

BAB 6 PENGUJIAN

Bab ini membahas tentang pengujian yang dilakukan pada Sistem Informasi *Showroom* Mobil dengan menggunakan jenis pengujian *Black Box* dan *White Box*. Yang mencakup perencanaan pengujian, hasil pengujian *black box* dan *white box*, serta analisis hasil pengujian.

6.1 Rencana Pengujian

Pengujian dilakukan dengan menggunakan metode *blackbox* dan dilakukan sesuai dengan *use case scenario* yang sudah didefinisikan pada penelitian perancangan SISMOB (Syafruddin, 2017). Tabel 6.1 menunjukkan rencana pengujian SISMOB dengan metode *blackbox*. Pengujian *blackbox* terfokus hanya pada apa yang akan dihasilkan sistem dari inputan yang diberikan tidak menguji alur logika sistem.

Tabel 6.1 Rencana pengujian *blackbox* SISMOB

No	Kode Fitur	Nama Fitur	Use Case	Jenis Pengujian
1	SMOB1	<i>Login</i>	<i>Login</i>	<i>Blackbox, Whitebox</i>
2	SMOB2	Mengelola data penjualan	Mengelola data penjualan	<i>Blackbox, Whitebox</i>
3	SMOB3	Mengelola data pembelian	Mengelola data pembelian	<i>Blackbox, Whitebox</i>
4	SMOB4	Mengelola data mobil	Mengelola data mobil	<i>Blackbox</i>
5	SMOB5	Mencetak bukti transaksi	Mengelola data penjualan	<i>Blackbox</i>
6	SMOB6	Pencarian data	Mengelola data penjualan, mengelola data pembelian	<i>Blackbox</i>
7	SMOB7	Melihat laporan keuangan	Melihat laporan keuangan	<i>Blackbox</i>
8	SMOB7	Melihat katalog	Melihat katalog	<i>Blackbox</i>
9	SMOB8	Melihat detail mobil	Melihat detail mobil	<i>Blackbox</i>

6.2 Pengujian *Black Box*

Pengujian *black box* dilakukan untuk melihat apakah program yang sudah diimplementasikan sudah sesuai dengan alur jalannya aplikasi yang sudah dirancang sebelumnya. Pengujian *black box* dilakukan dengan menggunakan *test case*. *Test case* terdiri dari masukan, kondisi yang akan dieksekusi dan juga hasil yang diharapkan. *Test case* dibuat berdasarkan spesifikasi *use case* yang telah dirancang pada penelitian lain atau perancangan SISMOB (Syafruddin, 2017).

6.2.1 Melihat Katalog

Pada tabel 6.2 merupakan *test case* dari melihat katalog dan terdapat dua skenario dan juga dua *test case*.

Tabel 6.2 Tabel *test case* melihat katalog

Melihat katalog		
<i>Test case</i> 1	<i>Use case</i> melihat katalog skenario 1	Pembeli berhasil melihat katalog mobil yang tersedia
	Masukan	-
	Keluaran yang diharapkan	Pembeli berhasil melihat katalog mobil yang tersedia
	Keluaran	Pembeli berhasil melihat katalog mobil yang tersedia
<i>Test case</i> 2	<i>Use case</i> melihat katalog skenario 2	Pembeli memfilter katalog sesuai dengan yang diinginkan
	Masukan	Pembeli memilih tombol harga tertinggi
	Keluaran yang diharapkan	Pembeli berhasil melihat katalog mobil yang telah diurutkan berdasarkan harga tertinggi
	Keluaran	Pembeli berhasil melihat katalog mobil yang telah diurutkan berdasarkan harga tertinggi

Dari dua *test case* yang diuji berdasarkan skenario *use case* melihat katalog pada tabel 6.2, didapatkan keluaran yang sama dengan keluaran yang diharapkan dan sudah didefinisikan pada spesifikasi *use case* melihat katalog.

6.2.2 Melihat Detail Mobil

Pada tabel 6.3 merupakan *test case* dari melihat detail mobil dan terdapat dua *test case*.

Tabel 6.3 Tabel *test case* melihat detail mobil

Melihat detail mobil		
<i>Test case</i> 3	<i>Use case</i> melihat detail mobil skenario 1	Pembeli berhasil melihat detail mobil yang dipilih
	Masukan	Pembeli memilih mobil yang ingin dilihat secara detail
	Keluaran yang diharapkan	Pembeli berhasil melihat detail mobil yang dipilih
	Keluaran	Pembeli berhasil melihat detail mobil yang dipilih

Tabel 6.3 Tabel *test case* melihat detail mobil (lanjutan)

Melihat detail mobil		
<i>Test case</i> 4	<i>Use case</i> melihat detail mobil skenario 2	Pembeli tidak jadi melihat detail mobil
	Masukan	-
	Keluaran yang diharapkan	Pembeli tidak mendapatkan detail mobil
	Keluaran	Pembeli tidak mendapatkan detail mobil

Dari dua *test case* yang diuji berdasarkan skenario *use case* melihat detail mobil pada tabel 6.3, didapatkan keluaran yang sama dengan keluaran yang diharapkan dan sudah didefinisikan pada spesifikasi *use case* melihat detail mobil.

6.2.3 Login

Pada tabel 6.4 merupakan *test case* dari *login* dan terdapat empat *test case*.

Tabel 6.4 Tabel *test case* login

Login		
<i>Test case</i> 5	<i>Use case login</i> skenario 1	Berhasil <i>login</i> ke dalam sistem
	Masukan	Username : admin Password : admin
	Keluaran yang dihasilkan	Sistem berhasil mengautentifikasi pengguna dan memberikan akses ke pengguna untuk menggunakan fitur di dalam sistem
	Keluaran	Sistem berhasil mengautentifikasi pengguna dan menampilkan halaman <i>home</i> sesuai dengan pengguna yang terotorisasi
<i>Test case</i> 6	<i>Use case login</i> skenario 2	Terdapat <i>field</i> yang belum diisi
	Masukan	Username : admin Password : NULL
	Keluaran yang diharapkan	Sistem akan kembali ke halaman <i>login</i> dan menampilkan peringatan " <i>The password field is required</i> "
	Keluaran	Sistem kembali ke halaman <i>login</i> dan menampilkan peringatan " <i>The password field is required</i> "
<i>Test case</i> 7	<i>Use case login</i> skenario 3	Pengguna tidak teridentifikasi

Tabel 6.4 Tabel *test case login* (lanjutan)

Melihat detail mobil		
	Masukan	Username : admin1 Password : admin
	Keluaran yang diharapkan	Sistem akan kembali ke halaman <i>login</i> dan menampilkan peringatan “user tidak ada”
	Keluaran	Sistem kembali ke halaman <i>login</i> dan menampilkan peringatan “user tidak ada”
<i>Test case</i> 8	<i>Use case login</i> skenario 4	Tidak jadi melakukan <i>login</i>
	Masukan	-
	Keluaran yang diharapkan	Tidak jadi melakukan login
	Keluaran	Tidak jadi melakukan login dan kembali ke halaman yang dituju

Dari empat *test case* yang diuji berdasarkan skenario *use case login* pada tabel 6.4, didapatkan keluaran yang sama dengan keluaran yang diharapkan dan sudah didefinisikan pada spesifikasi *use case login*.

6.2.4 Mengelola Data Penjualan

Pada tabel 6.5 merupakan *test case* dari mengelola data penjualan dan terdapat empat *test case*.

Tabel 6.5 Tabel *test case* mengelola data penjualan

Mengelola data penjualan		
<i>Test case</i> 9	<i>Use case</i> mengelola data penjualan skenario 1	Berhasil mengelola data penjualan (<i>Input</i> data penjualan)
	Masukan	nama: Samsuri alamat: jl. Abcd status: lunas No Telp: 08080900988 harga: 195000000 keterangan: surat lengkap dokumen: bukti.jpg
	Keluaran yang dihasilkan	Sistem akan menampilkan halaman list mobil yang dijual serta pesan bahwa data berhasil disimpan
	Keluaran	Sistem berhasil menampilkan halaman list mobil yang dijual serta pesan bahwa data berhasil disimpan

Tabel 6.5 Tabel test case mengelola data penjualan (lanjutan)

Mengelola data penjualan		
Test case 10	Use case mengelola data penjualan skenario 2	Terdapat <i>field</i> yang belum diisi
	Masukan	nama: Samsuri alamat: <i>NULL</i> status: lunas No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg
	Keluaran yang diharapkan	Sistem akan kembali ke halaman <i>form</i> penjualan dan menampilkan peringatan bahwa masih ada <i>field</i> yang kosong
	Keluaran	Sistem kembali ke halaman <i>form</i> penjualan dan menampilkan peringatan masih ada <i>field</i> yang kosong
Test case 11	Use case mengelola data penjualan skenario 3	Mencetak surat perjanjian
	Masukan	Admin memilih tombol surat perjanjian
	Keluaran yang diharapkan	Sistem akan menampilkan halaman surat perjanjian yang siap untuk dicetak
	Keluaran	Sistem menampilkan halaman surat perjanjian yang siap untuk dicetak
Test case 12	Use case mengelola data penjualan skenario 4	Tidak jadi melakukan pengelolaan data penjualan
	Masukan	-
	Keluaran yang diharapkan	Tidak jadi melakukan pengelolaan data penjualan dan sistem akan kembali ke halaman yang dituju
	Keluaran	Tidak jadi melakukan pengelolaan data penjualan dan sistem mengembalikan ke halaman yang dituju

Dari empat *test case* yang diuji berdasarkan skenario *use case* mengelola data penjualan pada tabel 6.5, didapatkan keluaran yang sama dengan keluaran yang diharapkan dan sudah didefinisikan pada spesifikasi *use case* mengelola data penjualan.

6.2.5 Mengelola Data Pembelian

Pada tabel 6.6 merupakan *test case* dari mengelola data pembelian dan terdapat tiga *test case*.

