

BAB 7 PENGUJIAN

Pada bab ini menjelaskan mengenai pengujian yang akan dilakukan pada sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan berbasis web pada Son Screen Printing Sidoarjo. Pengujian yang dilakukan terbagi menjadi pengujian fungsional, non-fungsional, dan *user acceptance testing*.

7.1 Pengujian Fungsional

Sub bab ini menjelaskan mengenai pengujian dari kebutuhan fungsional yang telah dijelaskan pada sub bab analisis kebutuhan fungsional dan diimpelentasikan menjadi sistem informasi pada Son Screen Printing. Pengujian fungsional yang dilakukan adalah *White-Box Testing* dan *Black-Box Testing*.

7.1.1 White-Box Testing

Pengujian ini dilakukan untuk dapat meneliti dan menganalisa jalannya kode dari program yang di buat, apakah ada yang salah atau tidak. Pengujian yang dilakukan meliputi seluruh keputusan yang menggunakan logika, keseluruhan pengulangan yang sesuai batasannya, dan struktur data yang sifatnya internal dan yang terjamin validasinya.

7.1.1.1 Pengujian Pseudocode Menghitung Harga Pokok Produksi dengan *Basis Path Testing*

Algoritma yang sudah mendekati bahasa pemrograman dan telah dipetakan untuk memudahkan dalam membuat *flow graph* dan menghitung *cyclomatic complexity* dari *function* menghitungHargaPokokProduksi() ditunjukkan pada Tabel 7.1 sebagai berikut.

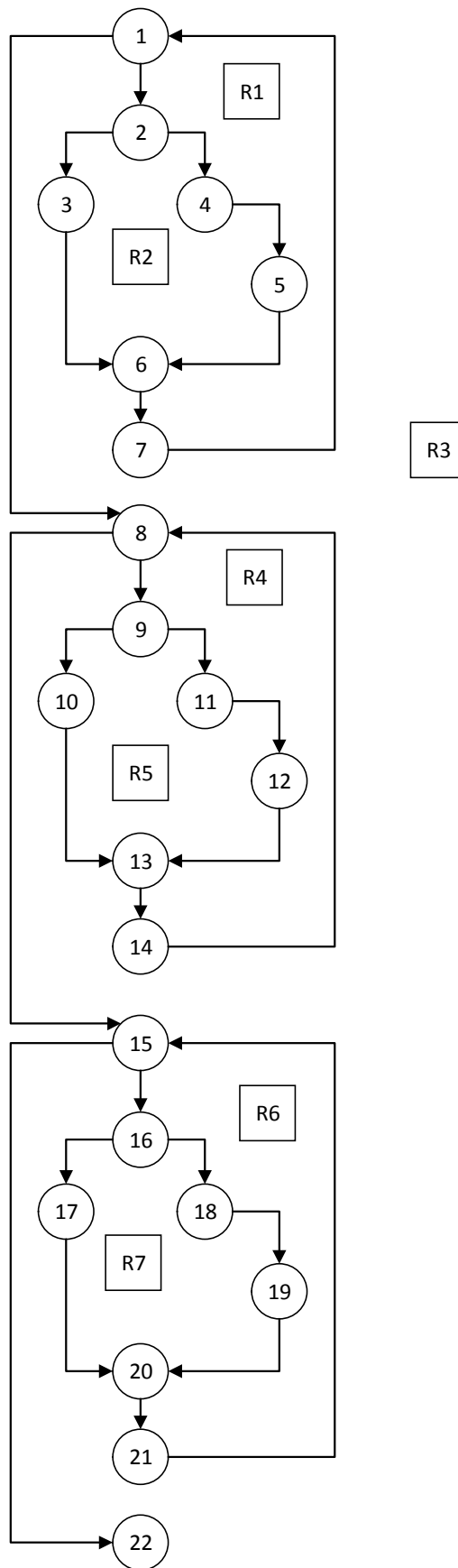
Tabel 7.1 Pengujian Pseudocode Menghitung Harga Pokok Produksi

1.	biaya_bahan_baku = inputan_biaya_bahan_baku
	total_biaya_bahan_baku = 0
	for (index from 1 to biaya_bahan_baku)
2.	if (inputan_jumlah_biaya_bahan_baku != NULL inputan_jumlah_biaya_bahan_baku != 0)
3.	nama = inputan_nama_bahan_baku
	jumlah = inputan_jumlah_bahan_baku
	harga = inputan_harga_bahan_baku
	dataProduksi['nama_bahan_baku'][index] = nama
	dataProduksi['jumlah_bahan_baku'][index] = jumlah
	dataProduksi['harga_bahan_baku'][index] = harga

	total_biaya_bahan_baku += (jumlah * harga)
4.	else
5.	biaya_bahan_baku--
6.	end if
7.	end for
8.	dataProduksi['biaya_bahan_baku'] = biaya_bahan_baku
	dataProduksi['total_biaya_bahan_baku'] = total_biaya_bahan_baku
	biaya_tenaga_kerja_langsung = inputan_biaya_tenaga_kerja_langsung
	total_biaya_tenaga_kerja_langsung = 0
	for (index from 1 to biaya_tenaga_kerja_langsung)
9.	if (inputan_jumlah_biaya_tenaga_kerja_langsung != NULL inputan_jumlah_biaya_tenaga_kerja_langsung != 0)
10.	nama = inputan_nama_tenaga_kerja_langsung
	jumlah = inputan_jumlah_tenaga_kerja_langsung
	harga = inputan_harga_tenaga_kerja_langsung
	dataProduksi['nama_tenaga_kerja_langsung'][index] = nama
	dataProduksi['jumlah_tenaga_kerja_langsung'][index] = jumlah
	dataProduksi['harga_tenaga_kerja_langsung'][index] = harga
	total_biaya_tenaga_kerja_langsung += (jumlah * harga)
11.	else
12.	biaya_tenaga_kerja_langsung --
13.	end if
14.	end for
15.	dataProduksi['biaya_tenaga_kerja_langsung'] = biaya_ tenaga_kerja_langsung
	dataProduksi['total_biaya_tenaga_kerja_langsung'] = total_biaya_ tenaga_kerja_langsung
	biaya_overhead_variabel = inputan_biaya_overhead_variabel
	total_biaya_overhead_variabel = 0
	for (index from 1 to biaya_overhead_variabel)
16.	if (inputan_jumlah_biaya_overhead_variabel != NULL inputan_jumlah_biaya_overhead_variabel != 0)

17.	nama = inputan_nama_overhead_variabel
	jumlah = inputan_jumlah_overhead_variabel
	harga = inputan_harga_overhead_variabel
	dataProduksi['nama_overhead_variabel'][index] = nama
	dataProduksi['jumlah_overhead_variabel'][index] = jumlah
	dataProduksi['harga_overhead_variabel'][index] = harga
	total_biaya_overhead_variabel += (jumlah * harga)
18.	else
19.	biaya_overhead_variabel --
20.	end if
21.	end for
22.	dataProduksi['biaya_overhead_variabel'] = biaya_overhead_variabel
	dataProduksi['total_biaya_overhead_variabel'] = total_biaya_overhead_variabel
	dataProduksi['total_produksi'] = total_biaya_bahan_baku + total_biaya_tenaga_kerja_langsung + total_overhead_variabel
	view('totalHargaPokokProduksi', dataProduksi)

Dengan dipetakannya pseudocode dari menghitung harga pokok produksi dapat digambarkan *flow graph* yang menggambarkan alur sederhana dari pseudocode yang telah dipetakan. *Flow graph* menghitung harga pokok produksi ditunjukkan pada Gambar 7.1 sebagai berikut.



