

UJIAN AKHIR SEMESTER
PRAKTIKUM PENGUJIAN PERANGKAT LUNAK



Disusun Oleh :

Shindi Purnama Putri (151811513009)

D3 SISTEM INFORMASI
FAKULTAS VOKASI
UNIVERSITAS AIRLANGGA

1. Pengantar White Box Testing

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode program ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya White Box Testing ini menguji dengan cara melihat Pure Code dari suatu aplikasi/software yang diuji tanpa memperdulikan Tampilan atau UI dari aplikasi tersebut..

Teknik White-box Testing

- Basis Path Testing

Basis path testing merupakan metode yang memungkinkan perancang testcase untuk membuat pengukuran kompleksitas logikal dari rancangan prosedural dan menggunakan pengukuran ini sebagai panduan untuk mendefinisikan himpunan basis dari jalur eksekusi. Test case yang dibuat untuk menguji himpunan basis dijamin akan mengeksekusi setiap statement di dalam program sekurangnya sekali pada saat pengujian

- Flow Graph

Flow graph merupakan notasi sederhana untuk merepresentasi control flow.

- Cyclomatic Complexity

Cyclomatic complexity digunakan untuk mengetahui jumlah jalur yang perlu dicari. Cyclomatic complexity adalah metric software yang menyediakan ukuran kuantitatif dari kompleksitas logikal program. Nilai yang dihitung bagi cyclomatic complexity menentukan jumlah jalur-jalur yang independen dalam kumpulan basis suatu program dan memberikan jumlah tes minimal yang harus dilakukan untuk memastikan bahwa semua pernyataan telah dieksekusi sekurangnya satu kali.

Cyclomatic complexity mempunyai fondasi dalam teori graph dan dapat dihitung dengan satu dari tiga cara :

1. Jumlah region sama dengan cyclomatic complexity.
2. Cyclomatic complexity, $V(G)$, untuk sebuah flow graph, G , didefinisikan sebagai: $V(G) = E - N + 2$ E adalah jumlah edge pada flow graph, dan N adalah jumlah node pada flow graph.
3. Cyclomatic complexity, $V(G)$, untuk flow graph, G , juga didefinisikan sebagai: $V(G) = P + 1$ P adalah jumlah predicate nodes yang terdapat pada flow graph G .

- Graph Matrix

Prosedur untuk membuat flow graph dan menentukan himpunan basis path dapat diterima berdasarkan mekanisme. Untuk mengembangkan software yang membantu pengujian basis path, sebuah struktur data yang disebut graph matrix, dapat sangat bermanfaat. Graph matrix adalah matriks kotak yang ukurannya (jumlah baris dan kolom) sama untuk jumlah node pada flow graph. Setiap baris dan kolom berhubungan dengan node yang teridentifikasi, dan data matriks berhubungan dengan koneksi (edge) antara

Kelemahan White Box-Testing:

1. Sangat mahal untuk dilakukan karena membutuhkan tester yang terampil untuk melakukan pengujian.
2. Pada perangkat lunak yang jenisnya besar, metode white box testing ini dianggap boros karena melibatkan banyak sumberdaya untuk melakukannya.
3. Tidak mempedulikan Tampilan UI aplikasinya.

2. Gambaran Aplikasi Aset Kantor

Sistem informasi aset kantor adalah aplikasi inventaris yang digunakan untuk mencatat barang aset kantor dan sejarah perawatannya.