Tabel 6.6 Tabel *test case* mengelola data pembelian

Mengelola data pembelian		
<i>Test case</i> 13	<i>Use case</i> mengelola data pembelian skenario 1	Berhasil mengelola data pembelian (<i>input</i> data pembelian)
	Masukan	gambar: mobil.jpg merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Keterangan: <i>NULL</i>
	Keluaran yang diharapkan	Sistem akan menampilkan pesan bahwa data berhasil disimpan
	Keluaran	Sistem menampilkan pesan bahwa data berhasil disimpan
<i>Test case</i> 14	<i>Use case</i> mengelola data pembelian skenario 2	Terdapat <i>field</i> yang belum diisi
	Masukan	gambar: mobil.jpg merk: Toyota type: Agya tahun: <i>NULL</i> nama pemilik: samsuri No Telp: <i>NULL</i> Harga: <i>NULL</i> Keterangan: <i>NULL</i>
	Keluaran yang diharapkan	Sistem akan kembali ke halaman <i>form</i> pembelian dan menampilkan peringatan bahwa masih ada <i>field</i> yang kosong
	Keluaran	Sistem kembali ke halaman <i>form</i> pembelian dan menampilkan peringatan bahwa masih ada <i>field</i> yang kosong

Tabel 6.6 Tabel test case mengelola data pembelian (lanjutan)

Test case 15	Use case mengelola data pembelian skenario 3	Tidak jadi melakukan pengelolaan data pembelian
	Masukan	-
	Keluaran yang diharapkan	Tidak jadi melakukan pengelolaan data pembelian dan sistem akan kembali ke halaman yang dituju
	Keluaran	Tidak jadi melakukan pengelolaan data pembelian dan sistem mengembalikan ke halaman yang dituju

Dari tiga *test case* yang diuji berdasarkan skenario *use case* mengelola data pembelian pada tabel 6.6, didapatkan keluaran yang sama dengan keluaran yang diharapkan dan sudah didefinisikan pada spesifikasi *use case* mengelola data pembelian.

6.2.6 Melihat Laporan Keuangan

Pada tabel 6.7 merupakan *test case* dari melihat laporan keuangan dan terdapat tiga *test case*.

Tabel 6.7 Tabel test case melihat laporan keuangan

Melihat laporan keuangan		
Test case 16	Use case melihat laporan keuangan skenario 1	Berhasil melihat laporan keuangan
	Masukan	-
	Keluaran yang diharapkan	Administrasi berhasil melihat laporan keuangan
	Keluaran	Administrasi berhasil melihat laporan keuangan
Test case 17	Use case melihat laporan keuangan skenario 2	Administrasi memfilter laporan keuangan berdasarkan bulan tertentu
	Masukan	Bulan Agustus
	Keluaran yang diharapkan	Sistem akan menampilkan laporan keuangan berdasarkan input bulan dari administasi
	Keluaran	Sistem menampilkan laporan keuangan berdasarkan input bulan yang dipilih

Tabel 6.7 Tabel test case melihat laporan keuangan (lanjutan)

Test case 18	Use case melihat laporan keuangan skenario 3	Tidak jadi melihat laporan keuangan
	Masukan	-
	Keluaran yang diharapkan	Tidak jadi melihat laporan keuangan dan sistem akan kembali ke halaman yang dituju
	Keluaran	Tidak jadi melihat laporan keuangan dan sistem mengembalikan ke halaman yang dituju

Dari tiga test case yang diuji berdasarkan skenario use case melihat laporan keuangan pada tabel 6.7, didapatkan keluaran yang sama dengan keluaran yang diharapkan dan sudah didefinisikan pada spesifikasi use case melihat laporan keuangan.

6.2.7 Mengelola Data Mobil

Pada tabel 6.8 merupakan test case dari mengelola data mobil dan terdapat tiga test case.

Tabel 6.8 Tabel test case mengelola data mobil

Mengelola data mobil		
Test case 19	Use case mengelola data mobil skenario 1	Berhasil mengelola data mobil (mengubah data mobil)
	Masukan	gambar: mobil.jpg merk: Toyota type: Agya tahun: 2014 Harga: 12345678 Keterangan: Baru
	Keluaran yang diharapkan	Sistem akan menampilkan pesan bahwa data berhasil diubah
	Keluaran	Sistem menampilkan pesan bahwa data berhasil diubah
Test case 20	Use case mengelola data mobil skenario 2	Terdapat field yang belum diisi

Tabel 6.8 Tabel *test case* mengelola data mobil (lanjutan)

	Masukan	gambar: NULL merk: Toyota type: Agya tahun: 2014 Harga: 12345678 Keterangan: <i>NULL</i>
	Keluaran yang diharapkan	Sistem akan menampilkan laporan keuangan berdasarkan input bulan dari administasi
	Keluaran	Sistem menampilkan laporan keuangan berdasarkan input bulan yang dipilih
<i>Test case</i> 21	<i>Use case</i> mengelola data mobil skenario 3	Tidak jadi melakukan pengelolaan data mobil
	Masukan	-
	Keluaran yang diharapkan	Tidak jadi melakukan pengelolaan data mobil dan sistem akan kembali ke halaman yang dituju
	Keluaran	Tidak jadi melakukan pengelolaan data mobil dan sistem mengembalikan ke halaman yang dituju

Dari tiga *test case* yang diuji berdasarkan skenario *use case* mengelola data mobil pada tabel 6.8, didapatkan keluaran yang sama dengan keluaran yang diharapkan dan sudah didefinisikan pada spesifikasi *use case* mengelola data mobil.

6.3 Pengujian *White Box*

Pengujian *white box* dilakukan untuk mendeteksi kesalahan yang ada pada kode program dan juga untuk mengetahui kompleksitas yang terdapat pada kode program yang telah diimplementasikan. Langkah-langkah dalam melakukan pengujian *white box* adalah sebagai berikut:

1. Membuat *flow graph*
2. Menghitung *cyclomatic complexity*
3. Menentukan *independent path*

Berikut merupakan pengujian *white box* dari fitur SISMOB:

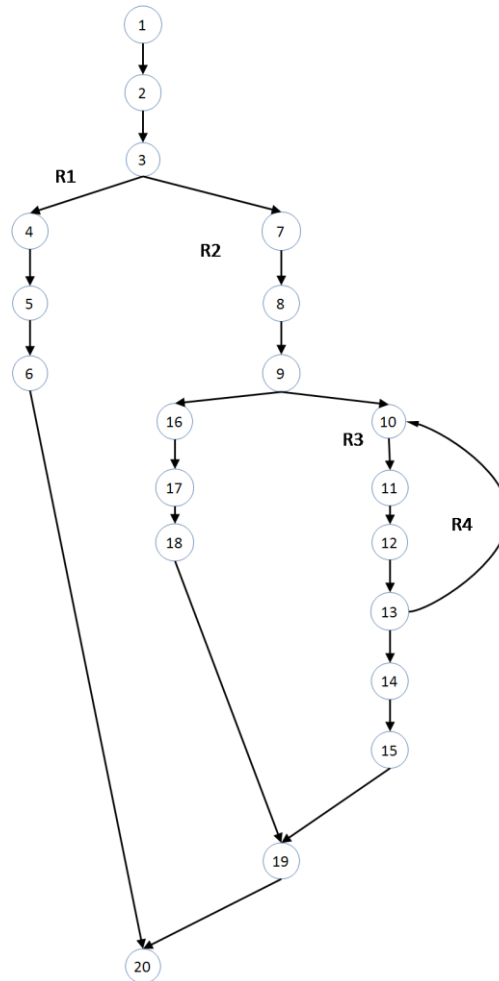
6.3.1 Login

Pada tabel merupakan kode program *login* dari SISMOB

Tabel 6.9 Tabel kode program *login*

1	public function login(){	1	
2	\$u = \$this->input->post('username');	}	
3	\$p = \$this->input->post('password');		
4		}	
5	\$this->load->model('mAuthentication');		
6	\$this->load->library('form_validation');		
7	\$this->form_validation->set_rules('username', 'Username',		
8	'trim required');	}	
9	\$this->form_validation->set_rules('password', 'Password',		
10	'required trim');		
11		}	
12	if (\$this->form_validation->run() == FALSE) {		3
13	\$this->session->set_flashdata('flash_message',		}
14	warn_msg(validation_errors()));		
15	redirect('cAuthentication');		4
16	}		5
17	else{		6
18	\$hasil = \$this->mAuthentication->cek_user(\$u,\$p);		7
19			8
20	if (\$hasil->num_rows() > 0) {		9
21	foreach (\$hasil->result() as \$row) {		10
22	\$session = array('username' => \$row-		}
23	>username,		
24			
25	'nama' => \$row->nama,	11	
26		}	
27	'role' => \$row->role);		
28	\$this->session->set_userdata(\$session);	12	
29	}	13	
30	redirect('cDashboard');	14	
31	}	15	
32	else{	16	
33	\$this->session->set_flashdata('flash_message',	}	
34	warn_msg('User tidak ada'));		
35	redirect('cAuthentication');	17	
36	}	18	
37	}	19	
38	}	20	

Gambar 6.1 merupakan *flow graph* dari *login*



Gambar 6.1 Flow graph login

Berikut merupakan perhitungan *cyclomatic complexity* dari *login*:

$$V(G) = 4$$

$$\begin{aligned} V(G) &= E - N + 2 \\ &= 22 - 20 + 2 \\ &= 4 \end{aligned}$$

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

Berikut merupakan *independent path* dari *login*:

- 1) 1-2-3-4-5-6-20
- 2) 1-2-3-7-8-9-16-17-18-19-20
- 3) 1-2-3-7-8-9-10-11-12-13-14-15-19-20
- 4) 1-2-3-7-8-9-10-11-12-13-10-11-12-13-14-15-19-20

Berdasarkan *independent path*, maka diperoleh *test case* sesuai dengan tabel .