Gambar 7.1 Flow Graph Menghitung Harga Pokok Produksi

Gambar 7.1 merupakan *flow graph* dari menghitung harga pokok produksi yang berguna untuk mempermudah perhitungan *cyclomatic complexity*. *Flow graph* Gambar 7.1 dibuat berdasarkan pemetaan dari Tabel 7.1. Sebelum melakukan perhitungan perlu dibuat jalur independen yaitu sebuah kondisi pada program yang menghubungkan node awal hingga mencapai node akhir atau jalur yang melalui program dimana sekurang-kurangnya terdapat proses perintah baru atau kondisi baru. Setelah dilakukan penghitungan terdapat 7 jalur yang dapat diproses pada proses menghitung harga pokok produksi yaitu sebagai berikut:

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

Setiap jalur independen yang telah didefinisikan diatas memiliki kondisi yang berbeda-beda. Kondisi setiap jalur independen menghitung harga pokok produksi dapat dilihat pada Tabel 7.2 yang merupakan test case dari setiap jalur independen.

Tabel 7.2 Test Case Menghitung Harga Pokok Produksi

No	Jalur	Test Case
1.	1-8-15-22	Jalur ini terjadi ketika masing-masing jumlah masukan biaya bahan baku, biaya tenaga kerja langsung, dan biaya overhead variabel aktor pada form input halaman menentukan harga pokok produksi bernilai 0.
2.	1-2-3-6-7-1-8-15-22	Jalur ini terjadi ketika jumlah masukan biaya bahan baku pada form input halaman menentukan harga pokok produksi tidak sama dengan 0 dan jumlah masing-masing masukan jumlahBBB sama dengan NULL atau 0. Namun, jumlah masukan biaya tenaga kerja langsung dan overhead variabel sama dengan 0.

Tabel 7.2 Test Case Menghitung Harga Pokok Produksi (lanjutan)

No	Jalur	Test Case
3.	1-2-4-5-6-7-1-8-15-22	Jalur ini terjadi ketika jumlah masukan biaya bahan baku pada form input halaman menentukan harga pokok produksi tidak sama dengan 0 dan jumlah masing-masing masukan jumlahBBB tidak sama dengan NULL atau 0. Namun, jumlah masukan biaya tenaga kerja langsung dan overhead variabel sama dengan 0.
4.	1-8-9-10-13-14-8-15-22	Jalur ini terjadi ketika jumlah masukan biaya tenaga kerja langsung pada form input halaman menentukan harga pokok produksi tidak sama dengan 0 dan jumlah masing-masing masukan jumlahBBB sama dengan NULL atau 0. Namun, jumlah masukan biaya bahan baku dan overhead variabel sama dengan 0.
5.	1-8-9-11-12-13-14-8-15-22	Jalur ini terjadi ketika jumlah masukan biaya tenaga kerja langsung pada form input halaman menentukan harga pokok produksi tidak sama dengan 0 dan jumlah masing-masing masukan jumlahBBB tidak sama dengan NULL atau 0. Namun, jumlah masukan biaya bahan baku dan overhead variabel sama dengan 0.
6.	1-8-15-16-17-20-21-15-22	Jalur ini terjadi ketika jumlah masukan biaya overhead variabel pada form input halaman menentukan harga pokok produksi tidak sama dengan 0 dan jumlah masing-masing masukan jumlahBBB sama dengan NULL atau 0. Namun, jumlah masukan biaya bahan baku dan tenaga kerja langsung sama dengan 0.
7.	1-8-15-16-18-19-20-21-15-22	Jalur ini terjadi ketika jumlah masukan biaya overhead variabel pada form input halaman menentukan harga pokok produksi tidak sama dengan 0 dan jumlah masing-masing masukan jumlahBBB tidak sama dengan NULL atau 0. Namun, jumlah masukan biaya bahan baku dan tenaga kerja langsung sama dengan 0.

Cyclomatic complexity digunakan untuk menentukan banyaknya jalur independen. Berdasarkan Gambar 7.1 dapat dilakukan perhitungan *cyclomatic complexity* pada proses menghitung harga pokok produksi yang ditunjukkan sebagai berikut:

$$V(G) = \text{EDGE} - \text{NODE} + 2$$

$$V(G) = 27 - 22 + 2 = 7$$

$$V(G) = \text{PREDIKAT} + 1$$

$$V(G) = 6 + 1 = 7$$

$$V(G) = \text{REGION}$$

$$V(G) = 7$$

Berdasarkan hasil pengujian di atas, pseudocode dari fungsi menghitung harga pokok produksi dari *controller c_produksi* memiliki 7 jalur independen dan hasil perhitungan *cyclomatic complexity* juga menghasilkan $V(G)=7$. Jadi dapat disimpulkan bahwa, jalur utama yang dapat dilalui berjumlah 7 jalur.

7.1.1.2 Pengujian Pseudocode Simpan Data Transaksi Penjualan dengan *Basis Path Testing*

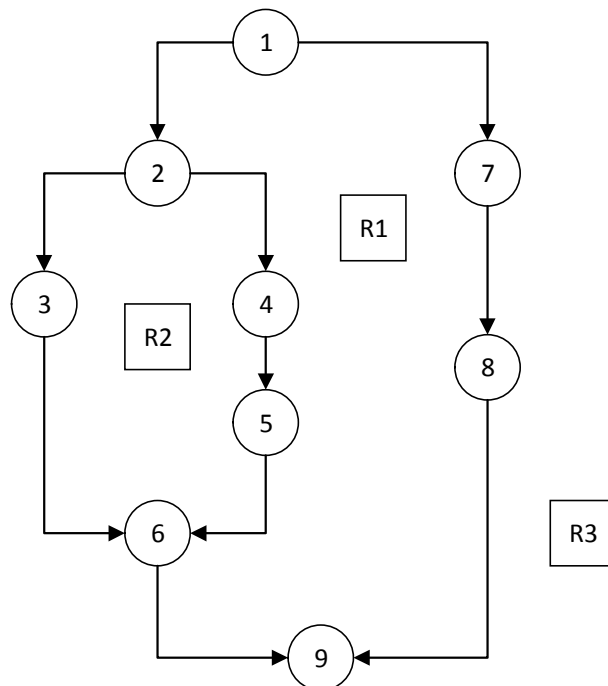
Algoritma yang sudah mendekati bahasa pemrograman dan telah dipetakan untuk memudahkan dalam membuat *flow graph* dan menghitung *cyclomatic complexity* dari *function* `simpanDataTransaksiPenjualan()` ditunjukkan pada Tabel 7.3 sebagai berikut.

