

BAB 6 PENGUJIAN

Pada bab ini menjelaskan mengenai proses pengujian dan hasil analisa dari pengujian yang dilakukan. Tujuan dilakukan pengujian ini adalah untuk menemukan kesalahan yang terdapat pada sistem dan memastikan sistem yang dibuat sudah sesuai dengan kebutuhan. Tahap pengujian merupakan tahap *deployment* pada model *waterfall*.

6.1 Pengujian *White-Box*

Pengujian *basis path* merupakan pengujian *white-box* yang dibuat untuk mengukur tingkat kompleksitas dari algoritme hasil perancangan yang nantinya akan digunakan sebagai acuan untuk mendefinisikan jalur eksekusi pada sistem. Pada pengujian *basis path*, algoritme sistem dibentuk ke dalam bentuk *flow graph* atau notasi sederhana yang merepresentasikan aliran kontrol yang nantinya digunakan untuk menghitung jumlah *cyclomatic complexity* atau ukuran kompleksitas dan *independent path* atau jalur independen. Jumlah *cyclomatic complexity* dapat dihitung dengan menggunakan persamaan berikut:

$$V(G) = R \quad (6.1)$$

$$V(G) = E - N + 2 \quad (6.2)$$

$$V(G) = P + 1 \quad (6.3)$$

Keterangan:

$V(G)$: Jumlah ukuran kompleksitas

E : Jumlah tanda panah atau *edge* pada *flow graph*

N : Jumlah *node* pada *flow graph*

P : Jumlah *predicate node* pada *flow graph*

Pengujian yang dilakukan pada sistem informasi pengelolaan keripik buah CV KAJEYE FOOD hanya menguji beberapa proses seperti proses mengelola stok, proses mengelola penjualan dan proses menampilkan peramalan permintaan. Pada sub bab selanjutnya akan menjelaskan pengujian yang dilakukan pada proses-proses tersebut.

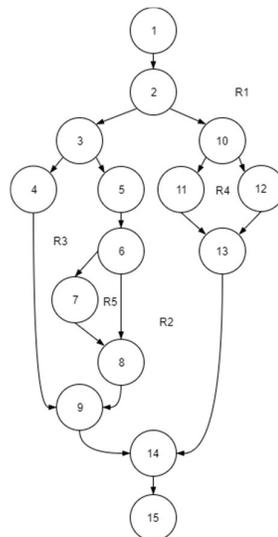
6.1.1 Pengujian Proses Mengelola Stok

Proses mengelola stok menjelaskan bagaimana proses penambahan data perubahan stok berjalan dan kondisi apa saja yang dapat mempengaruhi perubahan stok tersebut. Algoritme proses mengelola stok dapat dilihat pada tabel 6.1.

Tabel 6.1 Algoritme Proses Mengelola Stok

<i>Pseudocode</i>	<i>Node</i>
BEGIN	
Memilih produk yang stoknya akan dikelola Menentukan jumlah perubahan stok Memilih keterangan perubahan stok	1
IF produk yang dipilih sudah memiliki stok THEN	2
IF keterangan stok merupakan stok masuk THEN	3
Stok yang sudah ada ditambahkan jumlah stok yang baru ditentukan	4
ELSE Stok yang sudah ada dikurangi jumlah stok yang baru ditentukan	5
IF jumlah stok yang dikurangi kurang dari 0 THEN	6
Menampilkan pesan kesalahan bahwa stok tidak mencukupi	7
END IF	8
END IF	9
ELSE IF keterangan stok merupakan stok masuk THEN	10
Stok awal produk disamakan dengan jumlah stok yang baru ditentukan	11
ELSE Menampilkan pesan kesalahan bahwa stok belum ada	12
END IF	13
END IF	14
END	15

Berdasarkan algoritme yang ada pada tabel 6.1, maka *flow graph* yang dapat dibuat akan seperti pada gambar 6.1.