Aplikasi ini memiliki 2 peran yaitu:

- Admin
 - hanya berhak membuat, menghapus, memodifikasi data pengguna
 - Tidak berhak memodifikasi data yang lain seperti data barang dan data perawatan.
- Operator

Hanya berhak mengelola data barang, perawatan, dan mengganti sandi milik sendiri

3. Ringkasan Kasus Uji

Dalam pengujian Sistem informasi Aset Kantor ini saya melakukan pengujian dengan menggunakan jenis whitebox :

1. Basis Path
2. Complexity Cyclometric
3. Graph Matrix

Tabel Ringkasan Skenario Pengujian

Test Case/Script Identifier	Test Case/Script Title	Execution Priority
Test Case 1	edit_password.php	2
Test Case 2	hapus_perawatan.php	6
Test Case 3	login.php	4
Test Case 4	simpan_edit_pengguna.php	3
Test Case 5	simpan_pengguna.php	1
Test Case 6	simpan_perawatan.php	5

4. Rincian Kasus Uji

Berikut adalah spesifikasi setiap tes skenario dalam pengujian perangkat lunak sistem informasi Aset Kantor.

4.1. Test Case 1

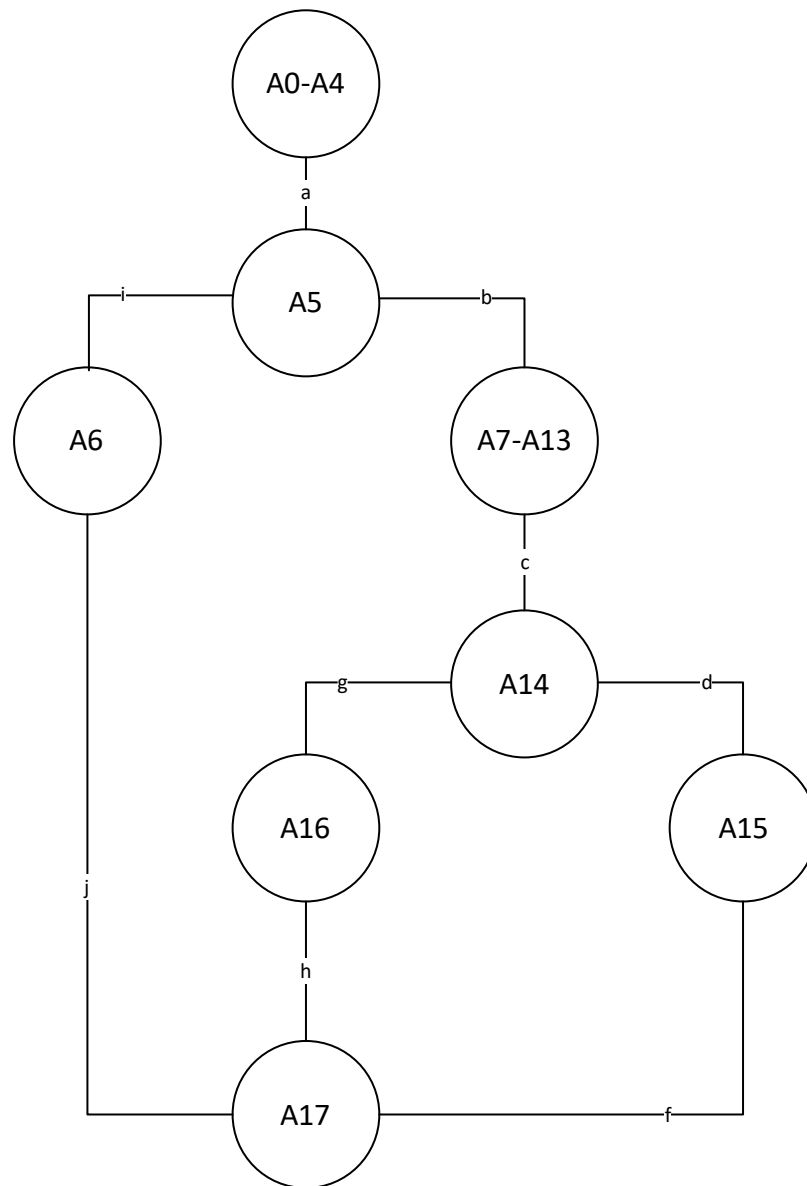
Pengujian : edit_password.php

Jenis Pengujian : Basis path, Complexity Cyclometric dan Graph Matrix.

