

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1. Implementasi

Dalam pembuatan aplikasi ini tentunya diperlukan kebutuhan perangkat lunak dan perangkat keras. Maka dari itu, kebutuhan-kebutuhan tersebut di antaranya:

##### 1. Proses Penginstalan Aplikasi

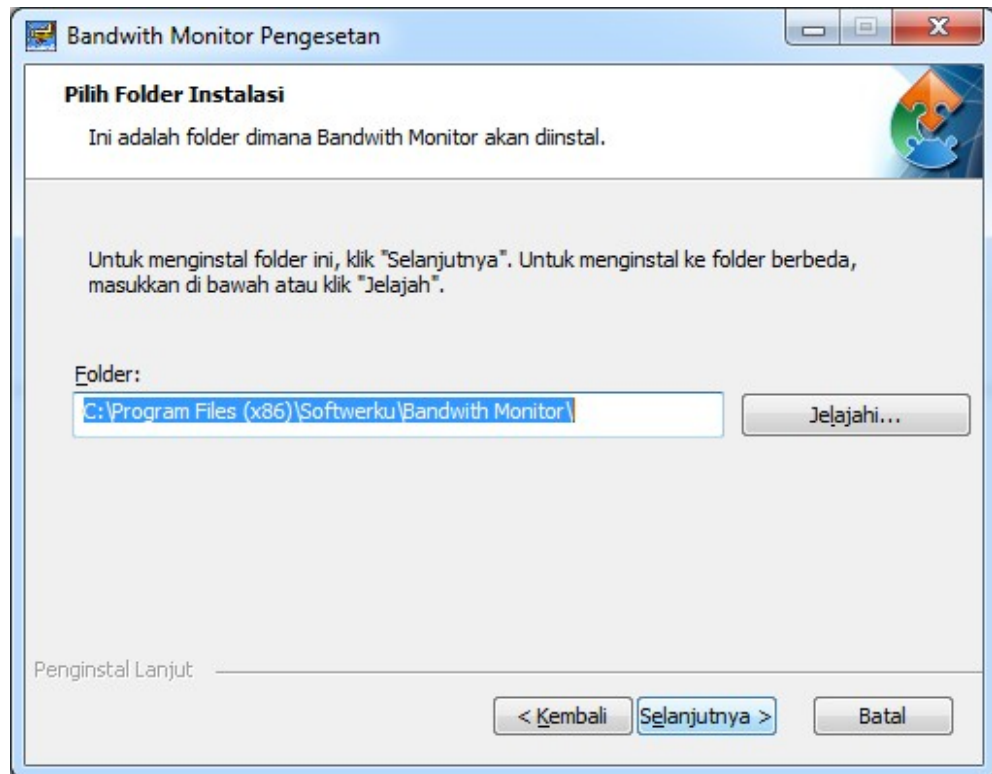
Klik dua kali *icon* aplikasi *bandwith monitor.exe*, akan muncul tampilan awal proses instalasi dan kemudian tekan tombol selanjutnya, seperti yang tampak pada gambar 8 berikut:



Gambar 8. Tampilan awal proses instalasi

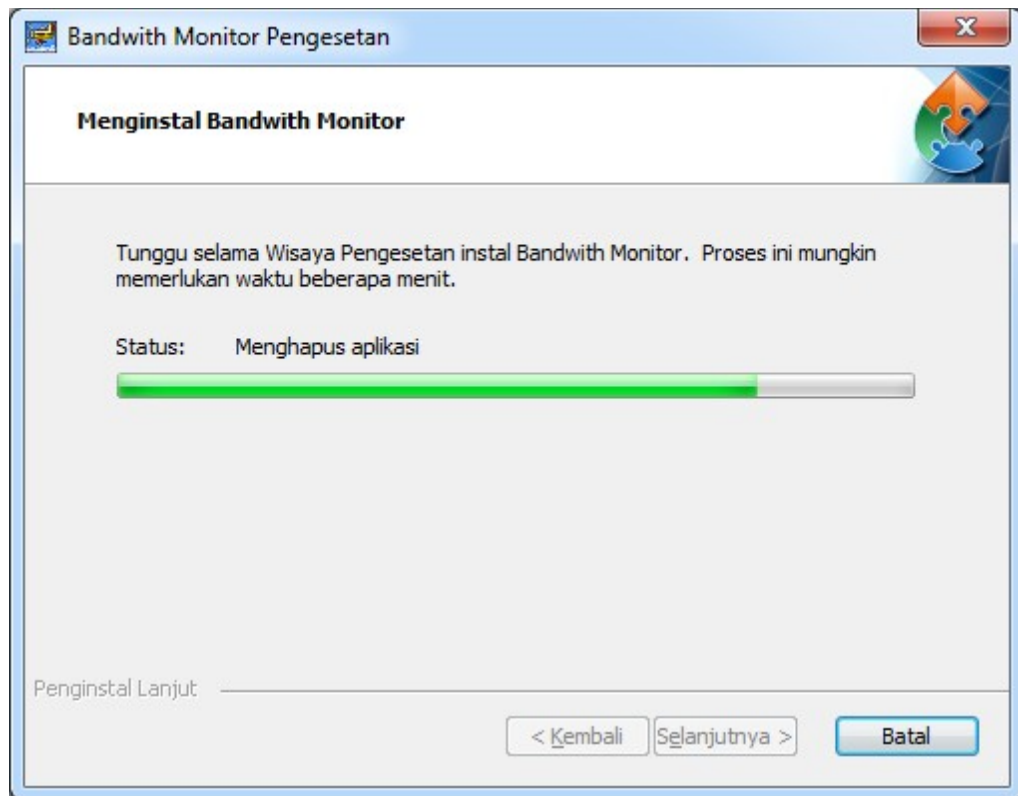
Tahapan berikutnya adalah memilih lokasi penginstalan aplikasi *bandwith monitor*. Secara default lokasi penginstalan berada pada drive *C:\Program Files (x86)\Softwareku\Bandwith Monitor*. Apabila ingin mengganti lokasi penginstalan maka tekan tombol jelajah dan kemudian tentukan lokasi penginstalan yang baru tapi jika tidak akan mengganti lokasi penginstalan,

tekan tombol selanjutnya. Seperti yang terlihat pada gambar 9 berikut:

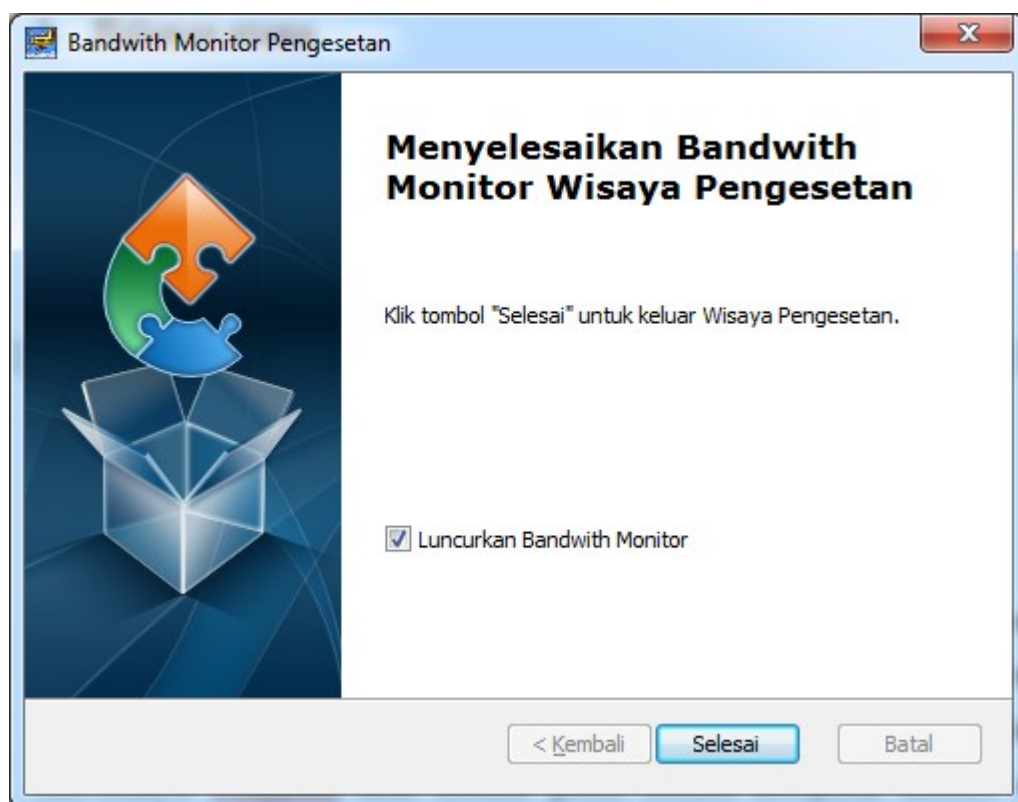


Gambar 9. Pemilihan folder penginstalan

Setelah menentukan folder penginstalan selanjutnya adalah proses penginstalan aplikasi ke folder yang sudah ditentukan sampai prosesnya selesai dan kemudian tekan tombol selesai dan secara otomatis program akan dijalankan untuk pertama kali, seperti yang terlihat pada gambar 10 dan 11 berikut:



Gambar 10. Proses pemasangan ke folder penginstalan



Gambar 11. Tahap akhir proses penginstalan

## 2. Halaman utama

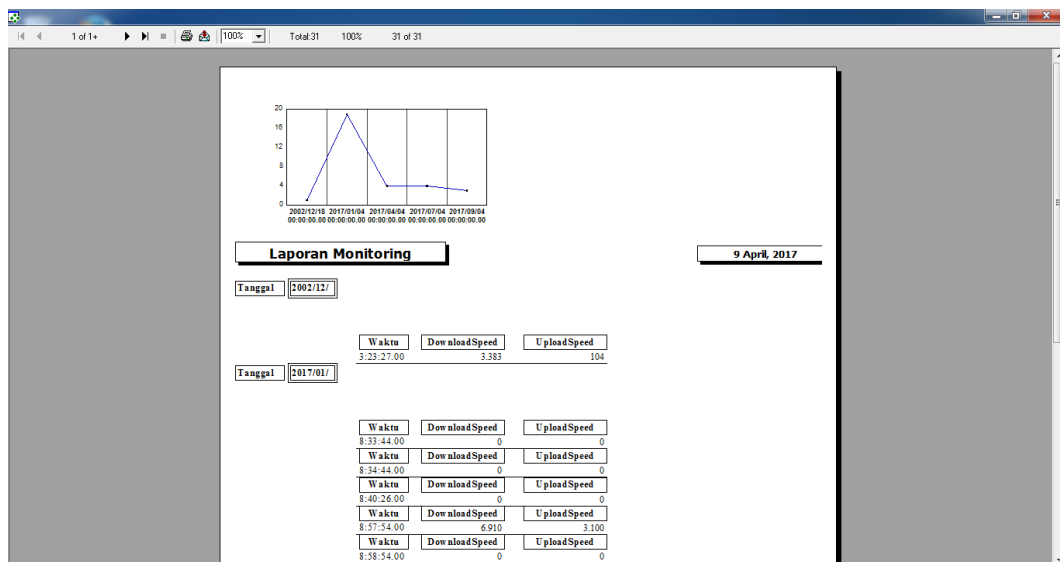
Halaman utama adalah halaman paling awal yang akan tampil dari program yang ingin dijalankan. Halaman menu utama menampilkan data kecepatan koneksi internet baik itu kecepatan download maupun kecepatan upload beserta rata-ratanya. Pengguna akan melihat grafik kondisi kecepatan internet secara simultan. Halaman utama dapat dilihat pada gambar 12 berikut:



Gambar 12. Halaman utama

### 3. Halaman laporan

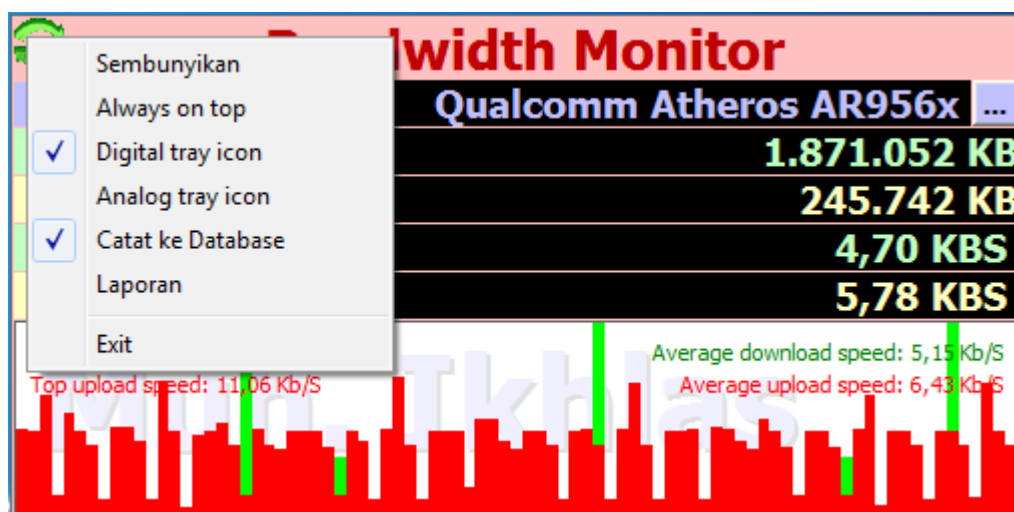
Halaman laporan adalah halaman yang dibuat untuk menampilkan hasil *monitoring bandwith* yang dicatat setiap interval 1 menit. Data yang dicatat adalah *download speed* dan *upload speed* setiap menit. Halaman laporan dapat dilihat pada gambar 13 berikut:



Gambar 13. Halaman Laporan

### 4. Halaman menu

Halaman menu adalah halaman yang digunakan untuk menampilkan menu aplikasi *monitoring bandwidth*. Halaman menu akan muncul ketika *icon* pada sudut kiri halaman utama diklik. Terdapat 7 menu yang bisa dipilih yaitu sembunyikan, *always on top*, *digital tray icon*, *analog tray icon*, catat ke database, laporan dan *exit*. Menu sembunyikan digunakan untuk meminimize halaman utama dan akan muncul pada ikon *tray*. Menu *always on top* berfungsi agar aplikasi *monitoring bandwidth* tetap tampil didepan layar tanpa tertutupi oleh aplikasi yang lain. Menu *digital tray icon* digunakan jika *tray icon* aplikasi *monitoring bandwidth* ingin diset ke bentuk digital. Menu *analog tray icon* fungsinya sama dengan menu *digital tray icon* perbedaannya hanya pada jenis ikonnya yaitu digital dan analog. Menu catat ke database digunakan untuk mencatat hasil *monitoring* ke dalam database. Menu laporan untuk menampilkan laporan hasil *monitoring*. Menu *exit* adalah menu untuk menutup aplikasi *monitoring bandwidth*. Halaman menu dapat dilihat pada gambar 14 berikut:



Gambar 14. Halaman registrasi

## 4.2. Pengujian Sistem

### 1. Pengujian basis path

Untuk menguji program aplikasi ini, penulis menggunakan pendekatan *White Box* yang merupakan salah satu metode pengujian yang menggunakan struktur kontrol design prosedur untuk mendapatkan kesalahan sebanyak-banyaknya dengan asumsi, setiap aplikasi yang dirancang pasti mempunyai

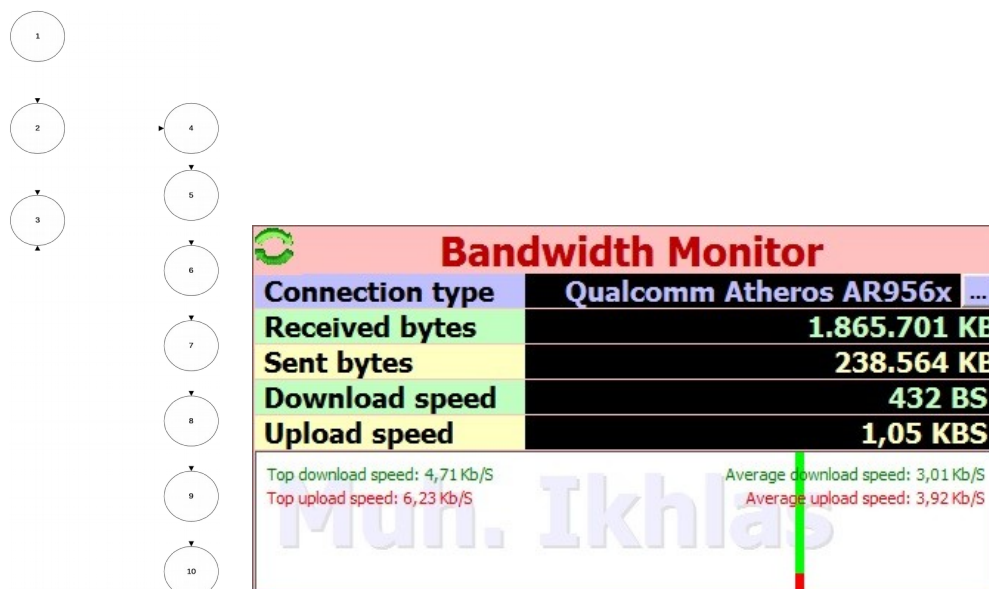
kesalahan sehingga dengan menemukan kesalahan sebanyak-banyaknya, dapat meningkatkan mutu atau kualitas dari aplikasi yang dirancang. Jadi, dengan menggunakan metode ini, perancang dapat mengetahui cara kerja dari aplikasi yang dirancang secara terperinci sesuai spesifikasinya dan menilai apakah setiap fungsi atau prosedur yang dirancang sudah berjalan dengan baik dan benar.