Gambar 6.1 Flow Graph Proses Mengelola Stok

Berdasarkan *flow graph* proses mengelola stok pada gambar 6.1, dapat diketahui bahwa terdapat 5 jalur independen seperti berikut:

- Jalur 1: 1-2-3-4-9-14-15
- Jalur 2: 1-2-3-5-6-8-9-14-15
- Jalur 3: 1-2-3-5-6-7-8-9-14-15
- Jalur 4: 1-2-10-11-13-14-15
- Jalur 5: 1-2-10-12-13-14-15

Dan untuk ukuran kompleksitas dari *flow graph* proses mengelola stok pada gambar 6.1 dijelaskan pada perhitungan berikut dengan menggunakan persamaan 6.1, 6.2 dan 6.3:

$$V(G) = R = 5$$

$$V(G) = E - N + 2 = 18 - 15 + 2 = 5$$

$$V(G) = P + 1 = 4 + 1 = 5$$

Berdasarkan hasil perhitungan, ukuran kompleksitas dari *flow graph* proses mengelola stok pada gambar 6.1 adalah 5.

Berdasarkan jumlah jalur independen dari *flow graph* proses mengelola stok pada gambar 6.1, maka kasus uji yang dibuat akan seperti pada tabel 6.2.

Tabel 6.2 Kasus Uji Proses Mengelola Stok

Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Produk yang sudah memiliki stok dan keterangan stok merupakan stok masuk	Data perubahan stok tersimpan	Data perubahan stok tersimpan	<i>Valid</i>
2	Produk yang sudah memiliki stok dan keterangan stok bukan merupakan stok masuk dan hasil pengurangan stok tidak kurang dari 0	Data perubahan stok tersimpan	Data perubahan stok tersimpan	<i>Valid</i>

Tabel 6.2 Kasus Uji Proses Mengelola Stok (lanjutan)

Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
3	Produk yang sudah memiliki stok dan keterangan stok bukan merupakan stok masuk dan hasil pengurangan stok kurang dari 0	Menampilkan pesan kesalahan bahwa stok yang dikurangi tidak mencukupi	Menampilkan pesan kesalahan bahwa stok yang dikurangi tidak mencukupi	<i>Valid</i>
4	Produk yang tidak memiliki stok dan keterangan stok merupakan stok masuk	Data perubahan stok tersimpan	Data perubahan stok tersimpan	<i>Valid</i>
5	Produk yang belum memiliki stok dan keterangan stok bukan merupakan stok masuk	Menampilkan pesan kesalahan bahwa stok belum ada	Menampilkan pesan kesalahan bahwa stok belum ada	<i>Valid</i>

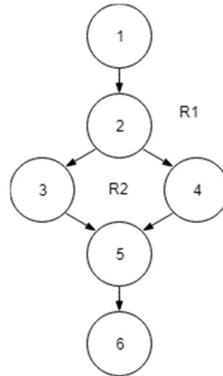
6.1.2 Pengujian Proses Mengelola Penjualan

Proses mengelola penjualan menjelaskan bagaimana proses penambahan data penjualan berjalan dan kondisi apa saja yang mempengaruhi proses tersebut. Algoritme proses mengelola penjualan dapat dilihat pada tabel 6.3.

Tabel 6.3 Algoritme Proses Mengelola Penjualan

<i>Pseudocode</i>	<i>Node</i>
BEGIN	
Menentukan nomor nota penjualan	
Memilih pelanggan atau konsumen yang membeli produk	1
Memilih produk yang dibeli beserta jumlahnya	
Memilih jenis pembelian yang dilakukan	
IF stok produk yang dibeli tidak mencukupi jumlah pembelian THEN	2
Menampilkan pesan kesalahan bahwa stok produk tidak mencukupi	3
ELSE	
Stok produk yang dibeli dikurangi dengan jumlah pembelian produk	
Menyimpan informasi detail data penjualan	4
Menyimpan data perubahan stok dan mencantumkan keterangan penjualan dengan nomor nota yang ditentukan	
END IF	5
END	6

Berdasarkan algoritme yang ada pada tabel 6.3, maka *flow graph* yang dapat dibuat akan seperti pada gambar 6.2.