A0	<?php
A1	session_start();
A2	\$id_pengguna = \$_SESSION['id_pengguna'];
A3	\$sandi = \$_POST['sandi'];
A4	\$conn = new mysqli('localhost', 'root', '', 'aset_kantor');
A5	if (\$conn->connect_error) {
A6	die("Connection failed: " . \$conn->connect_error);
A7	}
A8	\$sql = "
A9	UPDATE `pengguna`
A10	SET `sandi`='\$sandi'
A11	WHERE `id_pengguna`='\$id_pengguna'
A12	";
A13	\$result = \$conn->query(\$sql);
A14	if(\$result){
A15	echo "Password Berhasil Diupdate";
A16	}
A17	?>

Tabel 1 : Code edit_password.php

4.1.1. Basis Path



Gambar 1 : *Basis Path* edit_password.php

4.1.2. Complexity Cyclometric

$$V(G) \rightarrow Region = V(3)$$

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

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

$$V(G) = 3$$

4.1.3. Independent Path

1. A0-A4 – A5 – A7 – A8 – A9 – A10 – A11 – A12 – A13 – A14 – A16 – A17

2. A0-A4 – A5 – A6 – A7 – A8 – A9 – A10 – A11 – A12 – A13 – A14 – A16 – A17
3. A0-A4 – A5 – A6 – A7 – A8 – A9 – A10 – A11 – A12 – A13 – A14 – A15 – A16 – A17

4.1.4. Test Case

- I.
 1. A(5) = False
 2. A(14) = False
- II.
 1. A(5) = True
 2. A(14) = False
- III.
 1. A(5) = False
 2. A(14) = True
- IV.
 1. A(5) = True
 2. A(14) = True

4.1.5. Graph Matrix

4.1.5.1. Tabel Graph Matrix

To From	A0-A4	A5	A6	A7- A13	A14	A15	A16	A17
A0- A4		a						
A5			i	b				
A6								j
A7- A13					c			
A14						g	d	
A15								h
A16								f
A17								

Tabel 1.1 : Graph Matrix

4.1.5.2. Tabel Conection Matrix

To From	A0-A4	A5	A6	A7-A13	A14	A15	A16	A17	connection
A0- A4		1							1-1 = 0
A5			1	1					2-1 = 1
A6								1	1-1 = 0
A7- A13					1				1-1 = 0
A14						1	1		2-1 = 1
A15								1	1-1 = 0
A16								1	1-1 = 0
A17									0
TOTAL CONNECTION									2