Tabel 6.10 Test case login

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Submit: Login username: admin password: <i>NULL</i>	Kembali ke halaman <i>login</i> dan menampilkan pesan peringatan " <i>The password field is required</i> "	Kembali ke halaman login dan tampil pesan " <i>The password field is required</i> "	Valid
2	Submit: Login username: admin password: 123456	Kembali ke halaman <i>login</i> dan menampilkan pesan warning " <i>user tidak ada</i> "	Kembali ke halaman login dan tampil pesan " <i>user tidak ada</i> "	Valid
3	Submit: Login username: admin password: admin	Menampilkan halaman dashboard	Tampil halaman dashboard	Valid
4	Submit: Login username: admin password: admin	Menampilkan halaman dashboard	Tampil halaman dashboard	Valid

6.3.2 Mengelola data penjualan

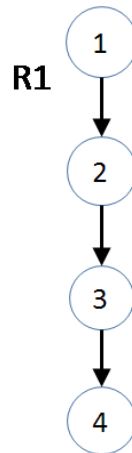
a. Lihat data penjualan

Pada tabel merupakan kode program lihat data penjualan

Tabel 6.11 Tabel kode program lihat data penjualan

1	<code>function get_data_penjualan(){</code>	1
2	<code> \$this->load->model('mPenjualan');</code>	2
3	<code> </code>	
4	<code> \$data = array();</code>	
5	<code> </code>	
6	<code> \$data['title'] = 'SISMOB Histori Penjualans';</code>	3
7	<code> \$data['user'] = \$this->session->userdata('nama');</code>	
8	<code> \$data['penjualan'] = \$this->mPenjualan->get_penjualan();</code>	
9	<code> </code>	
10	<code> \$this->load->view('vData_Penjualan', \$data);</code>	4
11	<code>}</code>	

Gambar 6.2 merupakan *flow graph* dari lihat data penjualan



Gambar 6.2 Flow graph lihat data penjualan

Berikut merupakan perhitungan *cyclomatic complexity* dari lihat data penjualan:

$$\begin{aligned}
 V(G) &= 1 \\
 V(G) &= E - N + 2 \\
 &= 3 - 4 + 2 \\
 &= 1 \\
 V(G) &= P + 1 \\
 &= 0 + 1 = 1
 \end{aligned}$$

Berikut merupakan *independent path* dari lihat data penjualan:

- 1) 1-2-3-4

Berdasarkan *independent path*, maka diperoleh *test case* sesuai dengan tabel

Tabel 6.12 Tabel test case lihat data penjualan

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Mengakses lihat histori penjualan	Sistem menampilkan halaman data penjualan	Tampil halaman data penjualan	Valid

b. Input data penjualan

Pada tabel merupakan kode program input data penjualan

Tabel 6.13 Tabel kode program input data penjualan

1	function input_data_penjualan() {	1
2	\$path = './assets/uploads/penjualan/';	
3	\$waktu_transaksi = date("y-m-d H:i:s");	
4	\$this->load->model('mPenjualan');	
5	\$this->load->model('mMobil');	
6	\$this->load->library('form_validation');	
7	\$this->form_validation->set_rules('nama', 'Nama Pembeli',	
8	'required');	
9	\$this->form_validation->set_rules('alamat', 'Alamat Pembeli',	2
10	'required');	
11	\$this->form_validation->set_rules('noHP', 'Nomor Telepon',	
12	'required');	
13	\$this->form_validation->set_rules('status', 'Status Pembeli',	
14	'required');	
15	\$this->form_validation->set_rules('hargabel', 'Harga',	
16	'required');	
17	\$id = \$this->input->post('id_mobil');	
18		
19	if (\$this->form_validation->run() == FALSE) {	3
20	\$this->session->set_flashdata('flash_message',	4
21	warn_msg(validation_errors()));	
22	redirect('cPenjualan/get_input_penjualan/'.\$id);	5
23	}	6
24	else{	7
25	\$dir = 'Penjualan'.date("dmyHis");	
26	\$path = \$path . \$dir;	8
27	\$buatDir = mkdir(\$path,0755,TRUE);	
28	if (\$buatDir) {	9
29	\$hitung = count(\$_FILES['gambar']['name']);	10
30	for (\$i=0; \$i < \$hitung; \$i++) {	11
31	\$_FILES['gambar']['name'] =	
32	\$_FILES['gambar']['name'][\$i];	
33	\$_FILES['gambar']['type'] =	
34	\$_FILES['gambar']['type'][\$i];	
35	\$_FILES['gambar']['tmp_name'] =	
36	\$_FILES['gambar']['tmp_name'][\$i];	
37	\$_FILES['gambar']['error'] =	12
38	\$_FILES['gambar']['error'][\$i];	
39	\$_FILES['gambar']['size'] =	
40	\$_FILES['gambar']['size'][\$i];	
41		
42	\$config['upload_path'] = \$path;	
43	\$config['allowed_types'] = 'jpg png';	

Tabel 6.13 Tabel kode program input data penjualan (lanjutan)

44	<code>\$config['file_name'] =</code>	
45	<code>'Dokumen_' . date("dmyHis");</code>	
46		
47	<code>\$this->load->library('upload', \$config);</code>	13
48	<code>if (\$this->upload->do_upload('gambar')) {</code>	14
49	<code> \$upload_data = \$this->upload-</code>	
50	<code>>data();</code>	15
51		
52	<code> \$gambar[] =</code>	
53	<code>\$upload_data['file_name'];</code>	16
54	<code> }</code>	17
55	<code> else{</code>	18
56	<code> \$this->session-</code>	
57	<code>>set_flashdata('flash_message', warn_msg(\$this->upload-</code>	19
58	<code>>display_errors()));</code>	
59	<code> redirect('cPenjualan/get_input_penjualan/' . \$id);</code>	20
60	<code> }</code>	21
61	<code> }</code>	22
62	<code> \$gambar = implode(", ", \$gambar);</code>	23
63		
64	<code> \$data = array('id_status' => \$this-</code>	
65	<code>>input->post('status'),</code>	
66	<code> 'id_pabrikan' => \$this-</code>	
67	<code>>input->post('pabrikan'),</code>	
68	<code> 'merk' => \$this-</code>	
69	<code>>input->post('merk'),</code>	
70	<code> 'tahun' => \$this-</code>	
71	<code>>input->post('tahun'),</code>	
72	<code> 'nama_pembeli' => \$this-</code>	
73	<code>>input->post('nama'),</code>	
74	<code> 'alamat' => \$this-</code>	
75	<code>>input->post('alamat'),</code>	24
76	<code> 'no_telp' => \$this-</code>	
77	<code>>input->post('noHP'),</code>	
78	<code> 'harga_jual' =></code>	
79	<code>\$this->input->post('harga'),</code>	
80	<code> 'harga_beli' =></code>	
81	<code>\$this->input->post('hargabel'),</code>	
82	<code> 'path' => \$path .</code>	
83	<code>"/",</code>	
84	<code> 'file_gambar' => \$gambar,</code>	
85	<code> 'keterangan' => \$this-</code>	
86	<code>>input->post('keterangan'),</code>	

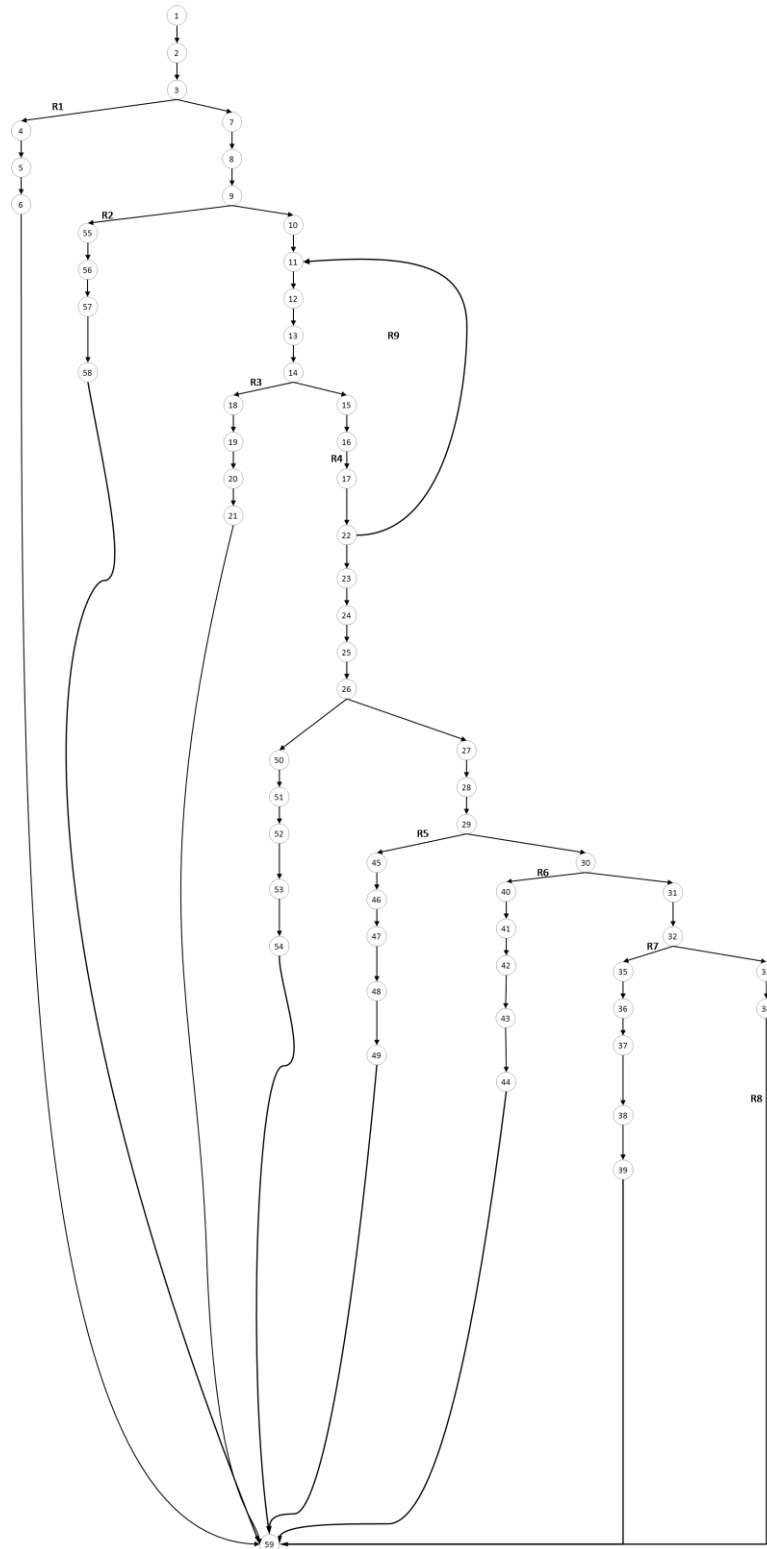
Tabel 6.13 Tabel kode program input data penjualan (lanjutan)