Gambar 6.2 Flow Graph Proses Mengelola Penjualan

Berdasarkan *flow graph* proses mengelola penjualan pada gambar 6.2, dapat diketahui bahwa terdapat 2 jalur independen seperti berikut:

Jalur 1: 1-2-3-5-6

Jalur 2: 1-2-4-5-6

Dan untuk ukuran kompleksitas dari *flow graph* proses mengelola penjualan pada gambar 6.2 dijelaskan pada perhitungan berikut dengan menggunakan persamaan 6.1, 6.2 dan 6.3:

$$V(G) = R = 2$$

$$V(G) = E - N + 2 = 6 - 6 + 2 = 2$$

$$V(G) = P + 1 = 1 + 1 = 2$$

Berdasarkan hasil perhitungan, ukuran kompleksitas dari *flow graph* proses mengelola penjualan pada gambar 6.2 adalah 2.

Berdasarkan jumlah jalur independen dari *flow graph* proses mengelola penjualan pada gambar 6.2, maka kasus uji yang dibuat akan seperti pada tabel 6.4.

Tabel 6.4 Kasus Uji Proses Mengelola Penjualan

Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Jumlah pembelian produk yang lebih dari jumlah stok produk	Menampilkan pesan kesalahan bahwa stok produk tidak mencukupi	Menampilkan pesan kesalahan bahwa stok produk tidak mencukupi	Valid

Tabel 6.4 Kasus Uji Proses Mengelola Penjualan (lanjutan)

Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
2	Jumlah pembelian produk yang kurang dari jumlah stok produk	Data penjualan dan data perubahan stok tersimpan	Data penjualan dan data perubahan stok tersimpan	Valid

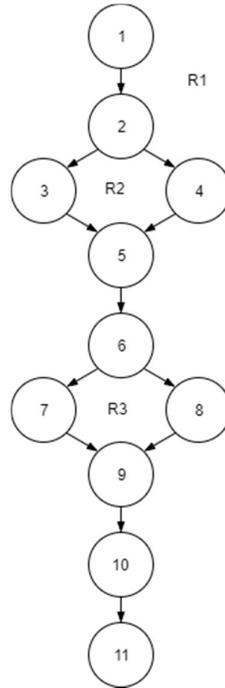
6.1.3 Pengujian Proses Peramalan Permintaan

Proses peramalan permintaan menjelaskan bagaimana proses menampilkan peramalan permintaan yang hasil peramalannya didapatkan dari data penjualan yang sudah tersimpan pada *database*. Algoritme proses peramalan permintaan dapat dilihat pada tabel 6.5.

Tabel 6.5 Algoritme Proses Peramalan Permintaan

Pseudocode	Node
BEGIN	
Menampilkan periode bulan dan tahun yang sudah dipilih	1
Menampilkan alias dari masing-masing produk keripik buah	
IF penjualan produk keripik buah pada periode sebelumnya tersedia	2
THEN	
Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya	3
Menampilkan jumlah stok masing-masing produk keripik buah yang tersisa setelah penjualan terakhir pada periode sebelumnya	
ELSE	
Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya dengan nilai 0	4
Menampilkan jumlah stok masing-masing produk keripik buah dengan nilai 0	
END IF	5
IF penjualan produk keripik buah pada periode yang dipilih tersedia	6
THEN	
Menampilkan jumlah penjualan sebenarnya pada periode yang dipilih	7
ELSE	
Menampilkan jumlah penjualan sebenarnya pada periode yang dipilih dengan nilai 0	8
END IF	9
Menampilkan hasil peramalan menggunakan metode <i>exponential smoothing</i> dan <i>moving averages</i> berdasarkan pada penjualan periode sebelumnya	10
END	11

Berdasarkan algoritme yang ada pada tabel 6.5, maka *flow graph* yang dapat dibuat akan seperti pada gambar 6.3.



Gambar 6.3 Flow Graph Proses Peramalan Permintaan

Berdasarkan *flow graph* proses peramalan permintaan pada gambar 6.3, dapat diketahui bahwa terdapat 4 jalur independen seperti berikut:

- Jalur 1: 1-2-3-5-6-7-9-10-11
- Jalur 2: 1-2-3-5-6-8-9-10-11
- Jalur 3: 1-2-4-5-6-7-9-10-11
- Jalur 4: 1-2-4-5-6-8-9-10-11

Dan untuk ukuran kompleksitas dari *flow graph* proses peramalan permintaan pada gambar 6.3 dijelaskan pada perhitungan berikut dengan menggunakan persamaan 6.1, 6.2 dan 6.3:

$$V(G) = R = 3$$

$$V(G) = E - N + 2 = 12 - 11 + 2 = 3$$

$$V(G) = P + 1 = 2 + 1 = 3$$

Berdasarkan hasil perhitungan, ukuran kompleksitas dari *flow graph* proses peramalan permintaan pada gambar 6.3 adalah 3.