Tabel 1.1 : Conection Graph Matrix

4.2.Test Case 2

Pengujian : Hapus_perawatan.php

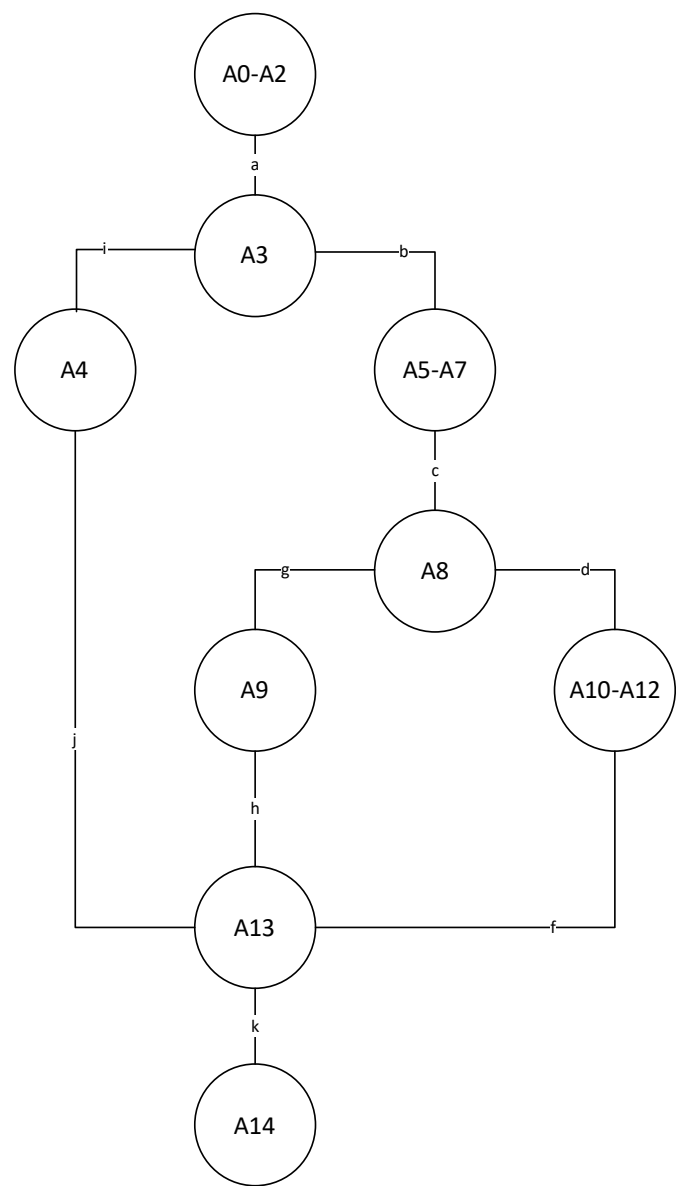
Jenis Pengujian : Basis path, Complexity Cyclometric dan Graph Matrix.

A0	<?php
A1	\$id_perawatan = \$_GET['id_perawatan'];
A2	\$conn = new mysqli('localhost', 'root', '', 'aset_kantor');
A3	if (\$conn->connect_error) {
A4	die("Connection failed: " . \$conn->connect
A5	}
A6	\$sql = "DELETE FROM `perawatan` WHERE `id_perawatan`='\$id_perawatan'";
A7	\$result = \$conn->query(\$sql);
A8	if(\$result){
A9	echo "Hapus data Berhasil";

A10	}
A11	else {
A12	echo "Hapus Data Gagal";
A13	}
A14	?>

Tabel 2 : *Code* Hapus_password.php

4.2.1. Basis Path



Gambar 2 : *Basis Path* hapus_password.php

4.2.2. Complexity Cyclometric

$$V(G) \rightarrow Region = V(3)$$

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

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

$$V(G) = 3$$

4.2.3. Independent Path

1. A0-A2 – A3 – A6 – A7 – A8 – A13 – A14
2. A0-A2 – A3 – A4 – A5 – A6 – A7 – A8 – A13 – A14
3. A0-A2 – A3 – A4 – A5 – A6 – A7 – A8 – A9 – A10 – A11 – A12 – A13 – A14

4.2.4. Test Case

I.

1. A(3) = False
2. A(8) = False

II.

1. A(3) = False
2. A(8) = True

III.

1. A(3) = True
2. A(8) = False

IV.

1. A(3) = True
2. A(8) = True

4.2.5. Graph Matrix

4.2.5.1. Tabel Graph Matrix

To From	A0-A2	A3	A4	A5- A7	A8	A9	A10- A12	A13	A14
A0- A2		a							
A3			i	b					
A4								j	
A5- A7					c				
A8						g	d		
A9								h	

A10- A12								f	
A13									k
A14									

Tabel 1.2 : Graph Matrix

4.2.5.2. Tabel Conection Matrix

To From	A0-A2	A3	A4	A5- A7	A8	A9	A10- A12	A13	A14	Connection
A0- A2		1								1-1 = 0
A3			1	1						2-1 = 1
A4								1		1-1 = 0
A5- A7					1					1-1 = 0
A8						1	1			2-1 = 1
A9								1		1-1 = 0
A10- A12								1		1-1 = 0
A13									1	1-1 = 0
A14										0
Total Connection										2