Tabel 7.3 Pengujian Pseudocode Simpan Data Transaksi Penjualan

1.	dataStatus = userdata('logged_user')
	if (data = "Marketing")
2.	dataTransaksi = array (nama_pembeli = inputan_nama_pembeli, persediaan_barang_id_barang = inputan_pembelian, jenis_barang_pembelian = inputan_jenis_barang_pembelian, jumlah_pembelian = inputan_jumlah_pembelian, tanggal_pembelian = inputan_tanggal_pembelian, harga = inputan_harga)
	status_simpan_data = simpanDataTransaksiPenjualan(dataTransaksi)
	if (status_simpan_data == TRUE)
3.	dataTransaksi['notif'] = 'Data Berhasil Disimpan'
	dataTransaksi['transaksi'] = getTransaksiPenjualanKelola()
	view('kelolaTransaksiPenjualan', dataTransaksi)

4.	else
5.	dataTransaksi['notif'] = 'Data Gagal Disimpan'
	dataTransaksi['transaksi'] = getTransaksiPenjualanKelola()
	view('kelolaTransaksiPenjualan', dataTransaksi)
6.	end if
7.	else
8.	session = 'Waktu Session Habis'
	data['notif'] = session
	view('Login', data)
9.	end if

Dengan dipetakannya pseudocode dari simpan data transaksi penjualan dapat digambarkan *flow graph* yang menggambarkan alur sederhana dari pseudocode yang telah dipetakan. *Flow graph* simpan data transaksi penjualan ditunjukkan pada Gambar 7.2 sebagai berikut.



Gambar 7.2 Flow Graph Simpan Data Transaksi Penjualan

Gambar 7.2 merupakan *flow graph* dari simpan data transaksi penjualan yang berguna untuk mempermudah perhitungan *cyclomatic complexity*. *Flow graph* Gambar 7.2 dibuat berdasarkan pemetaan dari Tabel 7.4. Sebelum melakukan perhitungan perlu dibuat jalur independen yaitu sebuah kondisi pada program yang menghubungkan node awal hingga mencapai node akhir atau jalur yang melalui program dimana sekurang-kurangnya terdapat proses perintah baru atau

kondisi baru. Setelah dilakukan penghitungan terdapat 3 jalur yang dapat diproses pada proses simpan data transaksi penjualan yaitu sebagai berikut:

1. Jalur 1 = **1-7-8-9**
2. Jalur 2 = **1-2-3-6-9**
3. Jalur 3 = **1-2-4-5-6-9**

Setiap jalur independen yang telah didefinisikan diatas memiliki kondisi yang berbeda-beda. Kondisi setiap jalur independen simpan data transaksi penjualan dapat dilihat pada Tabel 7.4 yang merupakan test case dari setiap jalur independen.

Tabel 7.4 Test Case Simpan Data Transaksi Penjualan

No	Jalur	Test Case
1.	1-7-8-9	Jalur ini terjadi ketika aktor belum login sebagai marketing atau waktu session telah habis. Sehingga akan menampilkan halaman login untuk melakukan login kembali.
2.	1-2-3-6-9	Jalur ini terjadi ketika aktor telah login sebagai marketing dan penyimpanan data transaksi penjualan berhasil disimpan sehingga menampilkan halaman kelola transaksi penjualan dengan pesan 'Data Berhasil Disimpan'.
3.	1-2-4-5-6-9	Jalur ini terjadi ketika aktor telah login sebagai marketing dan penyimpanan data transaksi penjualan gagal disimpan sehingga menampilkan halaman kelola transaksi penjualan dengan pesan 'Data Gagal Disimpan'.

Cyclomatic complexity digunakan untuk menentukan banyaknya jalur independen. Berdasarkan Gambar 7.2 dapat dilakukan perhitungan *cyclomatic complexity* pada proses simpan data transaksi penjualan yang ditunjukkan sebagai berikut:

$$V(G) = \text{EDGE} - \text{NODE} + 2$$

$$V(G) = 10 - 9 + 2 = 3$$

$$V(G) = \text{PREDIKAT} + 1$$

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

$$V(G) = \text{REGION}$$

$$V(G) = 3$$

Berdasarkan hasil pengujian di atas, pseudocode dari fungsi simpan data transaksi penjualan dari *controller c_TransaksiPenjualan* memiliki 3 jalur independen dan hasil perhitungan *cyclomatic complexity* juga menghasilkan $V(G)=3$. Jadi dapat disimpulkan bahwa, jalur utama yang dapat dilalui berjumlah 3 jalur.

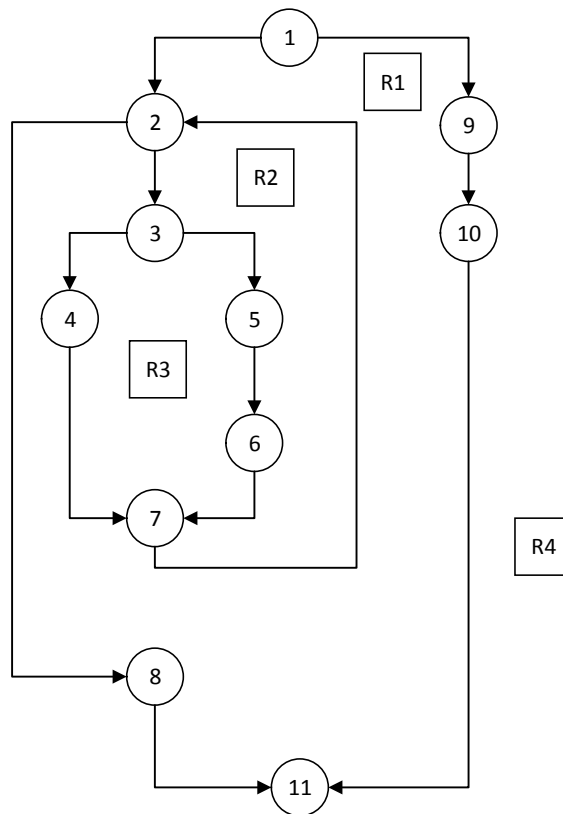
7.1.1.3 Pengujian Pseudocode Tampilkan Laporan pendukung SPT Tahunan dengan Basis Path Testing

Algoritma yang sudah mendekati bahasa pemrograman dan telah dipetakan untuk memudahkan dalam membuat *flow graph* dan menghitung *cyclomatic complexity* dari *function* tampilkanLaporanSptTahunan() ditunjukkan pada Tabel 7.5 sebagai berikut.

