *flow graph* algoritme *login*.

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 14 - 12 + 2 \\ &= 4 \end{aligned}$$

$$\begin{aligned} V(G) &= P + 1 \\ &= 3 + 1 \\ &= 4 \end{aligned}$$

$$\begin{aligned} V(G) &= \text{Jumlah region} \\ &= 4 \end{aligned}$$

Dari perhitungan yang sudah dihitung didapatkan sebanyak 4 jalur independen, antara lain:

6.1.1.1.1.1.1 1 – 2 – 11 – 12

6.1.1.1.1.1.2 1 – 2 – 3 – 4 – 5 – 6 – 7 – 10 – 12

6.1.1.1.1.1.3 1 – 2 – 3 – 4 – 5 – 6 – 8 – 10 – 12

6.1.1.1.1.1.4 1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 12

Dua jalur independen yang sudah dibuat selanjutnya dilakukan pengujian melalui *test case* atau kasus uji di masing-masing jalurnya. *Test case* pada tiap-tiap jalur independen didefinisikan pada Tabel 6.2.

Tabel 0.2 Test case algoritme login

No	Jalur	Data Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	1 – 2 – 11 – 12	Username = adm Password = admin123	Tidak dapat masuk kedalam sistem dan simtem memberitahu bahwa username dan	Tampil halaman login dan muncul peringatan “Maaf, kombinasi username dengan	Valid

No	Jalur	Data Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
			password salah	password salah.”	
2	1 – 2 – 3 – 4 – 5 – 6 – 7 – 10 – 12	Username = adm Password = adm	Pengguna dapat masuk ke dalam halaman admin	Berhasil masuk ke dalam halaman admin	Valid
3	1 – 2 – 3 – 4 – 5 – 6 – 8 – 10 – 12	Username = ela Password = ela	Pengguna dapat masuk ke dalam halaman kasir	Berhasil masuk ke dalam halaman kasir	
4	1 – 2 – 3 – 4 – 5 – 6 – 9 – 10 – 12	Username = ucik Password = ucik	Pengguna dapat masuk ke dalam halaman dapur	Berhasil masuk ke dalam halaman dapur	

6.1.1.2 Pengujian unit approve saji

Pada fungsi *approve saji* merupakan fungsi yang dapat merubah antrian pesanan menjadi antrian yang siap disajikan, fungsi ini hanya dapat dilakukan oleh pengguna dapur. Pada Tabel 6.3 menampilkan algoritme dari fungsi *approve saji*.

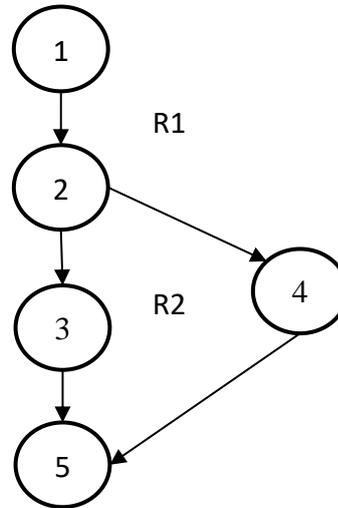
Kode: POS-KF-027

Tabel 0.3 Algoritme *approve saji*

1	FUNCTION queue_saji()	
2	\$Id <- INPUT ('id')	1
3	Load model	
4	IF LOAD model<-update(selesai)use \$id	2
5	LOAD library<-publish(queue, saji updated)	3
6	Echo 'pesanan berhasil disajikan'	
7	ELSE	
8	Echo 'pesanan gagal disajikan'	4
9	END IF	5

Dalam tabel yang ditunjukkan pada Tabel 6.3 merupakan pembahasan algoritme *approve saji*. Pada algoritme yang tertampil terdapat angka-angka yang menunjukkan bahwa algoritme tersebut memiliki node-node sebanyak angka yang ditunjukkan, pada Tabel 6.3 terdapat 5 node yang akan diubah ke bentuk *flow*

graph. Dalam Gambar 6.3 menggambarkan *flow graph approve* saji yang dibuat dari algoritme pada Tabel 6.3.



Gambar 0.3 Flow graph algoritme approve saji

Dalam gambar yang diperlihatkan pada Gambar 6.3 merupakan flow graph algoritme untuk menyetujui pesanan saji yang akan dilakukan perhitungan cyclometric complexity yang sudah dibahas pada landasan kepustakaan. Berikut perhitungan *cyclometric complexity* pada flow graph algoritme *approve saji*.