Berdasarkan jumlah jalur independen dari *flow graph* proses peramalan permintaan pada gambar 6.3, maka kasus uji yang dibuat akan seperti pada tabel 6.6.

Tabel 6.6 Kasus Uji Proses Peramalan Permintaan

Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
1	Periode yang pada periode sebelumnya memiliki data penjualan dan memiliki data penjualan pada periode itu sendiri	Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya, menampilkan jumlah penjualan pada periode itu sendiri, menampilkan hasil peramalan dan menampilkan jumlah stok yang tersisa pada periode sebelumnya	Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya, menampilkan jumlah penjualan pada periode itu sendiri, menampilkan hasil peramalan dan menampilkan jumlah stok yang tersisa pada periode sebelumnya	<i>Valid</i>
2	Periode yang pada periode sebelumnya memiliki data penjualan dan tidak memiliki data penjualan pada periode itu sendiri	Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya, menampilkan jumlah penjualan pada periode itu sendiri dengan nilai 0, menampilkan hasil peramalan dan menampilkan jumlah stok yang tersisa pada periode sebelumnya	Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya, menampilkan jumlah penjualan pada periode itu sendiri dengan nilai 0, menampilkan hasil peramalan dan menampilkan jumlah stok yang tersisa pada periode sebelumnya	<i>Valid</i>

Tabel 6.6 Kasus Uji Proses Peramalan Permintaan (lanjutan)

Jalur	Data input	Hasil yang diharapkan	Hasil yang diperoleh	Status
3	Periode yang pada periode sebelumnya tidak memiliki data penjualan dan memiliki data penjualan pada periode itu sendiri	Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya dengan nilai 0, menampilkan jumlah penjualan pada periode itu sendiri, menampilkan hasil peramalan dan menampilkan jumlah stok yang tersisa pada periode sebelumnya dengan nilai 0	Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya dengan nilai 0, menampilkan jumlah penjualan pada periode itu sendiri, menampilkan hasil peramalan dan menampilkan jumlah stok yang tersisa pada periode sebelumnya dengan nilai 0	<i>Valid</i>
4	Periode yang pada periode sebelumnya tidak memiliki data penjualan dan tidak memiliki data penjualan pada periode itu sendiri	Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya dengan nilai 0, menampilkan jumlah penjualan pada periode itu sendiri dengan nilai 0, menampilkan hasil peramalan dan menampilkan jumlah stok yang tersisa pada periode sebelumnya dengan nilai 0	Menampilkan jumlah penjualan tiap produk keripik buah pada periode sebelumnya dengan nilai 0, menampilkan jumlah penjualan pada periode itu sendiri dengan nilai 0, menampilkan hasil peramalan dan menampilkan jumlah stok yang tersisa pada periode sebelumnya dengan nilai 0	<i>Valid</i>

6.2 Pengujian *Black-Box*

Pengujian *black-box* menggunakan strategi pengujian validasi terhadap kebutuhan fungsional yang dilakukan pada saat menjalankan sistem. Pengujian ini dilakukan untuk memastikan bahwa semua kebutuhan fungsional dari sistem informasi pengelolaan penjualan keripik buah CV KAJEYE FOOD telah terpenuhi dan tidak terjadi kesalahan. Pengujian validasi pada sistem informasi pengelolaan keripik buah CV KAJEYE FOOD akan dijelaskan ke dalam tabel-tabel berikut:

Tabel 6.7 Pengujian Validasi Berhasil Menambah Data *User*

Nama Kasus Uji	Berhasil menambah data <i>user</i>
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data <i>user</i> jika data masukan lengkap
Prosedur Uji	1. Menekan tombol tambah pada halaman <i>user</i> 2. Mengisi <i>form</i> tambah data <i>user</i> 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menyimpan data masukan ke <i>database</i>
Hasil Pengujian	Sistem dapat menyimpan data masukan ke <i>database</i>
Status	<i>Valid</i>