Tabel 1.2 : Conection Graph Matrix

4.3. Test Case 3

Pengujian : login.php

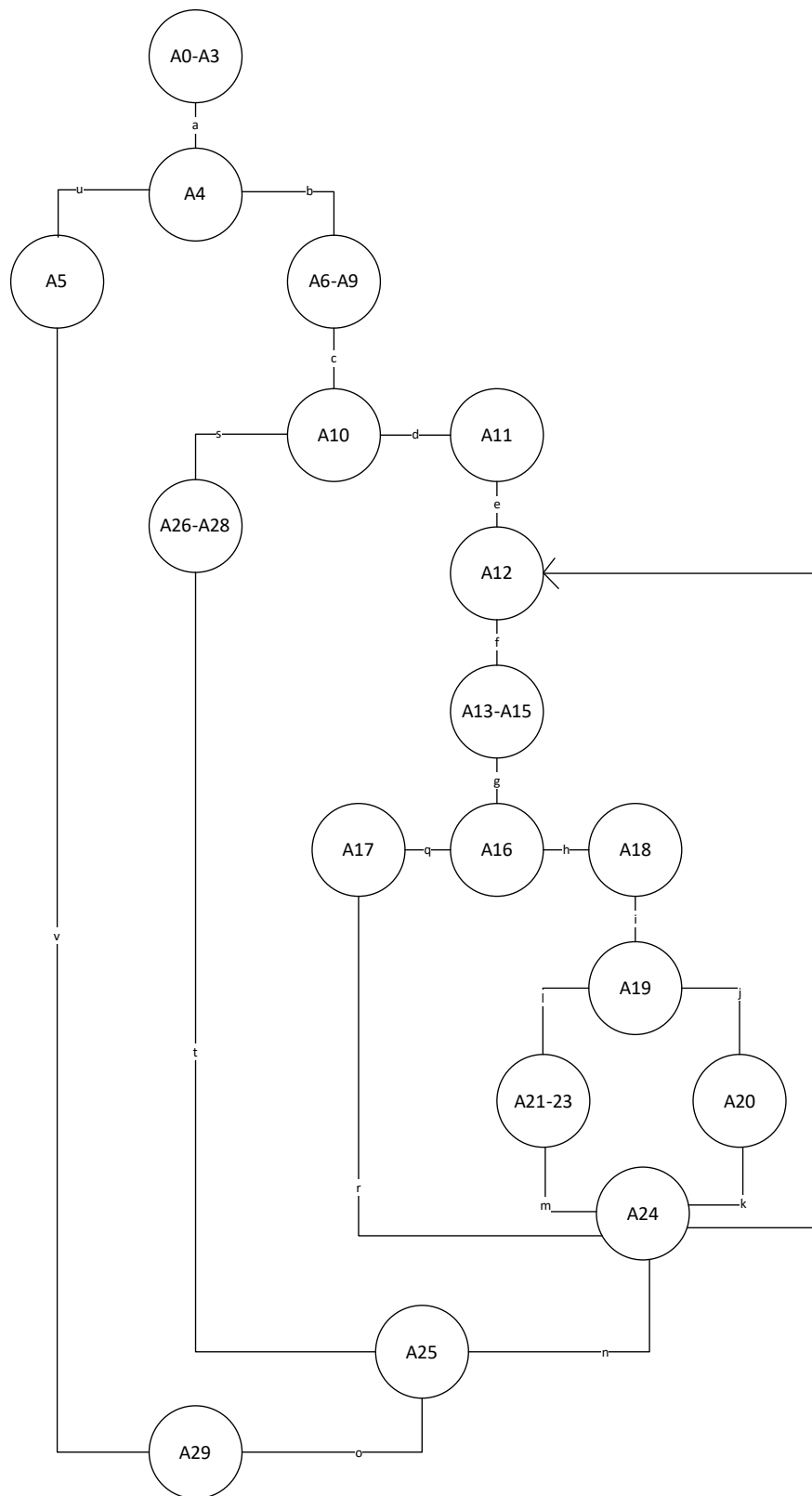
Jenis Pengujian : Basis path, Complexity Cyclometric dan Graph Matrix.