Tabel 7.5 Pengujian Pseudocode Tampilkan Laporan pendukung SPT Tahunan

1.	data = userdata('logged_user')
	if (data = "TRUE")
2.	dataTransaksi['transaksi'] = getTransaksiPenjualanLaporan()
	dataTransaksi['barang'] = getPersediaanBarang()
	dataTransaksi['omset'] = 0
	dataTransaksi['produksi'] = 0
	foreach (dataTransaksi['transaksi'] as transaksi)
3.	if (transaksi->tahun == date('Y'))
4.	dataTransaksi['omset'] += transaksi->harga
	dataTransaksi['produksi'] += transaksi->harga_barang
5.	else
6.	continue
7.	end if
8.	end for
	dataTransaksi['labaRugi'] = dataTransaksi['omset'] - dataTransaksi['produksi']
	dataTransaksi['neto'] = dataTransaksi['labaRugi']
	view('laporanSptTahunan', dataTransaksi)
9.	else
10.	session = 'Waktu Session Habis'
	data['notif'] = session
	view('Login', data)
11.	end if

Dengan dipetakannya pseudocode dari tampilkan laporan pendukung SPT Tahunan dapat digambarkan *flow graph* yang menggambarkan alur sederhana dari pseudocode yang telah dipetakan. *Flow graph* tampilkan laporan pendukung SPT Tahunan ditunjukkan pada Gambar 7.3 sebagai berikut.



Gambar 7.3 Flow Graph Tampilkan Laporan pendukung SPT Tahunan

Gambar 7.3 merupakan *flow graph* dari tampilan laporan pendukung SPT Tahunan yang berguna untuk mempermudah perhitungan *cyclomatic complexity*. *Flow graph* Gambar 7.3 dibuat berdasarkan pemetaan dari Tabel 7.6. Sebelum melakukan perhitungan perlu dibuat jalur independen yaitu sebuah kondisi pada program yang menghubungkan node awal hingga mencapai node akhir atau jalur yang melalui program dimana sekurang-kurangnya terdapat proses perintah baru atau kondisi baru. Setelah dilakukan penghitungan terdapat 4 jalur yang dapat diproses pada proses tampilan laporan pendukung SPT Tahunan yaitu sebagai berikut:

1. Jalur 1 = **1-9-10-11**
2. Jalur 2 = **1-2-8-11**
3. Jalur 3 = **1-2-3-4-7-2-8-11**
4. Jalur 4 = **1-2-3-5-6-7-2-8-11**

Setiap jalur independen yang telah didefinisikan diatas memiliki kondisi yang berbeda-beda. Kondisi setiap jalur independen tampilan laporan pendukung SPT Tahunan dapat dilihat pada Tabel 7.6 yang merupakan test case dari setiap jalur independen.

Tabel 7.6 Test Case Tampilkan Laporan pendukung SPT Tahunan

No	Jalur	Test Case
1.	1-9-10-11	Jalur ini terjadi ketika aktor belum login atau waktu session telah habis. Sehingga akan menampilkan halaman login untuk melakukan login kembali.
2.	1-2-8-11	Jalur ini terjadi ketika status login sukses namun data transaksi penjualan sama dengan NULL atau 0
3.	1-2-3-4-7-2-8-11	Jalur ini terjadi ketika status login sukses dan data transaksi penjualan tidak sama dengan NULL atau 0 yang setiap data transaksi penjualan terdapat data dengan tahun transaksi sama dengan tahun sekarang.
4.	1-2-3-5-6-7-2-8-11	Jalur ini terjadi ketika status login sukses dan data transaksi penjualan tidak sama dengan NULL atau 0 yang setiap data transaksi penjualan terdapat data dengan tahun transaksi tidak sama dengan tahun sekarang.

Cyclomatic complexity digunakan untuk menentukan banyaknya jalur independen. Berdasarkan Gambar 7.3 dapat dilakukan perhitungan *cyclomatic complexity* pada proses tampilkan laporan pendukung SPT Tahunan yang ditunjukkan sebagai berikut:

$$V(G) = \text{EDGE} - \text{NODE} + 2$$

$$V(G) = 12 - 11 + 2 = 4$$

$$V(G) = \text{PREDIKAT} + 1$$

$$V(G) = 3 + 1 = 4$$

$$V(G) = \text{REGION}$$

$$V(G) = 4$$

Berdasarkan hasil pengujian di atas, pseudocode dari fungsi tampilkan laporan pendukung SPT Tahunan dari *controller c_SPTTahunan* memiliki 4 jalur independen dan hasil perhitungan *cyclomatic complexity* juga menghasilkan $V(G)=4$. Jadi dapat disimpulkan bahwa, jalur utama yang dapat dilalui berjumlah 4 jalur.

7.1.2 Black-Box Testing

Black-Box Testing merupakan pengujian fungsional karena penguji hanya dikonsentrasikan terhadap fungsionalitas dan tidak pengimplementasi dari perangkat lunak. Pengujian ini penting dilakukan dikarenakan yang dilihat pertama kali dari jalannya perangkat lunak adalah fungsionalitas sistem dari luar yaitu mulai dari inputan user, aksi dari user, maupun yang lainnya. Sehingga sistem informasi persediaan barang, harga pokok produksi, dan transaksi

penjualan berbasis web pada Son Screen Printing Sidoarjo harus dilakukan *black-box testing* yang terdiri dari *validation testing* dan *user acceptance testing*.

7.1.2.1 Validation Testing

Validation testing digunakan untuk mengetahui apakah sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan berbasis web pada Son Screen Printing Sidoarjo yang dibangun sudah sesuai dengan yang dibutuhkan oleh perusahaan. Pengujian validasi dijelaskan pada Tabel 7.7 sebagai berikut.

Tabel 7.7 Validation Testing

Kode	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
SKPL_SI_F_01	Login	Pengguna mengisi username dan password yang benar. Kemudian memilih button Login	Halaman grafik penjualan ditampilkan	Halaman grafik penjualan ditampilkan	Valid
		Pengguna mengisi username atau password yang salah. Kemudian memilih button Login	Halaman login ditampilkan dengan notifikasi bahwa username atau password salah	Halaman login ditampilkan dengan notifikasi bahwa username atau password salah	Valid
SKPL_SI_F_02	Lihat grafik penjualan	Pengguna memilih menu grafik penjualan	Halaman grafik penjualan ditampilkan	Halaman grafik penjualan ditampilkan	Valid
SKPL_SI_F_03	Kelola persediaan barang	Produksi memilih menu kelola persediaan barang	Halaman kelola persediaan barang ditampilkan disertai dengan data persediaan barang	Halaman kelola persediaan barang ditampilkan disertai dengan data persediaan barang	Valid

Tabel 7.7 Validation Testing (lanjutan)