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 5 - 5 + 2 \\
 &= 2
 \end{aligned}$$

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

$$\begin{aligned}
 V(G) &= \text{Jumlah region} \\
 &= 2
 \end{aligned}$$

Dari perhitungan *cyclometric complexity* pada flow graph algoritme *approve saji* didapatkan sebanyak 2 jalur independen, antara lain:

6.1.1.2.1.1.1 1 – 2 – 3 – 5

6.1.1.2.1.1.2 1 – 2 – 4 – 5

Dua jalur independen yang sudah dibuat selanjutnya dilakukan pengujian melalui *test case* atau kasus uji di masing-masing jalurnya. *Test case* pada tiap-tiap jalur independen didefinisikan pada Tabel 6.4.

Tabel 0.4 Test case algoritme approve saji

No	Jalur	Data Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	1 – 2 – 3 – 5	Tekan tombol saji Peringatan = Ya	Pesanan berhasil disajikan. Sistem memperbaharui antrian pesanan	Sistem menampilkan antrian yang sudah diperbaharui	Valid
2	1 – 2 – 4 – 5	Tekan tombol saji Peringatan = Tidak	Pesanan gagal disajikan. Sistem tidak memperbaharui antrian pesanan	Sistem tidak memperbaharui antrian pesanan	Valid

6.1.1.3 Pengujian unit tambah transaksi

Pada fungsi tambah transaksi merupakan fungsi yang hanya dapat dilakukan oleh pengguna kasir dalam melakukan aktivitas transaksi dengan pembeli. Pada Tabel 6.5 menampilkan algoritme dari fungsi tambah transaksi.

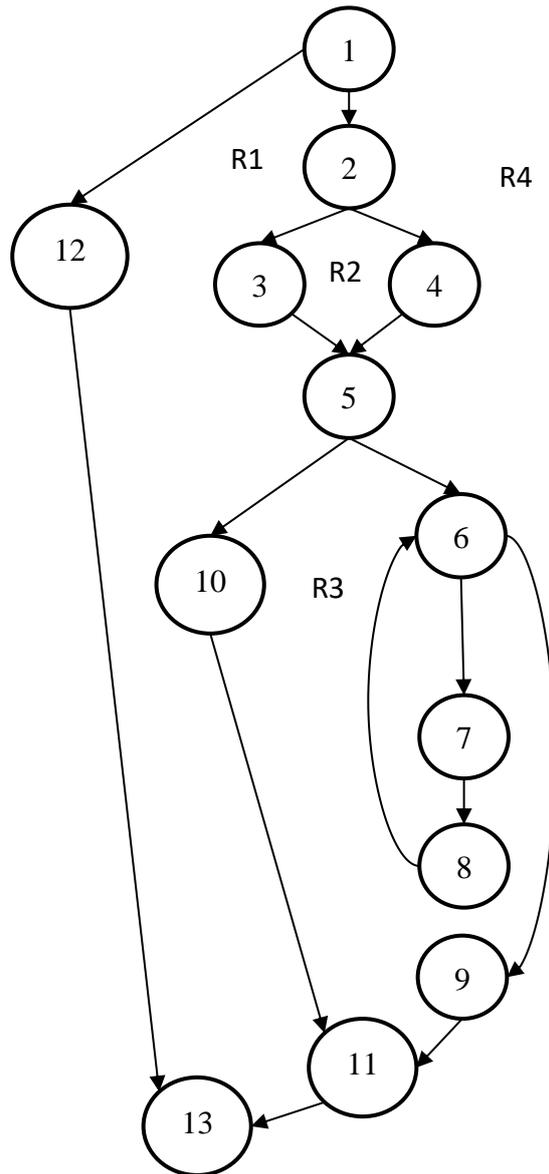
Kode: POS-KF-024

Tabel 0.5 Algoritme tambah transaksi

1	FUNCTION transaksi()	
2	IF 'STATUS' = "login" and 'LEVEL' = "kasir"	1
3	LOAD Model	
4	IF POST ('status'), 'ya' = 0	
5	no_meja = input('table')	2
6	IF POST ('note')	
7	Keterangan = input('note')	3
8	ELSE	4
9	keterangan = '-'	
10	POST <- grand_total (cost)	
11	POST <- id_user	
12	POST <- id_outlet	
13	IF LOAD Model <- INSERT(data)	5
14	id= Model<-INSERT(id)	
15	FOREACH item = POST(goods)	6
16	data[id_tm] = id	
17	data[id_produk] = item[i]	7
18		
19		
20		
21		
22		
23		