A0	<?php
A1	\$username = \$_POST['username'];
A2	\$password = \$_POST['password'];
A3	\$conn = new mysqli('localhost', 'root', '', 'aset_kantor');
A4	if (\$conn->connect_error) {
A5	die("Connection failed: " . \$conn->connect_error);

A6	}
A7	<code>\$sql = "SELECT `id_pengguna`, `login`, `sandi`, `peran` FROM `pengguna` WHERE `login`='\$username' and `sandi`='\$password' ";</code>
A8	<code>\$result = \$conn->query(\$sql);</code>
A9	<code>\$akses =null;</code>
A10	<code>if (\$result->num_rows > 0) {</code>
A11	<code> session_start();</code>
A12	<code> while(\$row = \$result->fetch_assoc()) {</code>
A13	<code> \$_SESSION['id_pengguna'] = \$row['id_pengguna'];</code>
A14	<code> \$_SESSION['peran'] = \$row["peran"];</code>
A15	<code> \$_SESSION['login'] = \$row["login"];</code>
A16	<code> if(\$row["peran"]=='0'){</code>
A17	<code> header('Location: v_halaman_admin.php');</code>
A18	<code> }</code>
A19	<code> else if(\$row["peran"]=='1'){</code>
A20	<code> header('Location: v_halaman_operator.php');</code>
A21	<code> }</code>
A22	<code> } else {</code>
A23	<code> echo "username salah";</code>
A24	<code> }</code>
A25	<code>}</code>
A26	<code>} else {</code>
A27	<code> echo "0 results";</code>
A28	<code>}</code>
A29	<code>?></code>

Tabel 3 : Code login.php

4.3.1. Basis Path



Gambar 3 : *Basis Path* login.php

4.3.2. Complexity Cyclometric

$$V(G) \rightarrow Region = V(6)$$

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

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

$$V(G) = 6$$

4.3.3. Independent Path

1. A0-A3 – A4 – A7 – A8 – A9 – A29
2. A0-A3 – A4 – A5 – A6 – A7 – A8 – A9 – A28 – A29
3. A0-A3 – A4 – A5 – A6 – A7 – A8 – A9 – A10 – A11 – A12 – A26 – A27 – A28 – A29
4. A0-A3 – A4 – A5 – A6 – A7 – A8 – A9 – A10 – A11 – A12 – A13 – A14 – A15 – A16 – A26 – A27 – A28 – A29
5. A0-A3 – A4 – A5 – A6 – A7 – A8 – A9 – A10 – A11 – A12 – A13 – A14 – A15 – A16 – A17 – A18 – A19 – A26 – A27 – A28 – A29
6. A0-A3 – A4 – A5 – A6 – A7 – A8 – A9 – A10 – A11 – A12 – A13 – A14 – A15 – A16 – A17 – A18 – A19 – A20 – A21 – A22 – A23 – A24 – A25 – A26 – A27 – A28 – A29

4.3.4. Test Case

I.

1. A(4) = False
2. A(9) = False
3. A(12) = False
4. A(16) = False
5. A(19) = False

II.

1. A(4) = True
2. A(9) = False
3. A(12) = False
4. A(16) = False
5. A(19) = False

III.

1. A(4) = True
2. A(9) = True
3. A(12) = False
4. A(16) = False
5. A(19) = False

IV.

1. A(4) = True
2. A(9) = True

3. $A(12) = \text{True}$
4. $A(16) = \text{False}$
5. $A(19) = \text{False}$

V.

1. $A(4) = \text{True}$
2. $A(9) = \text{True}$
3. $A(12) = \text{True}$
4. $A(16) = \text{True}$
5. $A(19) = \text{False}$

VI.

1. $A(4) = \text{True}$
2. $A(9) = \text{True}$
3. $A(12) = \text{True}$
4. $A(16) = \text{True}$
5. $A(19) = \text{True}$

VII.