Kode	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
SKPL_SI_F_03	Kelola persediaan barang	Produksi menekan tombol tambah persediaan barang, kemudian memasukkan data nama barang, jenis barang, jumlah barang, harga barang, dan tanggal barang masuk dan keluar.	Halaman form input persediaan barang ditampilkan dan data persediaan barang ditambahkan sesuai dengan inputan Produksi. Kemudian menampilkan halaman kelola persediaan barang yang telah diperbarui ditampilkan	Halaman form input persediaan barang ditampilkan dan data persediaan barang ditambahkan sesuai dengan inputan Produksi. Kemudian menampilkan halaman kelola persediaan barang yang telah diperbarui ditampilkan	Valid
		Produksi menekan tombol edit persediaan barang, kemudian memasukkan data nama barang, jenis barang, jumlah barang, harga barang, dan tanggal barang masuk dan keluar yang ingin diubah.	Halaman form edit persediaan barang ditampilkan dan data persediaan barang diubah sesuai dengan inputan Produksi. Kemudian menampilkan halaman kelola persediaan barang yang telah diperbarui ditampilkan.	Halaman form edit persediaan barang ditampilkan dan data persediaan diubah sesuai dengan inputan Produksi. Kemudian menampilkan halaman kelola persediaan barang yang telah diperbarui ditampilkan.	Valid
		Produksi menekan tombol hapus persediaan barang pada salah satu data persediaan barang.	Data persediaan barang dihapus sesuai dengan pilihan Produksi dan menampilkan halaman kelola persediaan barang yang telah diperbarui ditampilkan	Data persediaan barang dihapus sesuai dengan pilihan Produksi dan menampilkan halaman kelola persediaan barang yang telah diperbarui ditampilkan	Valid

Tabel 7.7 Validation Testing (lanjutan)

Kode	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
SKPL_SI_F_04	Lihat laporan persediaan barang	Pengguna memilih menu laporan persediaan barang	Halaman laporan persediaan barang ditampilkan	Halaman laporan persediaan barang ditampilkan	Valid
SKPL_SI_F_05	Cetak laporan persediaan barang	Owner atau produksi memilih tombol cetak laporan	Halaman print preview ditampilkan	Halaman print preview ditampilkan	Valid
SKPL_SI_F_06	Kelola transaksi penjualan	Marketing memilih menu kelola transaksi penjualan	Halaman kelola transaksi penjualan ditampilkan disertai dengan data transaksi penjualan	Halaman kelola transaksi penjualan ditampilkan disertai dengan data transaksi penjualan	Valid
		Marketing menekan tombol tambah transaksi penjualan, kemudian memasukkan data nama pembeli, pembelian, jenis order, jumlah pembelian, tanggal pembelian, dan harga.	Halaman form input transaksi penjualan ditampilkan dan data transaksi penjualan ditambahkan sesuai dengan inputan Marketing. Kemudian halaman transaksi penjualan yang telah diperbarui ditampilkan	Halaman form input transaksi penjualan ditampilkan dan data transaksi penjualan ditambahkan sesuai dengan inputan Marketing. Kemudian halaman transaksi penjualan yang telah diperbarui ditampilkan	Valid

Tabel 7.7 Validation Testing (lanjutan)

Kode	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
SKPL_SI_F_06	Kelola transaksi penjualan	Marketing menekan tombol edit transaksi penjualan, kemudian memasukkan data nama pembeli, pembelian, jenis order, jumlah pembelian, tanggal pembelian, dan harga yang ingin diubah.	Halaman form edit transaksi penjualan ditampilkan dan data transaksi penjualan diubah sesuai dengan inputan Marketing. Kemudian halaman kelola transaksi penjualan yang telah diperbarui ditampilkan.	Halaman form edit transaksi penjualan ditampilkan dan data transaksi penjualan diubah sesuai dengan inputan Marketing. Kemudian halaman kelola transaksi penjualan yang telah diperbarui ditampilkan.	Valid
		Marketing menekan tombol hapus transaksi penjualan pada salah satu data transaksi penjualan.	Data transaksi penjualan dihapus sesuai dengan pilihan Marketing dan halaman kelola transaksi penjualan yang telah diperbarui ditampilkan	Data transaksi penjualan dihapus sesuai dengan pilihan Marketing dan halaman kelola transaksi penjualan yang telah diperbarui ditampilkan	Valid
SKPL_SI_F_07	Lihat laporan transaksi penjualan	Pengguna memilih menu laporan transaksi penjualan	Halaman laporan transaksi penjualan ditampilkan	Halaman laporan transaksi penjualan ditampilkan	Valid
SKPL_SI_F_08	Cetak laporan transaksi penjualan	Owner atau marketing memilih tombol cetak laporan.	Halaman print preview ditampilkan	Halaman print preview ditampilkan	Valid

Tabel 7.7 Validation Testing (lanjutan)

Kode	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
SKPL_SI_F_09	Menentukan harga pokok produksi	Pengguna mengisi harga dan jumlah pokok produksi. Kemudian memilih tombol hitung dan melihat halaman total harga pokok produksi	Halaman total harga pokok produksi ditampilkan	Halaman total harga pokok produksi ditampilkan	Valid
SKPL_SI_F_10	Kelola harga pokok produksi	Produksi memilih menu kelola transaksi penjualan	Halaman kelola transaksi penjualan ditampilkan disertai dengan data transaksi penjualan	Halaman kelola transaksi penjualan ditampilkan disertai dengan data transaksi penjualan	Valid
		Produksi menekan tombol tambah pokok produksi, kemudian memasukkan data nama kebutuhan produksi, jenis kebutuhan produksi, dan biaya kebutuhan produksi.	Halaman form input harga pokok produksi ditampilkan dan data harga pokok produksi ditambahkan sesuai dengan inputan Produksi. Kemudian halaman harga pokok produksi yang telah diperbarui ditampilkan	Halaman form input harga pokok produksi ditampilkan dan data harga pokok produksi ditambahkan sesuai dengan inputan Produksi. Kemudian halaman harga pokok produksi yang telah diperbarui ditampilkan	Valid

Tabel 7.7 Validation Testing (lanjutan)