24	data[jumlah] = item[q]	
25	data[total] = item[q]*item[c]	
26	LOAD Model<-INSERT(data)	
27		
28	\$db<-set(tanggal, false)	
29	LOAD Model <- FIND(id)	
30	db->select('id_tm' AS 'queue')	8
31	db->where('id_tm' <=', \$id)	
32	db->where('tanggal')	
33		
34	queue= LOAD Model<-result_ARRAY[0]['queue']	
35		
36	LOAD Model <- UPDATE 'queue' USE 'id'	
37	db->WHERE id = 'id_tm'	
38		
39	LOAD library <- PUBLISH('queue')	
40	echo 'Transaksi Berhasil'	9
41		
42		
43	ELSE echo 'Transaksi Gagal'	10
44	END IF	11
45		
46		
47	ELSE Back to view login	12
48		
49	END IF	13

Dalam tabel yang ditunjukkan pada Tabel 6.5 merupakan pembahasan algoritme tambah transaksi. Pada algoritme yang tertampil terdapat angka-angka yang menunjukkan bahwa algoritme tersebut memiliki node-node sebanyak 7 node yang akan diubah ke bentuk flow graph. Dalam Gambar 6.4 menggambarkan *flow graph* tambah transaksi yang dibuat dari algoritme pada Tabel 6.4.



Gambar 0.4 Flow graph algoritme tambah transaksi

Dalam gambar yang diperlihatkan pada gambar 6.4 merupakan flow graph algoritme untuk menambah transaksi yang akan dilakukan perhitungan *cyclometric complexity* yang sudah dibahas pada landasan kepustakaan. Berikut perhitungan *cyclometric complexity* pada *flow graph* algoritme tambah transaksi.

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 15 - 13 + 2 \\
 &= 4
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= P + 1 \\
 &= 3 + 1 \\
 &= 4
 \end{aligned}$$

$$V(G) = \text{Jumlah region} \\ = 4$$

Dari perhitungan yang sudah dihitung didapatkan sebanyak 4 jalur independen, antara lain:

6.1.1.3.1.1.1 1 – 12 – 13

6.1.1.3.1.1.2 1 – 2 – 3 – 5 – 10 – 11 – 13

6.1.1.3.1.1.3 1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 11 – 13

6.1.1.3.1.1.4 1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 13

Tiga jalur independen yang sudah dibuat selanjutnya dilakukan pengujian melalui *test case* atau kasus uji di masing-masing jalurnya. *Test case* pada tiap-tiap jalur independen didefinisikan pada tabel 6.6.

Tabel 0.6 Test case algoritme tambah transaksi

No	Jalur	Data Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	1 – 12 – 13	Session = 7200 s	Tidak dapat mengakses ke halaman tambah transaksi dan kembali ke halaman login.	Tampil halaman login	Valid
2	1 – 2 – 3 – 5 – 10 – 11 – 13	Status = Ya No. meja = 1 Note = bayam Produk = Jumlah = Tekan tombol simpan	Transaksi gagal, sistem tidak memproses transaksi	Sistem tidak memproses masukan untuk melakukan transaksi	Valid
3	1 – 2 – 3 – 5 – 6 – 7 – 8 – 9 – 11 – 13	Status = Ya No. meja = 3 Note = bayam Produk = nasi pecel Jumlah = 1	Transaksi berhasil, sistem menampilkan struk dan antrian kepada Kasir	Sistem menampilkan struk dan antrian kepada Kasir	Valid

No	Jalur	Data Input	Hasil yang diharapkan	Hasil yang diperoleh	Status
		Tekan tombol simpan			
4	1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 13	Status = Ya No. meja = 7 Note = Produk = nasi pecel Jumlah = 1 Tekan tombol simpan	Transaksi berhasil, sistem menampilkan struk dan antrian kepada Kasir	Sistem menampilkan struk dan antrian kepada Kasir	Valid

6.1.2 Pengujian *Blackbox*

Pengujian *blackbox* juga dikenal sebagai pengujian berbasis spesifikasi. Pengujian *blackbox* mengacu pada aktivitas untuk memastikan bahwa unit program memenuhi kebutuhan yang disebutkan dalam spesifikasi dalam kata lain mengamati hasil eksekusi berdasarkan skenario *test case*. Pada penelitian ini menggunakan teknik requirement test untuk melakukan pengujian *blackbox*. Skenario pengujian yang dilakukan menggunakan *test case* normal dan *test case* alternatif.