Tabel 6.8 Pengujian Validasi Gagal Menambah Data *User*

Nama Kasus Uji	Gagal menambah data <i>user</i>
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data <i>user</i> jika data masukan tidak lengkap
Prosedur Uji	1. Menekan tombol tambah pada halaman <i>user</i> 2. Tidak mengisi <i>form</i> tambah data <i>user</i> secara lengkap 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Hasil Pengujian	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Status	<i>Valid</i>

Tabel 6.9 Pengujian Validasi Berhasil Mengubah Data User

Nama Kasus Uji	Berhasil mengubah data <i>user</i>
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses perubahan data <i>user</i> jika data masukan lengkap
Prosedur Uji	<ol style="list-style-type: none"> 1. Memilih data <i>user</i> yang ingin diubah 2. Menekan tombol ubah pada data tersebut 3. Mengisi <i>form</i> ubah data <i>user</i> 4. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menyimpan perubahan data ke <i>database</i>
Hasil Pengujian	Sistem dapat menyimpan perubahan data ke <i>database</i>
Status	<i>Valid</i>

Tabel 6.10 Pengujian Validasi Gagal Mengubah Data User

Nama Kasus Uji	Gagal mengubah data <i>user</i>
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses perubahan data <i>user</i> jika data masukan tidak lengkap
Prosedur Uji	<ol style="list-style-type: none"> 1. Memilih data <i>user</i> yang ingin diubah 2. Menekan tombol ubah pada data tersebut 3. Tidak mengisi <i>form</i> ubah data <i>user</i> secara lengkap 4. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Hasil Pengujian	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Status	<i>Valid</i>

Tabel 6.11 Pengujian Validasi Menghapus Data User

Nama Kasus Uji	Menghapus data <i>user</i>
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penghapusan data <i>user</i> dengan benar
Prosedur Uji	<ol style="list-style-type: none"> 1. Memilih data <i>user</i> yang ingin dihapus 2. Menekan tombol hapus pada data tersebut 3. Menekan tombol OK

Tabel 6.11 Pengujian Validasi Menghapus Data User (lanjutan)

Hasil Yang Diharapkan	Sistem dapat menghapus data yang dipilih
Hasil Pengujian	Sistem dapat menghapus data yang dipilih
Status	<i>Valid</i>

Tabel 6.12 Pengujian Validasi Berhasil Menambah Data Produk

Nama Kasus Uji	Berhasil menambah data produk
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data produk jika data masukan lengkap
Prosedur Uji	1. Menekan tombol tambah pada halaman produk 2. Mengisi <i>form</i> tambah data produk 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menyimpan data masukan ke <i>database</i>
Hasil Pengujian	Sistem dapat menyimpan data masukan ke <i>database</i>
Status	<i>Valid</i>

Tabel 6.13 Pengujian Validasi Gagal Menambah Data Produk

Nama Kasus Uji	Gagal menambah data produk
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data produk jika data masukan tidak lengkap
Prosedur Uji	1. Menekan tombol tambah pada halaman produk 2. Tidak mengisi <i>form</i> tambah data produk secara lengkap 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Hasil Pengujian	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Status	<i>Valid</i>

Tabel 6.14 Pengujian Validasi Berhasil Mengubah Data Produk

Nama Kasus Uji	Berhasil mengubah data produk
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses perubahan data produk jika data masukan lengkap
Prosedur Uji	1. Memilih data produk yang ingin diubah 2. Menekan tombol ubah pada data tersebut 3. Mengisi <i>form</i> ubah data produk 4. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menyimpan perubahan data ke <i>database</i>
Hasil Pengujian	Sistem dapat menyimpan perubahan data ke <i>database</i>
Status	<i>Valid</i>

Tabel 6.15 Pengujian Validasi Gagal Mengubah Data Produk

Nama Kasus Uji	Gagal mengubah data produk
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses perubahan data produk jika data masukan tidak lengkap
Prosedur Uji	1. Memilih data produk yang ingin diubah 2. Menekan tombol ubah pada data tersebut 3. Tidak mengisi <i>form</i> ubah data produk secara lengkap 4. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Hasil Pengujian	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Status	<i>Valid</i>