1. $A(4) = \text{True}$
2. $A(9) = \text{True}$
3. $A(12) = \text{True}$
4. $A(16) = \text{True}$
5. $A(19) = \text{False}$

VIII.

1. $A(4) = \text{True}$
2. $A(9) = \text{True}$
3. $A(12) = \text{True}$
4. $A(16) = \text{False}$
5. $A(19) = \text{False}$

IX.

1. $A(4) = \text{True}$
2. $A(9) = \text{True}$
3. $A(12) = \text{False}$
4. $A(16) = \text{False}$
5. $A(19) = \text{False}$

X.

1. $A(4) = \text{True}$
2. $A(9) = \text{False}$
3. $A(12) = \text{False}$
4. $A(16) = \text{False}$
5. $A(19) = \text{Fa}$

4.3.5. Graph Matrix

4.3.5.1. Tabel Graph Matrix

[illegible]

A26- A28																	t	
A29																		

Tabel 3.1 : Graph Matrix

4.3.5.2. Tabel Conection Matrix

To From	A0- A3	A4	A5	A6- A9	A10	A11	A12	A13- A15	A16	A17	A18	A19	A20	A21- A23	A24	A25	A26- A28	A29	Connection
A0- A3		1																	1-1 = 0
A4			1	1															2-1 = 1
A5																		1	1-1 = 0
A6- A9					1														1-1 = 0
A10						1											1		2-1 = 1
A11							1												1-1 = 0
A12								1											1-1 = 0
A13- A15									1										1-1 = 0
A16										1	1								2-1 = 1
A17															1				1-1 = 0
A18												1							1-1 = 0
A19													1	1					2-1 = 1
A20															1				1-1 = 0

A21- A23															1				1-1 = 0
A24							1									1			2-1 = 1
A25																		1	1-1 = 0
A26- A28																	1		1-1 = 0
A29																			0
Total Connection																			5