Kode	Fungsional Sistem	Kasus Uji	Yang diharapkan	Hasil yang didapat	Status
		Produksi menekan tombol edit pokok produksi, kemudian memasukkan data nama kebutuhan produksi, jenis kebutuhan produksi, dan biaya kebutuhan produksi yang ingin diubah.	Halaman form edit harga pokok produksi ditampilkan dan data harga pokok produksi diubah sesuai dengan inputan Produksi. Kemudian halaman kelola harga pokok produksi yang telah diperbarui ditampilkan.	Halaman form edit harga pokok produksi ditampilkan dan data harga pokok produksi diubah sesuai dengan inputan Produksi. Kemudian halaman kelola harga pokok produksi yang telah diperbarui ditampilkan.	Valid
		Produksi menekan tombol hapus pokok produksi pada salah satu data transaksi penjualan.	Data harga pokok produksi dihapus sesuai dengan pilihan Produksi dan halaman kelola harga pokok produksi yang telah diperbarui ditampilkan	Data harga pokok produksi dihapus sesuai dengan pilihan Produksi dan halaman kelola harga pokok produksi yang telah diperbarui ditampilkan	Valid
SKPL_SI_F_11	Lihat laporan pendukung SPT Tahunan	Pengguna memilih menu laporan pendukung SPT Tahunan	Halaman laporan pendukung SPT Tahunan ditampilkan	Halaman laporan pendukung SPT Tahunan ditampilkan	Valid
SKPL_SI_F_12	Cetak laporan pendukung SPT Tahunan	Owner memilih tombol cetak laporan. Kemudian melihat print preview dan memilih tombol cetak	Halaman print preview ditampilkan	Halaman print preview ditampilkan	Valid
SKPL_SI_F_13	Logout	Pengguna memilih tombol Logout.	Halaman login ditampilkan disertai dengan notifikasi “Terima kasih”	Halaman login ditampilkan disertai dengan notifikasi “Terima kasih”	Valid

Dapat dilihat pada pengujian diatas bahwa 13 fungsional sistem dan 23 kasus uji yang diuji dengan *validation testing* memiliki status yang valid. Sehingga dapat disimpulkan bahwa sistem informasi yang dibuat layak untuk digunakan dengan hasil 100% valid dalam pengujian *validation testing*.

7.2 Pengujian Non-Fungsional

Bab ini akan dijelaskan mengenai pengujian non-fungsional apa saja yang akan dilakukan untuk mengetahui layanan apa saja yang harus disediakan oleh sistem dan seberapa layak sistem dijalankan. Pengujian non-fungsional ini terdiri dari pengujian *Compatibility*, *User Acceptance*, dan Evaluasi Waktu Kinerja Sistem.

7.2.1 Pengujian *Compatibility*

Komputer yang berbeda, perangkat display, sistem operasi, browser, dan kecepatan koneksi jaringan dapat memiliki pengaruh signifikan pada operasi WebApp. Setiap konfigurasi komputasi dapat mengakibatkan perbedaan kecepatan pemrosesan sisi klien, resolusi layar, dan kecepatan koneksi. Sistem operasi yang rumit dapat menyebabkan masalah pemrosesan WebApp. Browser yang berbeda terkadang menghasilkan hasil yang sedikit berbeda, terlepas dari tingkat standarisasi HTML dalam WebApp. Plug-in yang diperlukan mungkin atau mungkin tidak tersedia untuk konfigurasi tertentu. Dalam beberapa kasus, masalah kompatibilitas kecil tidak menimbulkan masalah signifikan, namun di pihak lain, kesalahan serius dapat ditemui. Misalnya, kecepatan download mungkin tidak dapat diterima, kurangnya plug-in yang diperlukan dapat membuat konten tidak tersedia, perbedaan browser dapat mengubah tata letak halaman secara dramatis, gaya font dapat diubah. Dan menjadi tidak terbaca, atau formulir mungkin tidak berjalan dengan benar. Pengujian kompatibilitas berusaha untuk mengungkap masalah ini sebelum WebApp online.

Compatibility testing yang dilakukan pada sistem ini dilakukan berdasarkan pada sisi *browser*. *Compatibility testing* dilakukan dengan menggunakan alat SortSite versi 5.25.784.0. Alat ini akan melakukan analisis struktur *website* untuk mengetahui apakah terdapat bagian yang tidak kompatibel dengan *browser*. *Browser* yang digunakan SortSite dalam *compatibility testing* dapat dilihat pada Tabel 7.14:

Tabel 7.8 Browser Yang Digunakan Pada Compatibility Testing

No	Nama Browser	Versi Browser
1	Internet Explorer	8, 9, 10, dan 11
2	Microsoft Edge	15
3	Firefox	<=52 dan 53
4	Safari	<=9 dan 10
5	Opera	<=43 dan 44
6	Chrome	<=57 dan 58
7	iOS	<=8, 9, dan 10
8	Android	<=3 dan 4
9	BlackBerry	<=7.1 dan 10.0

Pada Gambar 7.4 menunjukkan hasil dari pengujian kompatibilitas yang memiliki 5 penjelasan yaitu sebagai berikut.

1. **Browser** berisi nama berbagai perambah web yang digunakan untuk pengujian.
2. **Version** adalah versi dari perambah web yang digunakan untuk pengujian.
3. **Critical issues** mengindikasikan sebuah konten atau fitur yang tidak didukung pada beberapa perambah web. Sehingga terdapat konten atau fungsi yang hilang pada beberapa perambah web.
4. **Major issues** mengindikasikan masalah utama pada tampilan atau performa pada beberapa perambah web.
5. **Minor issues** mengindikasikan masalah minor pada tampilan atau performa pada beberapa perambah web.

Browser	Internet Explorer				Edge	Firefox		Safari		Opera		Chrome		iOS			Android		BlackBerry	
Version	8	9	10	11	15	≤ 52	53	≤ 9	10	≤ 43	44	≤ 57	58	≤ 8	9	10	≤ 3	4*	≤ 7.1	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓
Minor Issues					✓												✓	✓	✓	✓

Gambar 7.4 Compatibility Testing dengan SortSite

Hasil dari pengujian kompabilitas pada Gambar 7.4 dapat membantu mengetahui apakah terdapat masalah dalam sistem jika digunakan di berbagai perambah web. Pada hasil pengujian kompabilitas di atas menunjukkan bahwa sistem daapt dijalankan pada 9 jenis perambah web yang berbeda.

7.2.3 User Acceptance Testing

Pada bagian ini akan menjelaskan mengenai *User Acceptance Testing* yang bertujuan untuk memeriksa sistem informasi yang telah dibuat terhadap kebutuhan bisnis yang telah didefinisikan pada bab analisis kebutuhan sehingga mendapatkan kepercayaan pengguna bahwa sistem informasi yang dibuat bebas dari kesalahan. Oleh karena itu sangat diperlukan pengujian *User Acceptance* pada sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan berbasis web pada Son Screen Printing Sidoarjo yang telah dibuat.

User Acceptance Testing dilakukan dengan cara membuat daftar pertanyaan yang mewakili seluruh kebutuhan yang telah didefinisikan pada bab analisis kebutuhan. Daftar pertanyaan akan ditanyakan saat pengguna mencoba sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan berbasis web pada Son Screen Printing Sidoarjo secara langsung sesuai dengan perannya agar pengisian dari daftar pertanyaan yang diberikan bernilai valid. Berikut ini merupakan tester sebagai perwakilan pengguna yang akan mengisi daftar pertanyaan ditunjukkan pada Tabel 7.9.

Setelah didapatkan tester yang akan melakukan simulasi dan menjawab daftar pertanyaan yang dibuat sesuai dengan perannya, maka selanjutnya dilakukan wawancara dan simulasi yang dilakukan oleh tester. Setelah analisa hasil wawancara dan hasil simulasi untuk menentukan tingkat penerimaan pengguna didapatkan maka akan diketahui apakah sistem informasi yang dibuat dapat diterima oleh pengguna atau sebaliknya.