Tabel 0.7 Test case requirement sistem POS

Test case requirement login			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna memasukkan username dan password pada form login yang telah disediakan dalam halaman login pengguna.	Pengguna masuk ke halaman home masing-masing.	Masukan pengguna berupa username dan password yang telah terdaftar di database, jika level pengguna adalah admin maka akan masuk ke halaman home admin, jika level pengguna adalah kasir maka akan masuk ke halaman home kasir dan jika level pengguna	Valid

		adalah dapur maka akan masuk ke halaman dapur.	
Kasus dan hasil uji (kondisi alternatif)			
Ada field atau semua field yang data masukannya salah/tidak terdaftar di database.	Tampil peringatan "Maaf, kombinasi username dengan password salah". Lalu kembali ke halaman login.	Muncul peringatan bahwa kombinasi username dan password salah dan kembali ke halaman login.	Valid
Ada field atau semua field tidak diisi.	Tampil peringatan bahwa field harus diisi.	Muncul peringatan pada <i>field</i> username dan password bahwa field harus diisi.	Valid
Test case requirement lihat produk			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih menu produk.	Sistem menampilkan halaman daftar produk.	Tampil halaman daftar produk.	Valid
Test case requirement tambah produk			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol tambah produk lalu mengisi semua data produk baru pada form tambah produk.	Produk yang baru dimasukkan tersimpan dan menampilkan halaman daftar produk dan menampilkan produk baru.	Tampil halaman daftar produk dan menampilkan produk baru.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin memilih	Sistem memberitahu	Terdapat pemberitahuan	Valid

tombol tambah produk lalu mengisi data produk baru pada form tambah produk dan ada <i>field</i> yang tidak diisi.	bahwa field harus diisi.	bahwa field harus diisi.	
Test case requirement edit produk			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih menu produk yang ingin diedit lalu mngubah data produk yang ingin diedit pada form edit produk.	Sistem menyimpan produk yang baru diubah dan menampilkan halaman daftar produk.	Produk yang baru diubah tersimpan dan sistem menampilkan halaman daftar produk.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin memilih menu produk lalu mengisi data produk baru pada form tambah produk dan ada <i>field</i> yang tidak diisi.	Sistem memberitahu bahwa field harus diisi.	Terdapat pemberitahuan bahwa field harus diisi.	Valid
Test case requirement hapus produk			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol hapus pada salah satu produk yang ingin dihapus.	Produk yang dipilih terhapus dari daftar produk.	Produk yang dipilih terhapus dari daftar produk dan database.	Valid
Test case requirement lihat bahan baku			
Kasus dan hasil uji (kondisi normal)			

Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih menu stok.	Sistem menampilkan halaman berisi daftar bahan baku.	Tampil halaman yang berisi daftar bahan baku.	Valid
Test case requirement tambah bahan baku			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol tambah bahan baku lalu mengisi semua data bahan baku baru pada form tambah bahan baku dan menekan tombol tambah.	Bahan baku yang baru dimasukkan tersimpan dan menampilkan halaman daftar bahan dan menampilkan bahan baru.	Tampil halaman daftar bahan baku yang menampilkan bahan baku yang baru ditambahkan dan bahan baku baru tersimpan dalam database.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin menekan tombol tambah bahan baku lalu mengisi data bahan baku baru pada form tambah bahan baku dan ada <i>field</i> yang tidak diisi.	Sistem memberitahu bahwa field harus diisi.	Terdapat pemberitahuan bahwa field harus diisi.	Valid
Test case requirement edit bahan			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol edit pada daftar bahan baku yang ingin diubah	Sistem menyimpan data bahan baku yang baru diubah dan menampilkan	Bahan baku yang baru diubah tersimpan dalam database dan sistem menampilkan	Valid

lalu mngubah data bahan baku pada form edit dan jika semua <i>field</i> sudah diisi, tekan tombol simpan.	halaman daftar bahan baku.	halaman daftar bahan baku.	
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin memilih tombol edit pada daftar bahan baku yang ingin diubah lalu mngubah data bahan baku pada form edit dan ada <i>field</i> yang tidak diisi. Lalu tekan tombol simpan.	Sistem memberitahu bahwa field harus diisi.	Terdapat pemberitahuan bahwa field harus diisi.	Valid
Test case requierement hapus bahan			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol hapus pada salah satu bahan baku yang ingin dihapus.	Bahan baku yang dipilih terhapus dari daftar bahan baku.	Bahan baku yang dipilih terhapus dari daftar bahan baku dan database.	Valid
Test case requirement lihat outlet			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih menu outlet.	Sistem menampilkan halaman berisi daftar outlet.	Tampil halaman yang berisi daftar bahan outlet.	Valid
Test case requierement tambah outlet			
Kasus dan hasil uji (kondisi normal)			

Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol tambah outlet lalu mengisi semua data outlet baru pada form tambah outlet dan menekan tombol tambah.	Outlet yang baru dimasukkan tersimpan dan menampilkan halaman daftar outlet.	Tampil halaman daftar outlet dan menampilkan outlet yang baru ditambahkan serta data outlet yang baru ditambahkan tersimpan di database.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin menekan tombol tambah outlet lalu mengisi data outlet baru pada form tambah outlet dan ada <i>field</i> yang tidak diisi.	Sistem memberitahu bahwa field harus diisi.	Terdapat pemberitahuan bahwa field harus diisi.	Valid
Test case requirement edit outlet			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol edit pada daftar outlet yang ingin diubah lalu mngubah data outlet pada form edit. Jika semua <i>field</i> sudah diisi, tekan tombol simpan.	Sistem menyimpan data outlet yang baru diubah dan menampilkan halaman daftar outlet.	Outlet yang baru diubah tersimpan dalam database dan sistem menampilkan halaman daftar outlet.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin memilih tombol edit pada daftar outlet yang ingin diubah lalu mngubah data	Sistem memberitahu bahwa field harus diisi.	Terdapat pemberitahuan bahwa field harus diisi.	Valid

outlet pada form edit dan ada <i>field</i> yang tidak diisi. Lalu tekan tombol simpan.			
Test case requirement hapus outlet			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol hapus pada salah satu outlet yang ingin dihapus.	Outlet yang dipilih terhapus dari daftar outlet.	Outlet yang dipilih terhapus dari daftar outlet dan database.	Valid
Test case requirement lihat pengguna			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih menu pengguna.	Sistem menampilkan halaman berisi daftar pengguna.	Tampil halaman yang berisi daftar pengguna.	Valid
Test case requirement tambah pengguna			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol tambah user lalu mengisi semua data user baru pada form tambah user dan menekan tombol tambah.	User yang baru dimasukkan tersimpan dan sistem menampilkan halaman daftar data user.	Tampil halaman daftar user dan menampilkan user yang baru ditambahkan serta data user yang baru ditambahkan tersimpan di database.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin menekan tombol tambah user	Sistem memberitahu	Terdapat pemberitahuan	Valid

lalu mengisi data user baru pada form tambah user dan ada <i>field</i> yang tidak diisi.	bahwa field harus diisi.	bahwa field harus diisi.	
Test case requirement edit pengguna			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol edit pada daftar user yang ingin diubah lalu mengubah data user pada form edit. Jika semua <i>field</i> sudah diisi, tekan tombol simpan.	Sistem menyimpan data user yang baru diubah dan menampilkan halaman daftar user.	User yang baru diubah tersimpan dalam database dan sistem menampilkan halaman daftar user.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin memilih tombol edit pada daftar user yang ingin diubah lalu mengubah data user pada form edit dan ada <i>field</i> yang tidak diisi. Lalu tekan tombol simpan.	Sistem memberitahu bahwa field harus diisi.	Terdapat pemberitahuan bahwa field harus diisi.	Valid
Test case requirement hapus pengguna			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol hapus pada salah satu user yang ingin dihapus.	User yang dipilih terhapus dari daftar outlet.	User yang dipilih terhapus dari daftar user dan database.	Valid
Test case requirement lihat pengeluaran			
Kasus dan hasil uji (kondisi normal)			

Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih menu pengeluaran.	Sistem menampilkan halaman berisi daftar pengeluaran.	Tampil halaman yang berisi daftar pengeluaran	Valid
Test case requirement tambah pengeluaran			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol tambah pengeluaran lalu mengisi semua data pengeluaran baru pada form tambah pengeluaran dan menekan tombol tambah.	Pengeluaran yang baru dimasukkan tersimpan dan sistem menampilkan halaman daftar pengeluaran	Tampil halaman daftar pengeluaran dan menampilkan pengeluaran yang baru ditambahkan serta pengeluaran yang baru ditambahkan tersimpan di database.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna level admin menekan tombol tambah pengeluaran lalu mengisi data pengeluaran baru pada form tambah pengeluaran dan ada <i>field</i> yang tidak diisi.	Sistem memberitahu bahwa field harus diisi.	Terdapat pemberitahuan bahwa field harus diisi.	Valid
Test case requirement hapus pengeluaran			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna level admin memilih tombol hapus pada salah satu	Pengeluaran yang dipilih terhapus dari	Pengeluaran yang dipilih terhapus dari daftar pengeluaran dan database.	Valid

pengeluaran yang ingin dihapus.	daftar pengeluaran.		
Test case requirement tambah transaksi			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna admin memilih kategori produk, lalu memilih produk yang dibeli dan mengisi semua data yang ada pada form transaksi dan menekan tombol simpan.	Terdapat pemberitahuan nomor antrian dan uang kembalian.	Sistem menampilkan pemberitahuan berisi nomor antrian dan uang kembalian	Valid
Test case requirement cetak struk			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna admin menekan tombol cetak pada struk.	Sistem mencetak struk dan kembali ke halaman tambah transaksi.	Sistem mencetak struk dan kembali ke halaman tambah transaksi.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna dapur menekan tombol close pada struk.	Sistem kembali ke halaman tambah transaksi baru tanpa harus mencetak struk.	Sistem menampilkan halaman tambah transaksi baru tanpa harus mencetak struk.	Valid
Test case requirement lihat urutan pesanan			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna dapur melakukan login dengan username dan password yang benar.	Masuk ke halaman awal dapur yaitu tampil urutan pesanan.	Sistem menampilkan halaman awal berisi urutan pesanan.	Valid

Test case requirement approve saji			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna dapur menekan tombol (-) pada item yang sudah dibuat. Jika semua item yang dipesan sudah dibuat maka tombol saji akan berubah warna menjadi biru lalu pengguna dapur menekan tombol saji.	Tampil peringatan berupa apakah makanan ingin disajikan. Jika ya, maka pesanan tersebut akan hilang dari antrian. Jika tidak, maka pesanan tetap pada antrian.	Sistem menampilkan peringatan berupa apakah makanan ingin disajikan. Jika ya, maka pesanan tersebut akan hilang dari antrian. Jika tidak, maka pesanan tetap pada antrian.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna dapur menekan tombol saji pada pesanan.	Tampil peringatan berupa apakah makanan ingin disajikan. Jika ya, maka pesanan tersebut akan hilang dari antrian dan jumlah kumulatif item yang harus dimasak akan berkurang secara otomatis. Jika tidak, maka pesanan tetap pada antrian.	Sistem menampilkan peringatan berupa apakah makanan ingin disajikan. Jika ya, maka pesanan tersebut akan hilang dari antrian dan jumlah kumulatif item yang harus dimasak akan berkurang secara otomatis.. Jika tidak, maka pesanan tetap pada antrian.	Valid
Test case requirement approve selesai			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna dapur menekan tombol selesai pada	Tampil peringatan berupa apakah	Sistem menampilkan peringatan berupa apakah anda yakin	Valid

pesanan yang sudah diantar ke pembeli.	anda yakin pesanan ini telah selesai. Jika ya, maka pesanan tersebut akan hilang dari saji. Jika tidak, maka pesanan tetap pada antrian saji.	pesanan ini telah selesai. Jika ya, maka pesanan tersebut akan hilang dari saji. Jika tidak, maka pesanan tetap pada antrian saji.	
Test case requirement tambah stok masuk			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna dapur menentukan bahan yang ingin ditambah stok masuknya lalu menekan <i>button link</i> stok masuk dan mengisi semua masukan, lalu menekan tombol tambah.	Masukan tersimpan dan tampil hamanan stok.	Sistem menampilkan halaman stok dan masukan pengguna tersimpan dalam database.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna dapur menentukan bahan yang ingin ditambah stok masuknya lalu menekan <i>button link</i> stok masuk dan tidak mengisi semua masukan atau hanya mengisi salah satu masukan saja, lalu menekan tombol tambah.	Tampil peringatan bahwa <i>field</i> harus diisi.	Sistem menampilkan peringatan bahwa <i>field</i> tidak boleh kosong.	Valid
Test case requirement tambah stok keluar			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan

Pengguna dapur menentukan bahan yang ingin ditambah stok masuknya lalu menekan <i>button link</i> stok keluar dan mengisi semua masukan, lalu menekan tombol tambah.	Masukan tersimpan dan tampil hamanan stok.	Sistem menampilkan halaman stok dan masukan pengguna tersimpan dalam database.	Valid
Kasus dan hasil uji (kondisi alternatif)			
Pengguna dapur menentukan bahan yang ingin ditambah stok masuknya lalu menekan <i>button link</i> stok keluar dan tidak mengisi semua masukan atau hanya mengisi salah satu masukan saja, lalu menekan tombol tambah.	Tampil peringatan bahwa <i>field</i> harus diisi.	Sistem menampilkan peringatan bahwa <i>field</i> tidak boleh kosong.	Valid
Test case requierement lihat penjualan harian			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna admin memasukkan tanggal range tanggal dan memilih outlet, lalu menekan tombol lihat.	Tampil tabel penjualan harian sesuai dengan tanggal dan outlet yang dimasukkan.	Sistem menampilkan tabel penjualan harian sesuai dengan tanggal dan outlet yang dimasukkan.	Valid
Test case requierement lihat laporan omset			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna admin memilih bulan, lalu	Tampil laporan omset sesuai	Sistem menampilkan laporan omset sesuai	Valid

menekan tombol lihat.	dengan bulan yang dipilih.	dengan bulan yang dipilih.	
Test case requirement lihat laporan data transaksi			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna admin memilih tanggal dan outlet, lalu menekan tombol lihat.	Tampil laporan data transaksi pada tanggal dan outlet sesuai dengan yang dipilih.	Sistem menampilkan laporan data transaksi pada sesuai dengan tanggal dan outlet yang dipilih.	Valid
Test case requirement lihat laporan produk			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna admin memilih tanggal dan outlet, lalu menekan tombol lihat.	Tampil laporan produk pada tanggal dan outlet sesuai dengan yang dipilih.	Sistem menampilkan laporan produk dengan tanggal dan outlet yang dipilih.	Valid
Test case requirement lihat laporan stok			
Kasus dan hasil uji (kondisi normal)			
Prosedur pengujian	Hasil yang diharapkan	Hasil pengamatan	Kesimpulan
Pengguna admin memilih tanggal dan outlet, lalu menekan tombol lihat.	Tampil laporan stok pada tanggal dan outlet sesuai dengan yang dipilih.	Sistem menampilkan laporan stok dengan tanggal dan outlet yang dipilih.	Valid

6.2 Pengujian Non-Fungsional

Dalam penelitian ini juga melakukan pengujian non-fungsional yaitu menguji performa dari sistem menggunakan komponen dari browser Google Chrome yang bernama Performance-Analyser. Pada pengujian ini mengambil 3 sampel halaman yang akan diakses lalu dilihat *loading time* nya. Halaman yang digunakan sebagai sample adalah halaman data user, halaman tambah transaksi dan halaman antrian

pesanan. Berikut akan dibahas hasil dari pengujian performa yang sudah dilakukan.

6.2.1 Pengujian performa akses halaman data user

Pada gambar 6.5 menampilkan bahwa dalam melakukan akses halaman data user maka terdapat 34 *requests*. Adapun total *loading time* nya adalah 1530 ms atau kurang dari 3 detik. Lalu ada DOM Content Loading dan DOM Processing masing-masing waktunya adalah 197 ms dan 293 ms yang sama-sama masih dibawah 1 detik.

Requests	Domains	Subdomains of TLD	Requests to Host	TLD& Subdomain Requests
34	2	0	32	32
Total	Time to First Byte	DOM Content Loading	DOM Processing	Slowest Call
1530ms	1202ms	197ms	293ms	57ms
Average Call				
9ms				

Gambar 0.5 Pengujian performa halaman data user

6.2.2 Pengujian performa tambah transaksi

Pada gambar 6.6 menampilkan bahwa dalam melakukan akses untuk tambah transaksi maka terdapat 40 *requests*. Adapun total *loading time* nya adalah 1067 ms atau kurang dari 3 detik. Lalu ada DOM Content Loading dan DOM Processing masing-masing waktunya adalah 638 ms dan 656 ms yang sama-sama masih dibawah 1 detik.