Sebagai alternatif pengembangan *white box testing*, penulis menggunakan metode *Basis Path Testing* yang berguna untuk:

1. Mengukur kompleksitas *logic* dari desain prosedur dan menggunakannya sekaligus sebagai pedoman untuk mendapatkan konsisten dari jalur aplikasi.
2. Pengujian yang dilakukan dijamin menggunakan *statement* dalam program minimal satu kali selama pengujian.
3. Menghitung *cyclometris complexity* sebagai ukuran kognitif untuk menentukan jumlah *independent path* sebagai jalur yang perlu diuji.

## 2. Flowgraph halaman utama

Dari *flowchart* halaman utama yang digunakan untuk pengujian perangkat lunak, maka dapat ditentukan *flowgraphnya* pada gambar 15 berikut:



Gambar 15. Flowgraph halaman utama

Perhitungan *Cyclomatic complexity* dari *flowgrap* memiliki region = 2

1. Menghitung *Cyclomatic complexity* dari *Edge* dan *Node*

Dengan Rumus :  $V(G) = E - N + 2$

Dimana :  $E$  ( jumlah *edge* pada *flowgraph* ) = 10

$N$  ( jumlah *node* pada *flowgraph* ) = 10

Penyelesaian :  $V(G) = 10 - 10 + 2$

$$V(G) = 2$$

Jadi jumlah *path* dari *flowgraph* di atas sebanyak 2 *path*.

2. Menghitung *Cyclomatic complexity* dari  $P$

$P$  adalah titik yang menyatakan logika dalam diagram alir dengan rumus  $V$

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

Penyelesaian :  $V(G) = 2 + 1$

$$V(G) = 3$$

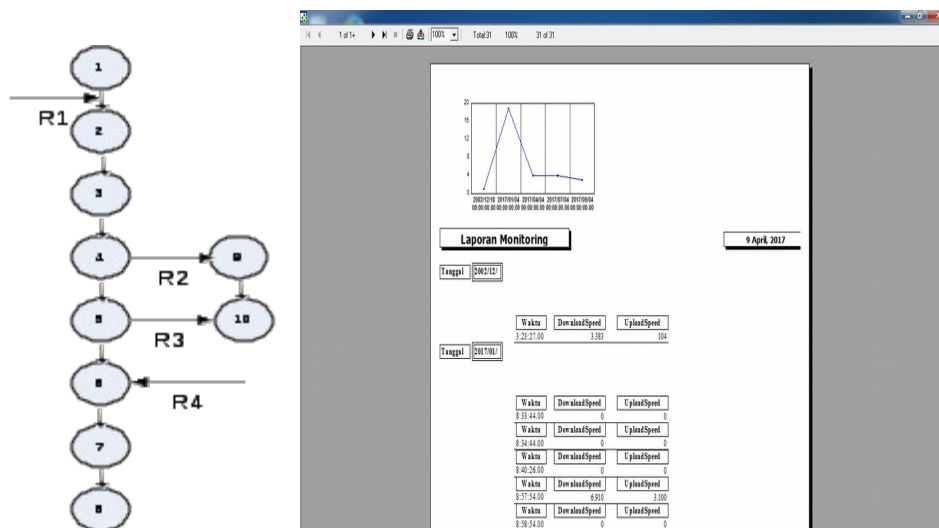
3. *Independent path* pada *flowgraph* di atas adalah :

*Path 1* = 1-2-3

*Path 2* = 1-2-4-5-6-7-8-9-10

3. **Flowgraph halaman laporan**

Dari *flowchart* halaman daftar pengguna yang digunakan untuk pengujian perangkat lunak, maka dapat ditentukan *flowgraphnya* pada gambar 16 berikut:



Gambar 16. Flowgraph halaman laporan

Perhitungan *Cyclomatic complexity* dari *flowgraph* memiliki region = 4

1. Menghitung *Cyclomatic complexity* dari *Edge* dan *Node*

Dengan Rumus :  $V(G) = E - N + 2$

Dimana :  $E$  ( jumlah *edge* pada *flowgraph* ) = 12

$N$  ( jumlah *node* pada *flowgraph* ) = 10

Penyelesaian :  $V(G) = 12 - 10 + 2$

$$V(G) = 4$$

Jadi jumlah *path* dari *flowgraph* di atas sebanyak 4 *path*.