87	'waktu_penjualan' =>	
88	\$waktu_transaksi);	
89	\$input = \$this->mPenjualan-	25
90	>insert_penjualan(\$data);	
91		
92	if (\$input != 0) {	26
93	\$this->load->helper('file');	27
94	\$hapusFile = delete_files(\$this->input-	28
95	>post('path') . '/', TRUE);	
96	if (\$hapusFile) {	29
97	if (rmdir(\$this->input-	30
98	>post('path')) {	
99	\$hapusMobil = \$this-	31
100	>mMobil->hapus_mobil(\$this->input->post('id_mobil'));	
101		
102	if (\$hapusMobil != 0) {	32
103	echo "	
104	<script language='JavaScript'>	
105	alert('Sukses Menyimpan Data');	33
106	document.location='/SISMOB/cPenjualan/penjualan'	
107	</script>	
108	";	
109	}	34
110	else{	35
111	\$this->session-	
112	>set_flashdata('flash_message', warn_msg('Gagal Menyimpan Data, Mobil	36
113	Belum Dihapus'));	
114	redirect('cPenjualan/get_input_penjualan/' . \$id);	37
115	}	38
116	}	39
117	else{	40
118	\$this->session-	
119	>set_flashdata('flash_message', warn_msg('Gagal Menyimpan Data,	41
120	Direktori Mobil Belum Dihapus'));	
121	redirect('cPenjualan/get_input_penjualan/' . \$id);	42
122	}	43
123	}	44
124	else{	45
125	\$this->session-	
126	>set_flashdata('flash_message', warn_msg('Gagal Menyimpan Data, Gambar	46
127	Mobil Belum Dihapus'));	
128	redirect('cPenjualan/get_input_penjualan/' . \$id);	47
129	}	48

Tabel 6.13 Tabel kode program input data penjualan (lanjutan)

130	}	49
131	else{	50
132	\$this->session-	51
133	>set_flashdata('flash_message', warn_msg('Gagal Menyimpan Data'));	52
134	redirect('cPenjualan/get_input_penjualan/' . \$id);	53
135	}	54
136	}	55
137	else{	56
138	\$this->session->set_flashdata('flash_message',	57
139	warn_msg('Gagal Membuat Direktori'));	58
140	redirect('cPenjualan/get_input_penjualan/' . \$id);	59
141	}	
142	}	
143	}	

Gambar 6.3 merupakan *flow graph* dari input data penjualan



Gambar 6.3 Flow graph input data penjualan

Berikut merupakan perhitungan *cyclomatic complexity* dari input data pembelian:

$$\begin{aligned}
 V(G) &= 9 \\
 V(G) &= E - N + 2 \\
 &= 66 - 59 + 2 \\
 &= 9 \\
 V(G) &= P + 1 \\
 &= 8 + 1 = 9
 \end{aligned}$$

Berikut merupakan *independent path* dari input data pembelian:

- 1) 1-2-3-4-5-6-59
- 2) 1-2-3-7-8-9-55-56-57-58-59
- 3) 1-2-3-7-8-9-10-11-12-13-14-18-19-20-21-59
- 4) 1-2-3-7-8-9-10-11-12-13-14-15-16-17-21-22-23-24-25-26-50-51-52-53-54-59
- 5) 1-2-3-7-8-9-10-11-12-13-14-15-16-17-21-22-23-24-25-26-27-28-29-45-46-47-48-49-59
- 6) 1-2-3-7-8-9-10-11-12-13-14-15-16-17-21-22-23-24-25-26-27-28-29-30-40-41-42-43-44-59
- 7) 1-2-3-7-8-9-10-11-12-13-14-15-16-17-21-22-23-24-25-26-27-28-29-30-31-32-35-36-37-38-39-59
- 8) 1-2-3-7-8-9-10-11-12-13-14-15-16-17-21-22-23-24-25-26-27-28-29-30-31-32-33-34-59
- 9) 1-2-3-7-8-9-10-11-12-13-14-15-16-17-21-22-11-12-13-14-15-16-17-21-22-23-24-25-26-27-28-29-30-31-32-33-34-59

Berdasarkan *independent path*, maka diperoleh, *test case* sesuai dengan tabel

Tabel 6.14 Test case input data penjualan

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Submit: Simpan nama: Samsuri alamat: <i>NULL</i> status: lunas No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman form input data penjualan dan menampilkan pesan warning bahwa masih ada field kosong	Kembali ke halaman form input data penjualan dan tampil pesan bahwa masih ada <i>field</i> yang kosong	Valid

Tabel 6.14 Test case input data penjualan (lanjutan)

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
2	Submit: Simpan nama: Samsuri alamat: Jl. Tirto status: lunas No Telp: 080989999 harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman form input data penjualan	Kembali ke halaman form input data penjualan	Valid
3	Submit: Simpan nama: Samsuri alamat: <i>NULL</i> status: lunas No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.gif	Kembali ke halaman form input data penjualan dan menampilkan pesan warning bahwa file yang di unggah tidak diperbolehkan	Kembali ke halaman form input data penjualan dan muncul pesan bahwa file yang di unggah tidak diperbolehkan	Valid
4	Submit: Simpan nama: Samsuri alamat: Jl. Tirto status: lunas No Telp: 080989999 harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman form input data penjualan	Kembali ke halaman form input data penjualan	Valid
5	Submit: Simpan nama: Samsuri alamat: Jl. Tirto status: lunas No Telp: 080989999 harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman form input data penjualan	Kembali ke halaman form input data penjualan	Valid
6	Submit: Simpan nama: Samsuri alamat: Jl. Tirto status: lunas No Telp: 080989999 harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman form input data penjualan	Kembali ke halaman form input data penjualan	Valid

Tabel 6.14 Test case input data penjualan (lanjutan)

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
7	Submit: Simpan nama: Samsuri alamat: Jl. Tirto status: lunas No Telp: 080989999 harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman form input data penjualan	Kembali ke halaman form input data penjualan	Valid
8	Submit: Simpan nama: Samsuri alamat:Jl.abc status: lunas No Telp: 080989999 harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Menampilkan halaman list mobil yang dijual dan pesan bahwa data berhasil disimpan	Tampil halaman list mobil yang dijual dan muncul pesan bahwa data berhasil disimpan	Valid
9	Submit: Simpan nama: Samsuri alamat: Samsuri status: hutang No Telp: 080989999 harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg, perjanjian.jpg	Menampilkan halaman list mobil yang dijual dan pesan bahwa data berhasil disimpan	Tampil halaman list mobil yang dijual dan muncul pesan bahwa data berhasil disimpan	Valid

c. Memperbarui data penjualan

Pada tabel merupakan kode program memperbarui data penjualan

Tabel 6.15 Tabel kode program memperbarui data penjualan

1	<code>function update_penjualan(\$id='') {</code>	1
2	<code> \$path = \$this->input->post('path');</code>	
3	<code> \$pathbaru = \$this->input->post('path');</code>	
4	<code> \$waktu_transaksi = date("y-m-d H:i:s");</code>	
5	<code> \$this->load->model('mPenjualan');</code>	
6	<code> \$this->load->library('form_validation');</code>	
7	<code> \$this->form_validation->set_rules('nama', 'Nama Pembeli',</code>	2
8	<code>'required');</code>	
9	<code> \$this->form_validation->set_rules('alamat', 'Alamat Pembeli',</code>	
10	<code>'required');</code>	
11	<code> \$this->form_validation->set_rules('noHP', 'Nomor Telepon',</code>	
12	<code>'required');</code>	

Tabel 6.15 Tabel kode program memperbarui data penjualan (lanjutan)

13	<code>\$this->form_validation->set_rules('status', 'Status Pembeli',</code>	
14	<code>'required');</code>	
15	<code>\$this->form_validation->set_rules('harga', 'Harga', 'required');</code>	
16		
17	<code>if (\$this->form_validation->run() == FALSE) {</code>	3
18	<code> \$this->session->set_flashdata('flash_message',</code>	
19	<code>warn_msg(validation_errors()));</code>	4
20	<code> redirect('cPenjualan/penjualan');</code>	
21	<code>}</code>	5
22	<code>else{</code>	6
23	<code> \$hapusFile = delete_files(\$path, TRUE);</code>	7
24	<code> if (\$hapusFile) {</code>	8
25	<code> \$hitung = count(\$_FILES['gambar']['name']);</code>	9
26	<code> for (\$i=0; \$i < \$hitung; \$i++) {</code>	10
27	<code> \$_FILES['gambar']['name'] =</code>	
28	<code>\$_FILES['gambar']['name'][\$i];</code>	
29	<code> \$_FILES['gambar']['type'] =</code>	
30	<code>\$_FILES['gambar']['type'][\$i];</code>	
31	<code> \$_FILES['gambar']['tmp_name'] =</code>	
32	<code>\$_FILES['gambar']['tmp_name'][\$i];</code>	
33	<code> \$_FILES['gambar']['error'] =</code>	
34	<code>\$_FILES['gambar']['error'][\$i];</code>	11
35	<code> \$_FILES['gambar']['size'] =</code>	
36	<code>\$_FILES['gambar']['size'][\$i];</code>	
37		
38	<code> \$config['upload_path'] = \$path;</code>	
39	<code> \$config['allowed_types'] = 'jpg png';</code>	
40	<code> \$config['file_name'] =</code>	
41	<code>'Dokumen_' . date("dmyHis");</code>	
42		
43	<code> \$this->load->library('upload',</code>	12
44	<code>\$config);</code>	
45	<code> if (\$this->upload->do_upload('gambar')) {</code>	13
46	<code> \$upload_data = \$this->upload->data();</code>	14
47	<code> \$gambar[] =</code>	15
48	<code>\$upload_data['file_name'];</code>	
49	<code> }</code>	16
50	<code> else{</code>	17
51	<code> \$this->session-></code>	
52	<code>set_flashdata('flash_message', warn_msg(\$this->upload-></code>	18
53	<code>display_errors()));</code>	
54	<code> redirect('cPenjualan/get_update_penjualan/' . \$id);</code>	
55	<code> }</code>	19