Tabel 6.16 Pengujian Validasi Menghapus Data Produk

Nama Kasus Uji	Menghapus data produk
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penghapusan data produk dengan benar

Tabel 6.16 Pengujian Validasi Menghapus Data Produk (lanjutan)

Prosedur Uji	1. Memilih data produk yang ingin dihapus 2. Menekan tombol hapus pada data tersebut 3. Menekan tombol OK
Hasil Yang Diharapkan	Sistem dapat menghapus data yang dipilih
Hasil Pengujian	Sistem dapat menghapus data yang dipilih
Status	<i>Valid</i>

Tabel 6.17 Pengujian Validasi Berhasil Menambah Data Stok

Nama Kasus Uji	Berhasil menambah data stok
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data stok jika data masukan lengkap
Prosedur Uji	1. Menekan tombol tambah pada halaman stok 2. Mengisi <i>form</i> tambah data stok 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menyimpan data masukan ke <i>database</i>
Hasil Pengujian	Sistem dapat menyimpan data masukan ke <i>database</i>
Status	<i>Valid</i>

Tabel 6.18 Pengujian Validasi Gagal Menambah Data Stok

Nama Kasus Uji	Gagal menambah data stok
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data stok jika data masukan tidak lengkap
Prosedur Uji	1. Menekan tombol tambah pada halaman stok 2. Tidak mengisi <i>form</i> tambah data stok secara lengkap 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Hasil Pengujian	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Status	<i>Valid</i>

Tabel 6.19 Pengujian Validasi Berhasil Menambah Data Pelanggan

Nama Kasus Uji	Berhasil menambah data pelanggan
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data pelanggan jika data masukan yang dibutuhkan lengkap
Prosedur Uji	<ol style="list-style-type: none"> 1. Menekan tombol tambah pada halaman pelanggan 2. Mengisi <i>form</i> tambah data pelanggan 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menyimpan data masukan ke <i>database</i>
Hasil Pengujian	Sistem dapat menyimpan data masukan ke <i>database</i>
Status	<i>Valid</i>

Tabel 6.20 Pengujian Validasi Gagal Menambah Data Pelanggan

Nama Kasus Uji	Gagal menambah data pelanggan
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data pelanggan jika data masukan yang dibutuhkan tidak lengkap
Prosedur Uji	<ol style="list-style-type: none"> 1. Menekan tombol tambah pada halaman pelanggan 2. Tidak mengisi <i>form</i> tambah data pelanggan secara lengkap 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Hasil Pengujian	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Status	<i>Valid</i>

Tabel 6.21 Pengujian Validasi Berhasil Mengubah Data Pelanggan

Nama Kasus Uji	Berhasil mengubah data pelanggan
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses perubahan data pelanggan jika data masukan yang dibutuhkan lengkap

Tabel 6.21 Pengujian Validasi Berhasil Mengubah Data Pelanggan (lanjutan)

Prosedur Uji	1. Memilih data pelanggan yang ingin diubah 2. Menekan tombol ubah pada data tersebut 3. Mengisi <i>form</i> ubah data pelanggan 4. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menyimpan perubahan data ke <i>database</i>
Hasil Pengujian	Sistem dapat menyimpan perubahan data ke <i>database</i>
Status	<i>Valid</i>

Tabel 6.22 Pengujian Validasi Gagal Mengubah Data Pelanggan

Nama Kasus Uji	Gagal mengubah data pelanggan
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses perubahan data pelanggan jika data masukan yang dibutuhkan tidak lengkap
Prosedur Uji	1. Memilih data pelanggan yang ingin diubah 2. Menekan tombol ubah pada data tersebut 3. Tidak mengisi <i>form</i> ubah data pelanggan secara lengkap 4. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Hasil Pengujian	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Status	<i>Valid</i>