Kriteria *User Acceptance Testing* terdiri dari *Functional Correctness and Completeness, Accuracy, Data Integrity, Data Conversion, Backup and Recovery, Competitive Edge, Usability, Performance, Start-Up Time, Reliability and Availability, Maintainability and Serviceability, Robustness, Timeliness, Confidentiality and Availability, Compatibility and Interoperability, Compliance, Installability and Upgradability, Scalability, dan Documentation* (Naik & Kshirasagar, 2008). Dari seluruh kriteria yang telah disebutkan pada pengujian sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan berbasis web pada Son Screen Printing Sidoarjo hanya menggunakan beberapa kriteria saja yang dapat dilihat pada Tabel 7.10.

Tabel 7.9 Analisa Tester User Acceptance Testing

No	Peran	Fungsi Sistem Yang Digunakan	Jumlah Populasi	Jumlah Tester (>=50%)
1.	Ower (Pemilik Perusahaan)	1. Login	1	1
		2. Lihat grafik penjualan		
		3. Menentukan harga pokok produksi		
		4. Lihat dan cetak laporan persediaan barang, transaksi penjualan, dan pendukung SPT tahunan.		
		5. Logout		
2.	Marketing (Bagian pemasaran)	1. Login	3	2
		2. Lihat grafik penjualan dan pendukung SPT tahunan		
		3. Menentukan harga pokok produksi		
		4. Lihat dan cetak laporan transaksi penjualan		
		5. Kelola transaksi penjualan (tambah, ubah, dan hapus)		
		6. Logout		
3.	Produksi (Bagian produksi barang)	1. Login	16	8
		2. Lihat grafik penjualan dan pendukung SPT tahunan		
		3. Menentukan harga pokok produksi		
		4. Lihat dan cetak laporan persediaan barang		
		5. Kelola persediaan barang dan harga pokok produksi (tambah, ubah, dan hapus)		
		6. Logout		
Total Tester				11

Tabel 7.10 Kriteria dan Pertanyaan *User Acceptance Testing*

No	Kriteria	Definisi (Naik & Kshirasagar, 2008)	Daftar Pertanyaan
1.	<i>Usability</i>	Evaluasi produk atau layanan dengan mengujinya dengan pengguna yang mewakili. Bertujuan untuk menentukan kepuasan pengguna terhadap produk.	1. Sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan mudah digunakan
			2. Tampilan sistem informasi persediaan barang dan harga pokok produksi, dan transaksi penjualan nyaman ketika digunakan
			3. Sistem informasi persediaan barang dan transaksi penjualan mempermudah penyampaian informasi antar pegawai dan pemilik perusahaan
			4. Penyaringan informasi persediaan barang dan harga pokok produksi dapat mempermudah dalam mencari persediaan barang tertentu
			5. Laporan persediaan barang mempermudah dalam melaporkan ke pemilik perusahaan
2.	<i>Functional Correctness and Completeness</i>	<i>Functional Correctness and Completeness</i> adalah pengujian untuk mengetahui apakah sistem yang dibuat sudah sesuai kebutuhan yang diinginkan pengguna dan telah didefinisikan pada spesifikasi kebutuhan	1. Laporan persediaan barang dan transaksi penjualan sesuai dengan kebutuhan.
			2. Sistem informasi persediaan barang dan transaksi penjualan dapat membantu dalam pembuatan laporan pendukung SPT Tahunan
			3. Perlu adanya laporan persediaan barang dan transaksi penjualan setiap bulan dan tahun untuk kemajuan perusahaan
			4. Grafik penjualan dapat membantu pegawai untuk mengetahui perkembangan perusahaan

Tabel 7.10 Kriteria dan Pertanyaan User Acceptance Testing (lanjutan)

No	Kriteria	Definisi (Naik & Kshirasagar, 2008)	Daftar Pertanyaan
			5. Sistem informasi harga pokok produksi sangat diperlukan perusahaan untuk pelayanan 6. Grafik penjualan dapat membangun kredibilitas yang tinggi antar pegawai 7. Laporan persediaan barang, transaksi penjualan, dan SPT tahunan diperlukan perusahaan 8. Cetak laporan diperlukan untuk arsip perusahaan sehingga dapat dimanfaatkan untuk masa depan
2.	<i>Performance</i>	Penentuan kecepatan atau efektivitas komputer, jaringan, program perangkat lunak atau perangkat keras.	1. Sistem informasi harga pokok produksi dapat mempercepat dalam menentukan harga pokok produksi
3.	<i>Confidentially and Availability</i>	<i>Confidentiality</i> adalah satu set aturan yang membatasi hak akses terhadap informasi. <i>Availability</i> adalah jaminan akses yang dapat dipercaya terhadap informasi oleh orang-orang yang berwenang	1. Diperlukan login user untuk keamanan informasi perusahaan

Setelah didapatkan kriteria dan pertanyaan yang sesuai dengan kriteria yang diharapkan. Tahap selanjutnya adalah dengan menganalisa hasil wawancara (data daftar pertanyaan yang telah dijawab oleh tester contoh pengisian

kuesioner dapat dilihat pada Lampiran A Pertanyaan dan Jawaban Kuesioner) yang ditunjukkan pada Tabel 7.11.

Tabel 7.11 Hasil Wawancara *User Acceptance Testing*

No	Kriteria	Jumlah Pertanyaan	Jumlah Setuju	Jumlah Tidak Setuju	Persentase yang setuju
1.	<i>Usability</i>	53	46	7	86,8%
2.	<i>Functional Correctness and Completeness</i>	97	87	10	89,7%
3.	<i>Performance</i>	11	10	1	90,9%
4.	<i>Confidentially and Availability</i>	11	10	1	90,9%
Rata-rata persentase yang setuju					89,6%

Dapat dilihat pada Tabel 7.11 bahwa pada kriteria *Usability* terdapat 7 tester yang tidak setuju dikarenakan menurut tester sistem informasi yang dibuat kurang mudah untuk dipahami dan terdapat tester yang belum biasa mengoperasikan laptop atau komputer. Pada kriteria *Functional Correctness and Completeness* terdapat 10 tester yang tidak setuju dikarenakan menurut beberapa tester lebih menyusahkan pegawai karena sebelumnya pegawai tidak perlu mencatat laporan persediaan barang maupun transaksi penjualan sehingga menambah pekerjaan dari pegawai. Pada kriteria *Performance* terdapat 1 tester yang tidak setuju dikarenakan terdapat 1 tester yang tidak pernah menggunakan atau mengoperasikan laptop atau komputer sehingga tester kesusahan dalam mengoperasikannya. Sedangkan pada kriteria *Confidentially and Availability* terdapat 1 tester yang tidak setuju dikarenakan menurut salah satu tester membuat keterbukaan dengan pegawai lain jadi kurang karena tidak semua pegawai dapat melihat informasi yang terdapat dalam sistem informasi yang dibuat. Untuk menghindari pengguna yang tidak menerima sistem informasi, perusahaan perlu mengadakan sosialisasi terkait penggunaan dari sistem informasi, kepentingan sistem informasi, dan keamanan informasi perusahaan yang terdapat pada sistem informasi yang dibuat, sebelum sistem informasi ini digunakan oleh perusahaan.