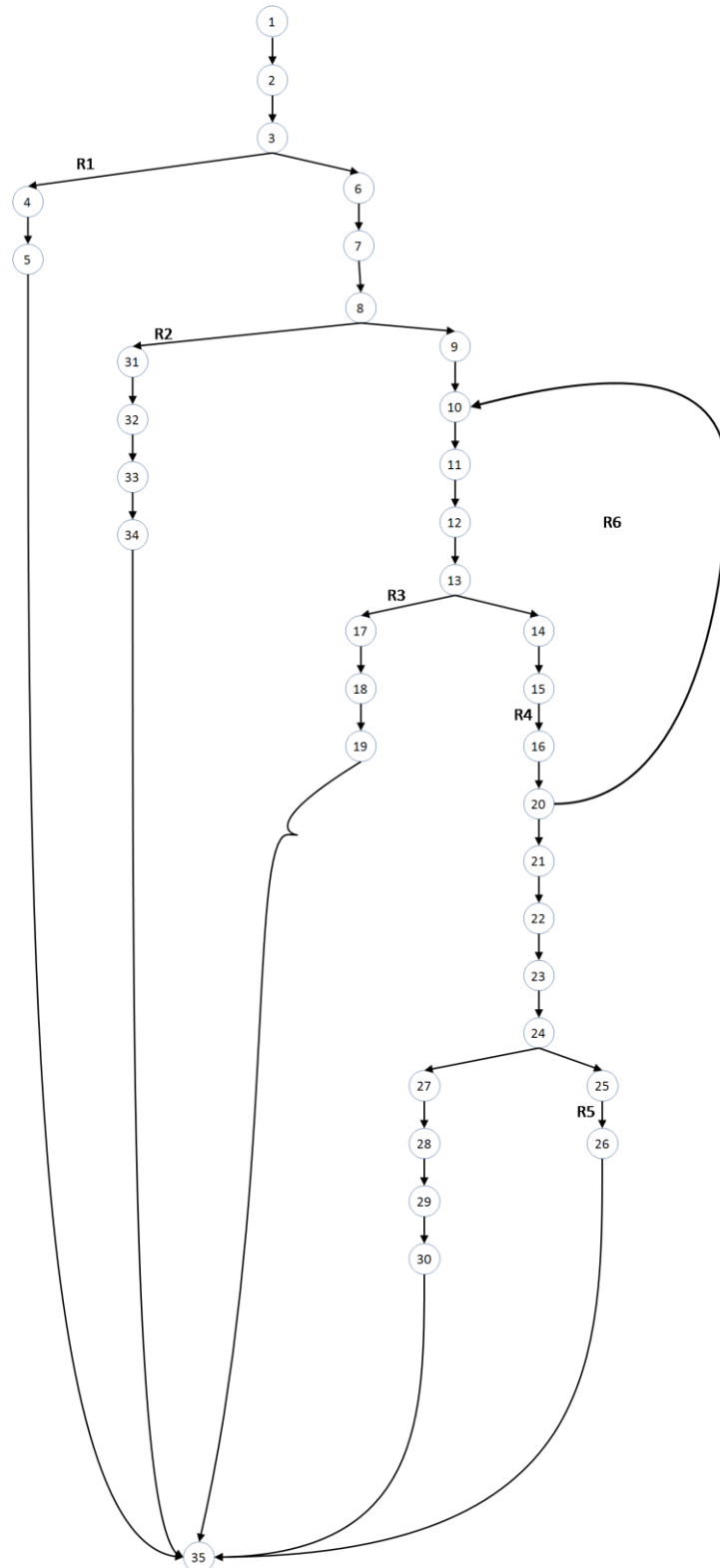
Tabel 6.15 Tabel kode program memperbarui data penjualan (lanjutan)

56	}	20
57	\$gambar = implode(" ", \$gambar);	21
58	\$data = array('id_status' =>	
59	\$this->input->post('status'),	
60	'nama_pembeli' =>	
61	\$this->input->post('nama'),	
62	'alamat' =>	
63	\$this->input->post('alamat'),	
64	'no_telp' =>	
65	\$this->input->post('noHP'),	
66	'harga' =>	22
67	\$this->input->post('harga'),	
68	'path' =>	
69	\$path,	
70	'file_gambar' =>	
71	\$gambar,	
72	'keterangan' =>	
73	\$this->input->post('keterangan'),	
74	'waktu_penjualan' =>	
75	\$waktu_transaksi);	
76	\$input = \$this->mPenjualan-	23
77	>update_penjualan(\$id,\$data);	
78		
79	if (\$input != 0) {	24
80	echo "	
81	<script language='JavaScript'>	
82	alert('Sukses Merubah Data');	25
83	document.location='/SISMOB/cPenjualan/get_data_penjualan'	
84	</script>	
85	";	
86	}	26
87	else{	27
88	\$this->session-	
89	>set_flashdata('flash_message', warn_msg('Gagal Merubah Data'));	28
90	redirect('cPenjualan/get_update_penjualan/'.\$id);	
91	}	29
92	}	30
93	else{	31
94	\$this->session->set_flashdata('flash_message',	32
95	warn_msg('Gagal Merubah Data'));	

Tabel 6.15 Tabel kode program memperbarui data penjualan (lanjutan)

96	<code>redirect('cPenjualan/get_update_penjualan/'.\$id);</code>	33
97	<code> }</code>	34
98	<code> }</code>	35
99	<code>}</code>	

Gambar 6.4 merupakan *flow graph* dari memperbarui data penjualan



Gambar 6.4 *Flow graph* memperbarui data penjualan

Berikut merupakan perhitungan *cyclomatic complexity* dari memperbarui data penjualan:

$$\begin{aligned}
 V(G) &= 6 \\
 V(G) &= E - N + 2 \\
 &= 39 - 35 + 2 \\
 &= 6 \\
 V(G) &= P + 1 \\
 &= 5 + 1 = 6
 \end{aligned}$$

Berikut merupakan *independent path* dari memperbaiki data penjualan:

- 1) 1-2-3-4-5-35
- 2) 1-2-3-6-7-8-31-32-33-34-35
- 3) 1-2-3-6-7-8-9-10-11-12-13-17-18-19-34-35
- 4) 1-2-3-6-7-8-9-10-11-12-13-14-15-16-19-20-21-22-23-24-27-28-29-30-34-35
- 5) 1-2-3-6-7-8-9-10-11-12-13-14-15-16-19-20-21-22-23-24-25-26-35
- 6) 1-2-3-6-7-8-9-10-11-12-13-14-15-16-19-20-10-11-12-13-14-15-16-19-20-21-22-23-24-25-26-35

Berdasarkan *independent path*, maka diperoleh *test case* sesuai dengan tabel

Tabel 6.16 Test case memperbaiki data penjualan

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Submit: Simpan nama: Samsuri alamat: <i>NULL</i> status: lunas No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman form perubahan data penjualan dan menampilkan pesan warning bahwa masih ada field kosong	Kembali ke halaman form perubahan data penjualan dan tampil pesan bahwa masih ada <i>field</i> yang kosong	Valid
2	Submit: Simpan nama: Samsuri alamat: <i>NULL</i> status: lunas No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman form perubahan data penjualan	Kembali ke halaman form perubahan data penjualan	Valid

Tabel 6.16 Test case memperbarui data penjualan (lanjutan)

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
3	Submit: Simpan nama: Samsuri alamat: <i>NULL</i> status: lunas No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.gif	Kembali ke halaman perubahan data penjualan dan menampilkan pesan warning bahwa file yang di unggah tidak diperbolehkan	Kembali ke halaman form perubahan data penjualan dan tampil pesan bahwa file yang di unggah tidak diperbolehkan	Valid
4	Submit: Simpan nama: Samsuri alamat: <i>NULL</i> status: lunas No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Kembali ke halaman perubahan data penjualan	Kembali ke halaman perubahan data penjualan	Valid
5	Submit: Simpan nama: Samsuri alamat: <i>NULL</i> status: lunas No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg	Menampilkan halaman list data penjualan	Tampil halaman list data penjualan	Valid
6	Submit: Simpan nama: Samsuri alamat: <i>NULL</i> status: hutang No Telp: <i>NULL</i> harga: 195000000 keterangan: <i>NULL</i> dokumen: bukti.jpg, perjanjian.jpg	Menampilkan halaman list data penjualan	Tampil halaman list data penjualan	Valid