## 2. Menghitung *Cyclomatic complexity* dari P

P adalah titik yang menyatakan logika dalam diagram alir dengan rumus  $V$

$(G) = P + 1$  dimana  $P = 3$

Penyelesaian :  $V(G) = 3 + 1$

$$V(G) = 4$$

## 3. *Independent path* pada *flowgraph* di atas adalah :

*Path 1* = 1-2-3-4-5-6-7-8

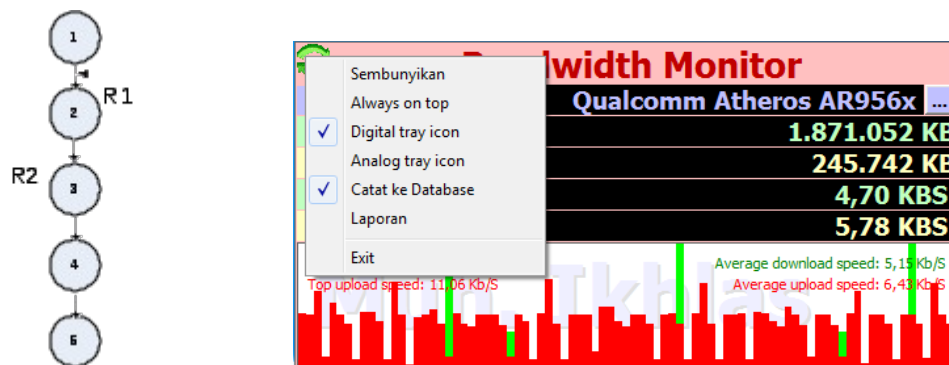
*Path 2* = 1-2-3-4-9-10-6-7-8

*Path 3* = 1-2-3-4-5-10-6-7-8

*Path 4* = 1-2-3-4-5-6-1

## 4. *Flowgraph* halaman menu

Dari *flowchart* halaman menu yang digunakan untuk pengujian perangkat lunak, maka dapat ditentukan *flowgraph*nya seperti pada gambar 17 berikut:



Gambar 17. *Flowgraph* halaman menu

Perhitungan *Cyclomatic complexity* dari *flowgrap* memiliki region = 2

## 1. Menghitung *Cyclomatic complexity* dari *Edge* dan *Node*

Dengan Rumus :  $V(G) = E - N + 2$

Dimana :  $E$  ( jumlah *edge* pada *flowgraph* ) = 5



$$N \text{ ( jumlah node pada flowgraph )} = 5$$

Penyelesaian :  $V(G) = 5 - 5 + 2$

$$V(G) = 2$$

Jadi jumlah *path* dari *flowgraph* di atas sebanyak 2 *path*.

2. Menghitung *Cyclomatic complexity* dari P

P adalah titik yang menyatakan logika dalam diagram alir dengan rumus  $V$

$$(G) = P + 1 \text{ dimana } P = 1$$

Penyelesaian :  $V(G) = 1 + 1$

$$(V(G) = 2$$

3. *Independent path* pada *flowgraph* di atas adalah :

$$\text{Path 1} = 1-2-3-4-5$$

$$\text{Path 2} = 1-2-3-1$$

Berdasarkan hasil pengujian program diperoleh hasil dimana jumlah Region = 13, *Cyclomatic Complexity* (CC) = 13 dan Independent Path = 13, maka hal ini menandakan bahwa program sudah bebas dari kesalahan logika. Hasil pengujian secara keseluruhan dapat dilihat pada tabel 7 berikut.

Tabel 7. Hasil pengujian program

No	Nama Bagan Alir Program	Jumlah	Independent	Jumlah
		CC	Path	Region
1	Halaman utama	2	2	2
2	Halaman laporan	4	4	4
3	Halaman menu	2	2	2
<b>Total</b>		<b>8</b>	<b>8</b>	<b>8</b>