Tabel 6.23 Pengujian Validasi Menghapus Data Pelanggan

Nama Kasus Uji	Menghapus data pelanggan
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penghapusan data pelanggan dengan benar
Prosedur Uji	1. Memilih data pelanggan yang ingin dihapus 2. Menekan tombol hapus pada data tersebut 3. Menekan tombol OK
Hasil Yang Diharapkan	Sistem dapat menghapus data yang dipilih
Hasil Pengujian	Sistem dapat menghapus data yang dipilih
Status	<i>Valid</i>

Tabel 6.24 Pengujian Validasi Berhasil Menambah Data Penjualan

Nama Kasus Uji	Berhasil menambah data penjualan
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data penjualan jika data masukan yang dibutuhkan lengkap
Prosedur Uji	1. Menekan tombol tambah pada halaman penjualan 2. Mengisi <i>form</i> tambah data penjualan 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menyimpan data masukan ke <i>database</i>
Hasil Pengujian	Sistem dapat menyimpan data masukan ke <i>database</i>
Status	<i>Valid</i>

Tabel 6.25 Pengujian Validasi Gagal Menambah Data Penjualan

Nama Kasus Uji	Gagal menambah data penjualan
Tujuan Pengujian	Untuk memastikan bahwa sistem dapat menangani proses penambahan data penjualan jika data masukan yang dibutuhkan tidak lengkap
Prosedur Uji	1. Menekan tombol tambah pada halaman penjualan 2. Tidak mengisi <i>form</i> tambah data penjualan secara lengkap 3. Menekan tombol simpan
Hasil Yang Diharapkan	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Hasil Pengujian	Sistem dapat menampilkan notifikasi bahwa kolom yang kosong harus diisi
Status	<i>Valid</i>

6.3 Pengujian Metode Peramalan

Pengujian metode peramalan dilakukan untuk mengetahui metode mana yang paling akurat dan bisa digunakan sebagai acuan untuk meramalkan permintaan pada periode yang diinginkan. Pengujian ini dilakukan dengan membandingkan nilai *error* peramalan yang didapat dari selisih hasil peramalan dan jumlah penjualan sebenarnya pada suatu periode. Kemudian, nilai *error* pada tiap periode direkap dan dilakukan perhitungan nilai rata-rata deviasi mutlak (*mean absolute deviation*), rata-rata kuadrat kesalahan (*mean square error*) dan rata-rata persentase kesalahan mutlak (*mean absolute percent error*) untuk mendapatkan hasil akhir keakuratan metode peramalan.

Pada pengujian ini, menggunakan data penjualan keripik apel sebagai objek untuk membandingkan metode peramalan. Jumlah penjualan dari keripik apel dapat dilihat pada tabel 6.26.

Tabel 6.26 Penjualan Keripik Apel

Periode	Penjualan (Bungkus)
Agustus 2016	4.975
September 2016	3.905
Oktober 2016	3.139

Perhitungan berikut merupakan contoh perhitungan dari peramalan yang dilakukan pada periode September 2016 dengan metode *exponential smoothing* menggunakan persamaan 2.2 berdasarkan pada tabel 6.26 yang menggunakan jumlah penjualan keripik apel pada periode Agustus 2016 sebagai acuan dari peramalan pada periode sebelumnya (F_{t-1}), menggunakan jumlah penjualan sebenarnya pada periode Agustus 2016 (A_{t-1}) dan menggunakan konstanta 0,1 untuk mendapatkan hasil peramalan:

$$F_t = F_{t-1} + \alpha (A_{t-1} - F_{t-1})$$

$$F_t = 5.000 + 0,1 (4.975 - 5.000) = 5.000 + (-2,5) = 4.997,5 = 4.998$$

Perhitungan berikut merupakan contoh perhitungan *mean absolute deviation* yang menggunakan persamaan 2.6 berdasarkan hasil peramalan metode *exponential smoothing* dengan konstanta 0,1 pada periode September 2016 dan Oktober 2016:

$$MAD = \sum \left| \frac{A_t - F_t}{n} \right|$$

$$MAD = \frac{|(3.905 - 4.998)| + |(3.139 - 3.991)|}{2} = \frac{1.093}{2} = \frac{1.945}{2} = 972,5$$