6.3.3 Mengelola data pembelian

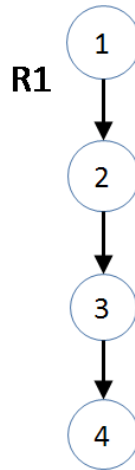
- a. Lihat data pembelian

Pada tabel merupakan kode program lihat data pembelian

Tabel 6.17 Tabel kode program lihat data pembelian

1	function get_data_pembelian(){	1
2	\$this->load->model('mPembelian');	2
3		
4	\$data = array();	
5		
6	\$data['title'] = 'SISMOB Histori Pembelian';	3
7	\$data['user'] = \$this->session->userdata('nama');	
8	\$data['pembelian'] = \$this->mPembelian->get_pembelian();	
9		
10	\$this->load->view('vData_Pembelian', \$data);	4
11	}	

Gambar 6.5 merupakan *flow graph* dari lihat data pembelian



Gambar 6.5 Flow graph lihat data pembelian

Berikut merupakan perhitungan *cyclomatic complexity* dari lihat data pembelian:

$$\begin{aligned}
 V(G) &= 1 \\
 V(G) &= E - N + 2 \\
 &= 3 - 4 + 2 \\
 &= 1 \\
 V(G) &= P + 1 \\
 &= 0 + 1 = 1
 \end{aligned}$$

Berikut merupakan *independent path* dari lihat data pembelian:

- 1) 1-2-3-4

Berdasarkan *independent path*, maka diperoleh *test case* sesuai dengan tabel

Tabel 6.18 Test case lihat data pembelian

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Mengakses lihat histori pembelian	Sistem menampilkan halaman data pembelian	Tampil halaman data pembelian	Valid

b. Input data pembelian

Pada tabel merupakan kode program input data pembelian:

Tabel 6.19 Tabel kode program input data pembelian

1	function input_data_pembelian(){function input_data_pembelian(){	1
2	\$path = './assets/uploads/mobil/';	
3	\$waktu_transaksi = date("y-m-d H:i:s");	
4		
5	\$this->load->model('mMobil');	
6	\$this->load->model('mPembelian');	
7	\$this->load->library('form_validation');	
8		
9	\$this->form_validation->set_rules('merk', 'Merk Mobil',	
10	'required');	
11	\$this->form_validation->set_rules('type', 'Tipe Mobil',	2
12	'required');	
13	\$this->form_validation->set_rules('tahun', 'Tahun Mobil',	
14	'required');	
15	\$this->form_validation->set_rules('harga', 'Harga Mobil',	
16	'required');	
17	\$this->form_validation->set_rules('nama', 'Nama Pemilik',	
18	'required');	
19	\$this->form_validation->set_rules('noHP', 'Nomor Telepon',	
20	'required');	
21		
22	if (\$this->form_validation->run() == FALSE) {	3
23	\$this->session->set_flashdata('flash_message',	4
24	warn_msg(validation_errors()));	
25	redirect('cPembelian/pembelian');	5
26	}	6
27	else{	7
28	\$dir = 'Mobil'.date("dmyHis");	8
29	\$path = \$path . \$dir;	
30	\$buatDir = mkdir(\$path,0755,TRUE);	

Tabel 6.19 Tabel kode program input data pembelian (lanjutan)

31	if (\$buatDir) {	9
32	\$hitung = count(\$_FILES['gambars']['name']);	10
33	for (\$i=0; \$i < \$hitung; \$i++) {	11
34	\$_FILES['gambar']['name'] =	
35	\$_FILES['gambars']['name'][\$i];	
36	\$_FILES['gambar']['type'] =	
37	\$_FILES['gambars']['type'][\$i];	
38	\$_FILES['gambar']['tmp_name'] =	
39	\$_FILES['gambars']['tmp_name'][\$i];	
40	\$_FILES['gambar']['error'] =	
41	\$_FILES['gambars']['error'][\$i];	12
42	\$_FILES['gambar']['size'] =	
43	\$_FILES['gambars']['size'][\$i];	
44		
45	\$config['upload_path'] = \$path;	
46	\$config['allowed_types'] = 'jpg png';	
47	\$config['file_name'] =	
48	'Gambar_'.date("dmyHis");	
49		
50	\$this->load->library('upload', \$config);	13
51	if (\$this->upload->do_upload('gambar')) {	14
52	\$upload_data = \$this->upload->	
53	>data();	15
54	\$gambars[] =	
55	\$upload_data['file_name'];	
56	}	16
57	else{	17
58	\$this->session->	
59	>set_flashdata('flash_message', warn_msg(\$this->upload->	18
60	>display_errors()));	
61	redirect('cPembelian/pembelian');	
62	}	19
63	}	20
64	\$gambar = implode(", ", \$gambars);	21
65	\$data = array('id_pabrikan' => \$this->input->	
66	>post('merk'),	
67	'merk' =>	
68	\$this->input->post('type'),	22
69	'harga' =>	
70	\$this->input->post('harga'),	
71	'tahun' =>	
72	\$this->input->post('tahun'),	
73	'path' =>	

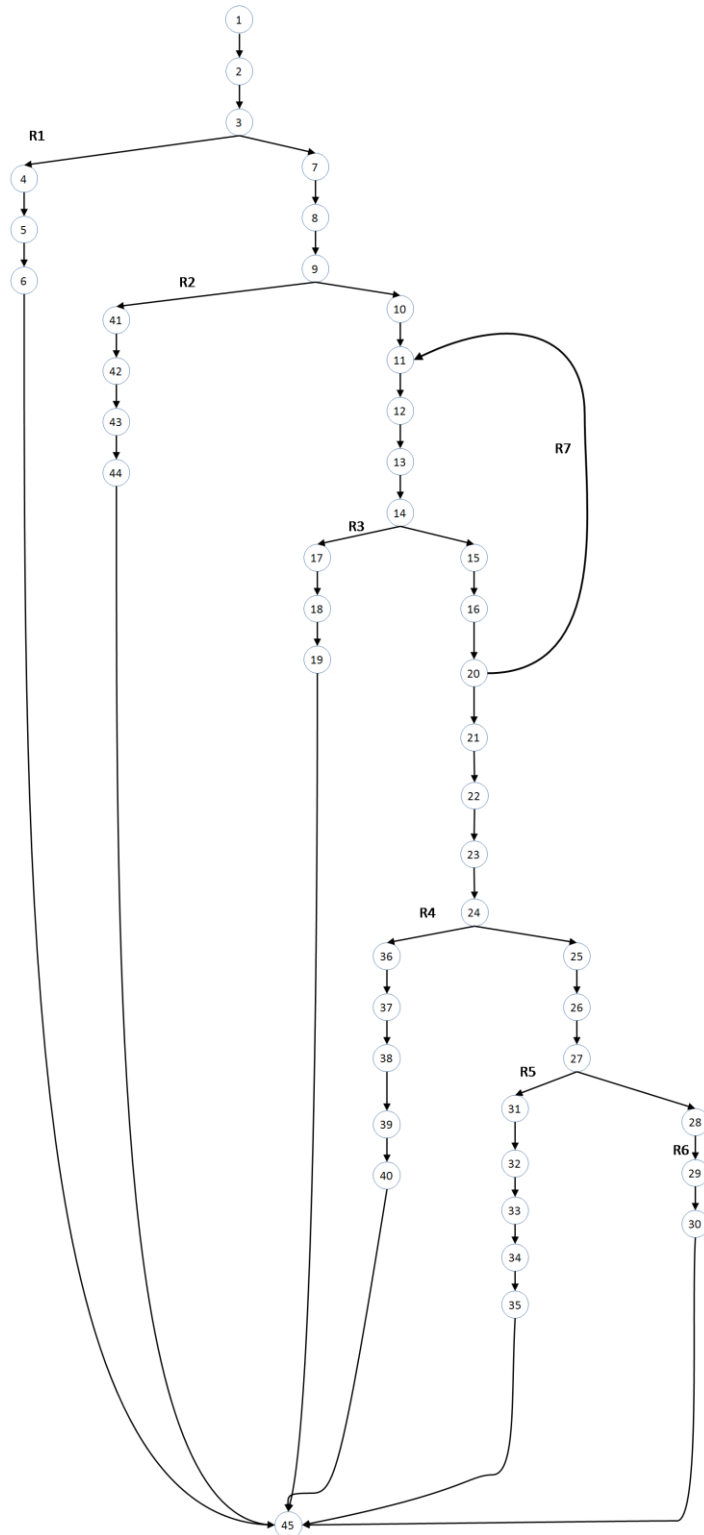
Tabel 6.19 Tabel kode program input data pembelian (lanjutan)

74	<code>\$path . "/"</code> ,	
75		
76	<code>'file_gambar' => \$gambar,</code>	
77	<code>'keterangan' =></code>	
78	<code>\$this->input->post('t1');</code>	
79	<code>\$input = \$this->mMobil->insert_mobil(\$data);</code>	23
80		
81	<code>if (\$input != 0) {</code>	24
82	<code>\$data1 = array('id_pabrikan' => \$this-</code>	
83	<code>>input->post('merk'),</code>	
84	<code>'merk' => \$this-</code>	
85	<code>>input->post('type'),</code>	
86	<code>'tahun' => \$this-</code>	
87	<code>>input->post('tahun'),</code>	
88	<code>'nama_penjual' => \$this-</code>	
89	<code>>input->post('nama'),</code>	25
90	<code>'no_telp' => \$this-</code>	
91	<code>>input->post('noHP'),</code>	
92	<code>'harga_beli' => \$this-</code>	
93	<code>>input->post('harga'),</code>	
94	<code>'kelengkapan' => \$this-</code>	
95	<code>>input->post('t2'),</code>	
96	<code>'waktu_pembelian'=></code>	
97	<code>\$waktu_transaksi);</code>	
98	<code>\$simpan = \$this->mPembelian-</code>	26
99	<code>>insert_pembelian(\$data1);</code>	
100		
101	<code>if (\$simpan != 0) {</code>	27
102	<code>\$this->session-</code>	28
103	<code>>set_flashdata('flash_message', succ_msg('Sukses Menyimpan Data'));</code>	
104	<code>redirect('cPembelian/pembelian');</code>	29
105	<code>}</code>	30
106	<code>else{</code>	31
107	<code>\$this->session-</code>	32
108	<code>>set_flashdata('flash_message', warn_msg('Gagal Menyimpan Data'));</code>	
109	<code>redirect('cPembelian/pembelian');</code>	33
110	<code>}</code>	34
111	<code>}</code>	35
112	<code>else{</code>	36
113	<code>\$this->session-</code>	37
114	<code>>set_flashdata('flash_message', warn_msg('Gagal Menyimpan Data'));</code>	
115	<code>redirect('cPembelian/pembelian');</code>	38
116	<code>}</code>	39