Requests	Domains	Subdomains of TLD	Requests to Host	TLD& Subdomain Requests
40	2	0	38	38
Total	Time to First Byte	DOM Content Loading	DOM Processing	Slowest Call
1067ms	361ms	638ms	656ms	125ms
Average Call				
28ms				

Gambar 0.6 Pengujian performa tambah transaksi

6.2.3 Pengujian performa akses halaman antrian pesanan

Pada gambar 6.7 menampilkan bahwa dalam melakukan akses halaman antrian pesanan maka terdapat 21 *requests*. Adapun total *loading time* nya adalah 681 ms atau kurang dari 3 detik. Lalu ada DOM Content Loading dan DOM Processing masing-masing waktunya adalah 300 ms dan 406 ms yang sama-sama masih dibawah 1 detik.

Requests	Domains	Subdomains of TLD	Requests to Host	TLD& Subdomain Requests
21	2	0	20	20
Total	Time to First Byte	DOM Content Loading	DOM Processing	Slowest Call
681ms	234ms	300ms	406ms	1785ms
Average Call				
93ms				

Gambar 0.7 Pengujian performa halaman antrian pesanan

6.3 Analisis Pengujian

Pada pengujian yang sudah dilakukan maka langkah selanjutnya adalah melakukan analisis pengujian. Adapun hasil pengujian yang akan dianalisis yaitu pengujian *whitebox*, pengujian *blackbox* dan pengujian non-fungsional. Pada pengujian *whitebox* yang dilakukan pada penelitian ini menguji 3 unit yaitu *login*, *approve* saji dan tambah transaksi. Dari algoritme yang dibuat pada 3 unit tersebut, masing-masing unit menghasilkan total jalur independen yang berbeda. Pada *login* memiliki jalur independen dan tambah transaksi sebanyak 4 sedangkan pada fungsi *approve* saji memiliki jalur independen sebanyak 2. Semakin banyak jalur independen yang ditemukan maka semakin tinggi pula tingkat kompleksitasnya. Dalam penelitian ini, jalur independen yang ditemukan paling tinggi adalah pada *login* dan tambah transaksi yaitu lebih dari 3, itu menandakan bahwa tingkat kompleksitas pada unit tersebut tidak terlalu kompleks.

Pada penelitian ini juga dilakukan pengujian *blackbox*. Dari 33 fungsi yang diuji, kesemuanya memiliki hasil yang memuaskan karena semua fungsi tersebut valid atau dapat berjalan sesuai dengan fungsionalnya baik dari kondisi normal maupun alternatif. Hal tersebut membuktikan bahwa sistem POS sudah siap untuk digunakan adapun kekurangannya bisa menjadi saran untuk penelitian selanjutnya. Sedangkan pada pengujian non-fungsional yang dilakukan pada penelitian ini yaitu menguji performance pada sistem POS yang dibangun. Dengan menggunakan komponen dari browser Google Chrome yang bernama Performance-Analyser didapatkan. Dari 3 akses halaman yang diuji, menunjukkan hasil yang baik sebab tidak ada yang lebih dari 3 detik namun rata-rata total load yang dihasilkan adalah 1 detik. Dimana jika *loading time* yang dihasilkan kurang dari 3 detik maka dapat dikatakan bahwa tingkat *performance* pada sistem POS ini adalah cepat.

6.4 Analisis Hasil Penelitian

Berdasarkan hasil analisis pengujian yang sudah dilakukan, tidak ditemukan kesalahan yang berarti yang berdampak pada jalannya sistem. Pada pembangunan sistem POS terintegrasi dalam lingkup rumah makan beserta cabangnya sudah melalui berbagai macam proses. Proses yang paling *urgent* selama dalam

pembangunan sistem ini adalah penentuan identifikasi dan spesifikasi kebutuhan sebab pada saat sistem sudah diimplementasi dan ditunjukkan pada skateholder, selalu ada yang ditambahkan lagi, tidak hanya dari sisi penambahan fitur namun juga pada desain database. Proses yang paling sulit dalam pembangunan sistem ini adalah pada proses implementasi. Dibutuhkan waktu yang lama hingga sistem ini benar-benar selesai dibangun. Namun setelah akhirnya sistem sudah jadi tanpa ada perubahan, skateholder merasa puas dengan hasilnya karena melebihi ekspektasi awal sewaktu yang pernah diungkapkan pada hasil wawancara dalam penggalan identifikasi masalah.