Perhitungan berikut merupakan contoh perhitungan *mean square error* yang menggunakan persamaan 2.7 berdasarkan hasil peramalan metode *exponential smoothing* dengan konstanta 0,1 pada periode September 2016 dan Oktober 2016:

$$MSE = \sum \frac{(A_t - F_t)^2}{n}$$

$$MSE = \frac{(3.905 - 4.998)^2 + (3.139 - 3.991)^2}{2} = \frac{1.194.649 + 725.904}{2} = \frac{1.920.553}{2} = 960,276.5$$

Perhitungan berikut merupakan contoh perhitungan *mean absolute percent error* yang menggunakan persamaan 2.8 berdasarkan hasil peramalan metode *exponential smoothing* dengan konstanta 0,1 pada periode September 2016 dan Oktober 2016:

$$MAPE = \left(\frac{100}{n}\right) \sum At - \frac{Ft}{At}$$

$$MAPE = \frac{100(1.093/3.905)+1 (852/3.139)}{2} = \frac{27,99\%+27,14\%}{2} = 27,56\%$$

Tabel-tabel berikut menjelaskan tentang keakuratan tiap metode yang digunakan untuk meramalkan permintaan dengan menerapkan perhitungan seperti contoh sebelumnya.

Tabel 6.27 Nilai Keakuratan Metode Peramalan *Exponential Smoothing* $\alpha = 0,1$

Periode	Penjualan (Bungkus)	Hasil Peramalan		MAD	MSE	MAPE
		Sistem	Excel			
September 2016	3.905	4.998	4.998	1.093	1.194.649	27,99%
Oktober 2016	3.139	3.991	3.991	852	725.904	27,14%
Hasil				972,5	960.276,5	27,56%

Tabel 6.28 Nilai Keakuratan Metode Peramalan *Exponential Smoothing* $\alpha = 0,5$

Periode	Penjualan (Bungkus)	Hasil Peramalan		MAD	MSE	MAPE
		Sistem	Excel			
September 2016	3.905	4.988	4.988	1.083	1.172.889	13,87%
Oktober 2016	3.139	3.953	3.953	814	662.596	25,93%
Hasil				948,5	917.742,5	26,83%

Tabel 6.29 Nilai Keakuratan Metode Peramalan *Exponential Smoothing* $\alpha = 0,9$

Periode	Penjualan (Bungkus)	Hasil Peramalan		MAD	MSE	MAPE
		Sistem	Excel			
September 2016	3.905	4.978	4.978	1.073	1.151.329	27,48%
Oktober 2016	3.139	3.915	3.915	776	602.176	24,72%
Hasil				924,5	876.752,5	26,1%

Tabel 6.30 Nilai Keakuratan Metode Peramalan *Moving Averages*

Periode	Penjualan (Bungkus)	Hasil Peramalan		MAD	MSE	MAPE
		Sistem	Excel			
September 2016	3.905	2.488	2.488	1.417	2.007.889	36,29%
Oktober 2016	3.139	4.440	4.440	1.301	1.692.601	41,45%
Hasil				1.359	1.850.245	38,87%

Pada tabel 6.27 sampai dengan tabel 6.30 terdapat hasil peramalan menggunakan sistem dan menggunakan *Microsoft Excel*. Kedua hasil peramalan tersebut menunjukkan bahwa perhitungan menggunakan sistem dan menggunakan *Microsoft Excel* atau perhitungan secara manual mendapatkan hasil yang sama. Tetapi, pada saat melakukan perhitungan secara manual, memerlukan waktu kurang lebih 5 menit. Sedangkan perhitungan yang dilakukan oleh sistem informasi hanya memerlukan waktu kurang dari 2 menit untuk dapat menghasilkan hasil peramalan pada periode yang diinginkan.

Dan berdasarkan nilai keakuratan masing-masing metode peramalan yang dijelaskan pada tabel 6.27 sampai dengan tabel 6.30, maka dapat diketahui bahwa peramalan dengan menggunakan metode *exponential smoothing* dengan konstanta 0,9 adalah metode peramalan yang paling akurat dibanding metode lainnya karena memiliki nilai *error* yang paling rendah yaitu, MAD (*Mean Absolute Deviation*) sebesar 924,5, MSE (*Mean Square Error*) sebesar 876.752,5 dan MAPE (*Mean Absolute Percentage Error*) sebesar 26,1%.