Tabel 6.19 Tabel kode program input data pembelian (lanjutan)

117	}	40
118	else{	41
119	\$this->session->set_flashdata('flash_message',	42
120	warn_msg('Gagal Membuat Direktori'));	
121	redirect('cPembelian/pembelian');	43
122	}	44
123	}	45
124	}	

Gambar 6.6 merupakan *flow graph* input data pembelian



Gambar 6.6 Flow graph input data pembelian

Berikut merupakan perhitungan *cyclomatic complexity* dari input data pembelian:

$$\begin{aligned}
 V(G) &= 7 \\
 V(G) &= E - N + 2 \\
 &= 50 - 45 + 2 \\
 &= 7 \\
 V(G) &= P + 1 \\
 &= 6 + 1 = 7
 \end{aligned}$$

Berikut merupakan *independent path* dari input data pembelian:

- 1) 1-2-3-4-5-6-45
- 2) 1-2-3-7-8-9-41-42-43-44-45
- 3) 1-2-3-7-8-9-10-11-12-13-14-17-18-19-45
- 4) 1-2-3-7-8-9-10-11-12-13-14-15-16-19-20-21-22-23-24-36-37-38-39-40-45
- 5) 1-2-3-7-8-9-10-11-12-13-14-15-16-19-20-21-22-23-24-25-26-27-31-32-33-34-35-45
- 6) 1-2-3-7-8-9-10-11-12-13-14-15-16-19-20-21-22-23-24-25-26-27-28-29-30-45
- 7) 1-2-3-7-8-9-10-11-12-13-14-15-16-19-20-11-12-13-14-15-16-19-20-21-22-23-24-25-26-27-28-29-30-45

Berdasarkan *independent path*, maka diperoleh *test case* sesuai dengan tabel

Tabel 6.20 Test case input data pembelian

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Submit: Submit gambar: mobil.jpg merk: Toyota type: Agya tahun: <i>NULL</i> nama pemilik: samsuri No Telp: <i>NULL</i> Harga: <i>NULL</i> Spesifikasi mobil: <i>NULL</i> Keterangan: <i>NULL</i>	Kembali ke halaman form input data pembelian dan menampilkan pesan warning bahwa masih ada field kosong	Kembali ke halaman form input data pembelian dan muncul pesan bahwa masih ada <i>field</i> yang kosong	Valid

Tabel 6.20 Test case input data pembelian (lanjutan)

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
2	Submit: Submit gambar: mobil.jpg merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Spesifikasi mobil: <i>NULL</i> Keterangan: <i>NULL</i>	Kembali ke halaman form input data pembelian	Kembali ke halaman form input data pembelian	Valid
3	Submit: Submit gambar: mobil.gif merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Spesifikasi mobil: <i>NULL</i> Keterangan: <i>NULL</i>	Kembali ke halaman form input data pembelian dan menampilkan pesan warning bahwa file yang di unggah tidak diperbolehkan	Kembali ke halaman form input data pembelian dan muncul pesan bahwa file yang diunggah tidak diperbolehkan	Valid
4	Submit: Submit gambar: mobil.jpg merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Spesifikasi mobil: <i>NULL</i> Keterangan: <i>NULL</i>	Kembali ke halaman form input data pembelian	Kembali ke halaman form input data pembelian	Valid
5	Submit: Submit gambar: mobil.jpg merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Spesifikasi mobil: <i>NULL</i> Keterangan: <i>NULL</i>	Kembali ke halaman form input data pembelian	Kembali ke halaman form input data pembelian	Valid

Tabel 6.20 Test case input data pembelian (lanjutan)

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
6	Submit: Submit gambar: mobil.jpg merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Spesifikasi mobil: <i>NULL</i> Keterangan: <i>NULL</i>	Kembali ke halaman form input data pembelian dan menampilkan pesan bahwa data berhasil disimpan	Kembali ke halaman form input data pembelian dan menampilkan pesan bahwa data berhasil disimpan	Valid
7	Submit: Submit gambar: mobil.jpg, mobil1.jpg merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Spesifikasi mobil: <i>NULL</i> Keterangan: <i>NULL</i>	Kembali ke halaman form input data pembelian dan menampilkan pesan bahwa data berhasil disimpan	Kembali ke halaman form input data pembelian dan menampilkan pesan bahwa data berhasil disimpan	Valid

c. Memperbarui data pembelian

Pada tabel merupakan kode program memperbarui data pembelian:

Tabel 6.21 Tabel kode program memperbarui data pembelian

1	<code>function update_pembelian(\$id='') {</code>	1
2	<code> \$path = './assets/uploads/mobil/';</code>	
3	<code> \$gambar = 'aarara';</code>	
4	<code> \$waktu_transaksi = date("y-m-d H:i:s");</code>	
5		
6	<code> \$this->load->model('mMobil');</code>	
7	<code> \$this->load->model('mPembelian');</code>	
8	<code> \$this->load->library('form_validation');</code>	2
9		
10	<code> \$this->form_validation->set_rules('merk', 'Merk Mobil',</code>	
11	<code>'required');</code>	
12	<code> \$this->form_validation->set_rules('type', 'Tipe Mobil',</code>	
13	<code>'required');</code>	
14	<code> \$this->form_validation->set_rules('tahun', 'Tahun Mobil',</code>	

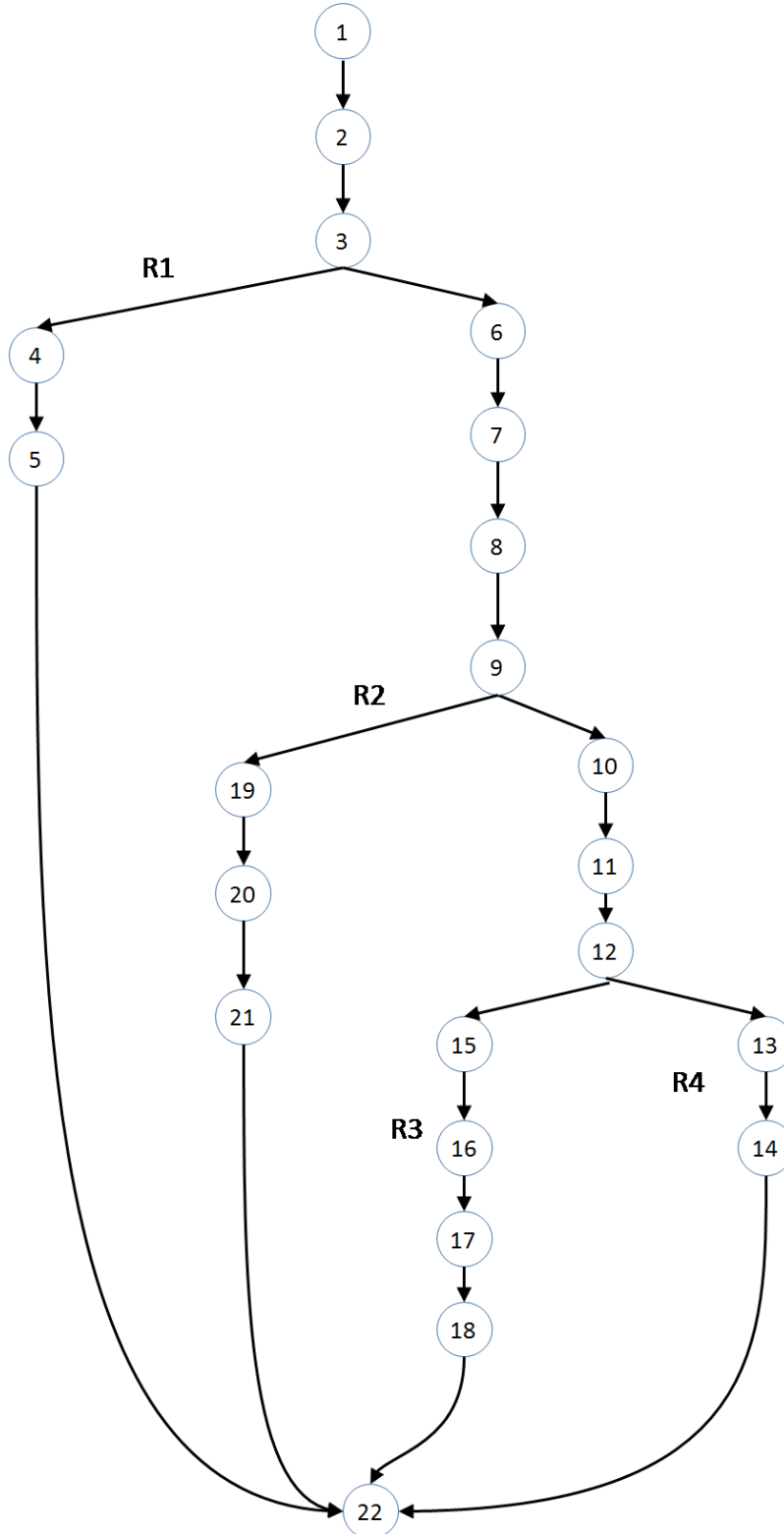
Tabel 6.21 Tabel kode program memperbarui data pembelian (lanjutan)