Namun, dari hasil wawancara yang telah dipetakan dalam Tabel 7.11, dapat dilihat bahwa 89,6% dari keseluruhan pertanyaan yang telah dijawab oleh 11 tester yang berhubungan langsung dengan sistem informasi yang dibuat tealh menerima sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan berbasis web pada Son Screen Printing Sidoarjo. Jadi dapat dikatakan bahwa sistem informasi ini layak digunakan oleh perusahaan untuk memaksimalkan kinerja perusahaan dalam menangani pelanggan.

7.2.4 Evaluasi Waktu Kinerja Sistem

Pada bagian ini akan menjelaskan mengenai evaluasi salah satu kinerja sistem pada sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan berbasis web pada Son Screen Printing Sidoarjo. Kinerja sistem yang dievaluasi adalah kinerja dari fungsi sistem untuk menghitung harga pokok produksi. Kemudian hasil kinerja sistem dibandingkan dengan kinerja perhitungan secara manual.

Evaluasi kinerja dari fungsi sistem untuk menghitung harga pokok produksi dihitung dengan cara observasi, wawancara, dan simulasi terhadap fungsi sistem yang dievaluasi. Observasi dan wawancara dilakukan ketika simulasi pada fungsi sistem untuk menghitung harga pokok produksi sedang dilakukan oleh tester. Kemudian hasil yang didapat dari observasi, wawancara, dan simulasi dianalisa untuk mendapatkan hasil dari evaluasi kinerja sistem yang bertujuan untuk mengetahui bahwa fungsi sistem untuk menghitung harga pokok produksi lebih cepat dibandingkan dengan kinerja perhitungan secara manual.

Untuk melakukan evaluasi kinerja sistem pada fungsi menghitung harga pokok produksi dibutuhkan tester pada masing-masing bagian yang berhubungan. Pada sistem informasi persediaan barang, harga pokok produksi, dan transaksi penjualan berbasis web pada Son Screen Printing Sidoarjo memiliki terdapat 3 aktor penting yang berhubungan langsung dengan sistem yaitu Owner (pemilik perusahaan), Marketing (yang menangani pelanggan), dan Produksi (yang memproduksi barang sesuai permintaan pelanggan). Perusahaan dipegang oleh 1 orang, untuk bagian pemasaran/marketing terdapat 3 pegawai, dan bagian produksi terdapat 16 orang yang terdiri dari 2 orang bagian desain, 7 orang bagian sablon, dan 7 orang bagian jahit. Jadi untuk melakukan evaluasi ini dibutuhkan tester minimal 50% pada setiap peran dalam perusahaan yang dapat dilihat pada Tabel 7.12

Tabel 7.12 Jumlah Tester Evaluasi Kinerja Sistem

No	Peran	Jumlah Populasi	Jumlah Tester ($\geq 50\%$)
1.	Ower (Pemilik Perusahaan)	1	1
2.	Marketing (Bagian pemasaran)	3	2
3.	Produksi (Bagian produksi barang)	16	8
Total Minimal Tester			11

Setelah dilakukan observasi, wawancara, dan simulasi didapatkan hasil wawancara yang ditunjukkan pada Tabel 7.13 sebagai berikut.

Tabel 7.13 Hasil Wawancara Kinerja Sistem

No	Peran	Jumlah Tester	Jumlah Tester Setuju	Persentase Tester Setuju
1.	Ower (Pemilik Perusahaan)	1	1	100%
2.	Marketing (Bagian pemasaran)	2	2	100%
3.	Produksi (Bagian produksi barang)	8	7	87,5%
Rata-rata Persentase Tester Setuju				95,8%

Dari hasil wawancara yang telah dipetakan dalam Tabel 7.20, dapat dilihat bahwa 95,8% dari keseluruhan pertanyaan yang telah dijawab oleh 11 tester yang berhubungan langsung dengan kinerja fungsi sistem menghitung harga pokok produksi menyatakan bahwa dengan menggunakan sistem informasi harga pokok produksi jauh lebih cepat daripada menghitung secara manual. Jadi dapat dikatakan bahwa sistem informasi harga pokok produksi lebih cepat dibandingkan dengan menghitung secara manual.

Hasil wawancara telah didapatkan dan untuk memperkuat dalam menjawab permasalahan terkait apakah kinerja fungsi sistem menghitung harga pokok produksi lebih cepat dibandingkan dengan menghitung secara manual yaitu dengan melakukan simulasi sistem informasi menghitung harga pokok produksi oleh tester yang bersangkutan dan melakukan perhitungan waktu jika dilakukan secara manual. Setelah didapatkan hasil perhitungan antara sistem dengan manual maka dapat dianalisis apakah kinerja fungsi sistem lebih cepat dibandingkan dengan menghitung secara manual. Berikut ini merupakan hasil simulasi yang telah dilakukan pengguna dan hasil menghitung secara manual yang dapat dilihat pada Tabel 7.14.

Tabel 7.14 Hasil Simulasi dan Menghitung Secara Manual

No	Peran	Waktu Menghitung Manual	Waktu Menghitung Sistem	Selisih (Manual - Sistem)
1.	Owner	192 detik	28 detik	164 detik
2.	Marketing 1	120 detik	40 detik	80 detik
3.	Marketing 2	118 detik	28 detik	90 detik
4.	Produksi 1	255 detik	55 detik	200 detik
5.	Produksi 2	242 detik	56 detik	186 detik
6.	Produksi 3	230 detik	112 detik	118 detik
7.	Produksi 4	168 detik	95 detik	73 detik
8.	Produksi 5	133 detik	37 detik	96 detik
9.	Produksi 6	207 detik	85 detik	122 detik
10.	Produksi 7	275 detik	66 detik	209 detik
11.	Produksi 8	127 detik	52 detik	75 detik
Rata-rata Selisih (Manual - Sistem)				128,45 detik

Dari hasil simulasi fungsi menghitung harga pokok produksi dan menghitung secara manual dapat dilihat bahwa menghitung dengan menggunakan sistem informasi harga pokok produksi lebih cepat daripada menghitung secara manual. Rata-rata selisih dari perhitungan antara menghitung dengan menggunakan sistem informasi dengan menghitung secara manual adalah 128,45 detik. Jadi, dapat disimpulkan bahwa kinerja fungsi sistem menghitung harga pokok produksi lebih cepat dibandingkan dengan menghitung secara manual dengan dukungan dari hasil wawancara dan simulasi yang dilakukan oleh tester.