15	'required');	
16	\$this->form_validation->set_rules('harga', 'Harga Mobil',	
17	'required');	
18	\$this->form_validation->set_rules('nama', 'Nama Pemilik',	
19	'required');	
20	\$this->form_validation->set_rules('noHP', 'Nomor Telepon',	
21	'required');	
22		
23	if (\$this->form_validation->run() == FALSE) {	3
24	\$this->session->set_flashdata('flash_message',	
25	warn_msg(validation_errors()));	4
26	redirect('cPembelian/get_update_pembelian/'.\$id);	
27	}	5
28	else{	6
29	\$datal = array('id_pabrikan' => \$this->input->post('merk'),	
30	'merk' => \$this->input->post('type'),	
31	'harga' => \$this->input-	7
32	>post('harga'),	
33	'tahun' => \$this->input-	
34	>post('tahun'));	
35	\$input = \$this->mMobil->update_mobil_part(\$id, \$datal);	8
36	if (\$input != 0) {	9
37	\$data = array('id_pabrikan' => \$this->input-	
38	>post('merk'),	
39	'merk' => \$this->input-	
40	>post('type'),	
41	'tahun' => \$this->input-	
42	>post('tahun'),	
43	'nama_penjual' => \$this->input-	
44	>post('nama'),	
45	'no_telp' => \$this->input-	10
46	>post('noHP'),	
47	'harga_beli' => \$this->input-	
48	>post('harga'),	
49	'kelengkapan' => \$this->input-	
50	>post('t2'),	
51	'waktu_pembelian' =>	
52	\$waktu_transaksi);	
53	\$simpan = \$this->mPembelian->update_pembelian(\$id,	11
54	\$data);	
55		
56	if (\$simpan != 0) {	12

Tabel 6.21 Tabel kode program memperbarui data pembelian (lanjutan)

57	echo "	
58	<script language='JavaScript'>	
59	alert('Sukses Merubah Data');	13
60	document.location='/SISMOB/cPembelian/get_data_pembelian'	
61	</script>	
62	";	
63	}	14
64	else{	15
65	\$this->session-	
66	>set_flashdata('flash_message', warn_msg('Gagal Menyimpan Data'));	16
67	redirect('cPembelian/get_update_pembelian/'.\$id);	
68	}	17
69	}	18
70	else{	19
71	\$this->session->set_flashdata('flash_message',	
72	warn_msg('Gagal Menyimpan Data'));	20
73	redirect('cPembelian/get_update_pembelian/'.\$id);	
74	}	21
75	}	22
76	}	

Gambar 6.7 merupakan *flow graph* dari memperbaiki data pembelian



Gambar 6.7 Flow graph memperbaiki data pembelian

Berikut merupakan perhitungan *cyclomatic complexity* dari memperbaiki data pembelian:

$$\begin{aligned}
 V(G) &= 4 \\
 V(G) &= E - N + 2 \\
 &= 24 - 22 + 2 \\
 &= 4 \\
 V(G) &= P + 1 \\
 &= 3 + 1 = 4
 \end{aligned}$$

Berikut merupakan *independent path* dari memperbaiki data pembelian:

- 1) 1-2-3-4-5-22
- 2) 1-2-3-6-7-8-9-19-20-21-22
- 3) 1-2-3-6-7-8-9-10-11-12-15-16-17-18-22
- 4) 1-2-3-6-7-8-9-10-11-12-13-14-17-18-22

Berdasarkan *independent path*, maka diperoleh *test case* sesuai dengan table

Tabel 6.22 Test case memperbaiki data pembelian

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Submit: Submit merk: Toyota type: Agya tahun: <i>NULL</i> nama pemilik: samsuri No Telp: <i>NULL</i> Harga: <i>NULL</i> Keterangan: <i>NULL</i>	Kembali ke halaman form update pembelian dan menampilkan pesan warning bahwa masih ada <i>field</i> yang kosong	Kembali ke halaman form update pembelian dan menampilkan pesan warning bahwa masih ada <i>field</i> yang kosong	Valid
2	Submit: Submit merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Keterangan: <i>NULL</i>	Kembali ke halaman form update pembelian	Kembali ke halaman form update pembelian	Valid

Tabel 6.22 Test case memperbarui data pembelian (lanjutan)

Jalur	Data masukan	Hasil yang diharapkan	Hasil yang diperoleh	Status
3	Submit: Submit merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Keterangan: <i>NULL</i>	Kembali ke halaman form update pembelian	Kembali ke halaman form update pembelian	Valid
4	Submit: Submit merk: Toyota type: Agya tahun: 2014 nama pemilik: samsuri No Telp: 0809984041 Harga: 12345678 Keterangan: <i>NULL</i>	Menampilkan halaman list data pembelian	Menampilkan list data pembelian	Valid

6.4 Matriks Kerunutan Pengujian

Matriks kerunutan pengujian digunakan untuk menyesuaikan antara fitur pada aplikasi dengan *test case* pengujian. Matriks kerunutan terdiri dari kode fitur yang telah didefinisikan, kode dasar fungsional, kode lengkap fungsional, *use case*, dan juga *test case*. Pada tabel 6.23 merupakan matriks kerunutan dari pengujian *black box*.

Tabel 6.23 Matriks kerunutan pengujian

Kode Fitur	Kode Dasar Fungsional	Kode Lengkap Fungsional	Use Case	Nomor Skenario	Test case	Validasi
SMOB 1	SRS-F-SISMOB-P01	SRS-F-SISMOB-P01-1	<i>Login</i>	1	<i>Test case 5</i>	Valid
				2	<i>Test case 6</i>	Valid
				3	<i>Test case 7</i>	Valid
				4	<i>Test case 8</i>	Valid
SMOB 2	SRS-F-SISMOB-P02	SRS-F-SISMOB-P02-1	Mengelola data penjualan	1	<i>Test case 12</i>	Valid
		SRS-F-SISMOB-P02-2		2	<i>Test case 9</i>	Valid
		SRS-F-SISMOB-P02-3		3	<i>Test case 10</i>	Valid

Tabel 6.23 Matriks kerunutan pengujian (lanjutan)

SMOB 3	SRS-F-SISMOB-P03	SRS-F-SISMOB-P03-1	Mengelola data pembelian	1	<i>Test case 15</i>	Valid
		SRS-F-SISMOB-P03-2		2	<i>Test case 13</i>	Valid
		SRS-F-SISMOB-P03-3		3	<i>Test case 14</i>	Valid
SMOB 4	SRS-F-SISMOB-P04	SRS-F-SISMOB-P04-1	Mengelola data mobil	1	<i>Test case 21</i>	Valid
		SRS-F-SISMOB-P04-2		2	<i>Test case 19</i>	Valid
		SRS-F-SISMOB-P04-3		3	<i>Test case 20</i>	Valid
SMOB 5	SRS-F-SISMOB-P05	SRS-F-SISMOB-P05-1	Mencetak bukti transaksi	1	<i>Test case 11</i>	Valid
SMOB 6	SRS-F-SISMOB-P06	SRS-F-SISMOB-P06-1	Melakukan pemfilteran data	1	<i>Test case 2</i>	Valid
SMOB 7	SRS-F-SISMOB-P07	SRS-F-SISMOB-P07-1	Melihat laporan keuangan	1	<i>Test case 16</i>	Valid
				2	<i>Test case 17</i>	Valid
				3	<i>Test case 18</i>	Valid
SMOB 8	SRS-F-SISMOB-P08	SRS-F-SISMOB-P08-1	Melihat katalog mobil	1	<i>Test case 1</i>	Valid
SMOB 9	SRS-F-SISMOB-P09	SRS-F-SISMOB-P09-1	Melihat detail mobil	1	<i>Test case 3</i>	Valid
				2	<i>Test case 4</i>	Valid

6.5 Analisis Hasil Pengujian

Berdasarkan hasil pengujian *black box* dan *white box* yang telah dilakukan dapat disimpulkan sebagai berikut:

1. Terdapat 21 *test case* yang berdasarkan kepada skenario dari setiap spesifikasi *use case*.
2. Seluruh *test case* yang diuji dengan menggunakan metode *black box* mendapatkan hasil yang sesuai dengan skenario berdasarkan masing-masing spesifikasi *use case*, sehingga menghasilkan hasil pengujian yang valid.
3. Hasil pengujian *white box* pada *login*, mengelola data penjualan dan mengelola data pembelian telah menghasilkan *flowgraph* dengan hasil

perhitungan *cyclomatic complexity*, *independent path* dan *test case* sebagai berikut:

a. Login

Hasil perhitungan *cyclomatic complexity* dari *login* adalah 4, 4 *independent path* dan 4 *test case*. Setiap *test case* diujikan dan mendapatkan hasil yang valid.

b. Mengelola data penjualan

1) Lihat data penjualan

Hasil perhitungan *cyclomatic complexity* dari lihat data penjualan adalah 1, 1 *independent path* dan 1 *test case*. Setiap *test case* diujikan dan mendapatkan hasil yang valid.

2) Input data penjualan

Hasil perhitungan *cyclomatic complexity* dari input data penjualan adalah 9, 9 *independent path* dan 9 *test case*. Setiap *test case* diujikan dan mendapatkan hasil yang valid.

3) Memperbarui data penjualan

Hasil perhitungan *cyclomatic complexity* dari memperbarui data penjualan adalah 6, 6 *independent path* dan 6 *test case*. Setiap *test case* diujikan dan mendapatkan hasil yang valid.

c. Mengelola data pembelian

1) Lihat data pembelian

Hasil perhitungan *cyclomatic complexity* dari lihat data pembelian adalah 1, 1 *independent path* dan 1 *test case*. Setiap *test case* diujikan dan mendapatkan hasil yang valid.

2) Input data pembelian

Hasil perhitungan *cyclomatic complexity* dari input data pembelian adalah 7, 7 *independent path* dan 7 *test case*. Setiap *test case* diujikan dan mendapatkan hasil yang valid.

3) Memperbarui data pembelian

Hasil perhitungan *cyclomatic complexity* dari memperbarui data pembelian adalah 4, 4 *independent path* dan 4 *test case*. Setiap *test case* diujikan dan mendapatkan hasil yang valid.