Tabel 3.2 : Conection Graph Matrix

4.4. Test Case 4

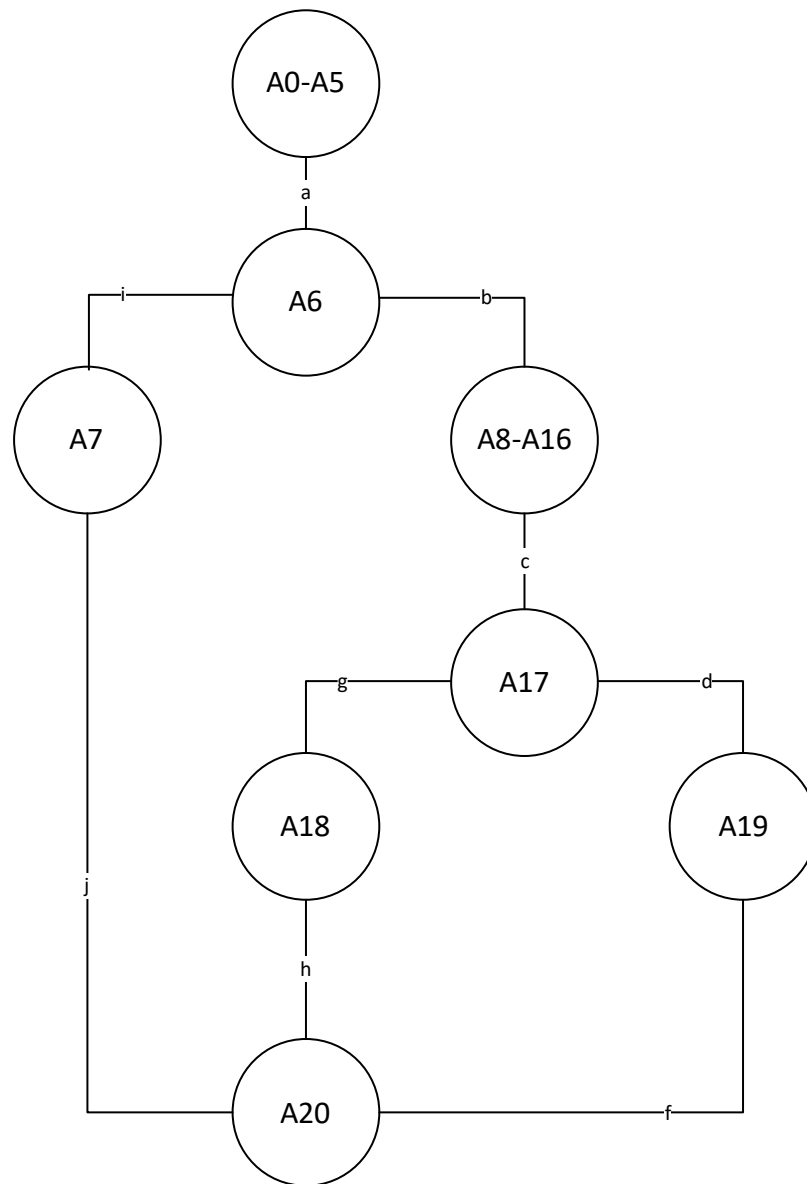
Pengujian : `simpan_edit_pengguna.php`

Jenis Pengujian : Basis path, Complexity Cyclometric dan Graph Matrix.

A0	<code><?php</code>
A1	<code>\$id_pengguna = \$_POST['id_pengguna'];</code>
A2	<code>\$login = \$_POST['login'];</code>
A3	<code>\$sandi = \$_POST['sandi'];</code>
A4	<code>\$optradio = \$_POST['optradio'];</code>
A5	<code>\$conn = new mysqli('localhost', 'root', '', 'aset_kantor');</code>
A6	<code>if (\$conn->connect_error) {</code>
A7	<code>die("Connection failed: " . \$conn->connect_error);</code>
A8	<code>}</code>
A9	<code>\$sql = "</code>
A10	<code>UPDATE `pengguna`</code>
A11	<code>SET `login`='\$login',</code>
A12	<code>`sandi`='\$sandi',</code>
A13	<code>`peran`='\$optradio'</code>
A14	<code>WHERE `id_pengguna`='\$id_pengguna'</code>
A15	<code>";</code>
A16	<code>\$result = \$conn->query(\$sql);</code>
A17	<code>if(\$result){</code>
A18	<code>echo "Data Berhasil Diubah";</code>
A19	<code>}</code>
A20	<code>?></code>

Tabel 4 : *Code* `simpan_edit_pengguna.php`

4.4.1. Basis Path



Gambar 4 : *Basis Path* `simpan_edit_pengguna.php`

4.4.2. Complexity Cyclometric

$$V(G) \rightarrow Region = V(3)$$

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

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

$$V(G) = 3$$

4.4.3. Independent Path

1. A0-A5 – A6 – A9-A16 – A17 – A19 – A20
2. A0-A5 – A6 – A7 – A8 – A9-A16 – A17 – A19 – A20

3. A0-A5 – A6 – A7 – A8 – A9-A16 – A17 – A18 – A19 – A20

4.4.4. Test Case

I.

1. A(6) = False
2. A(17) = False

II.

1. A(6) = False
2. A(17) = True

III.

1. A(6) = True
2. A(17) = False

IV.

1. A(6) = True
2. A(17) = True

4.4.5. Graph Matrix

4.4.5.1. Tabel Graph Matrix

To From	A0-A5	A6	A7	A8- A16	A17	A18	A19	A20
A0- A5		a						
A6			i	b				
A7								j
A8- A16					c			
A17						g	d	
A18								h
A19								f
A20								

Tabel 4.1 : Graph Matrix

4.4.5.2. Tabel Conection Matrix

To From	A0-A5	A6	A7	A8- A16	A17	A18	A19	A20	Connection
A0- A5		1							1-1 = 0

A6			1	1					2-1 = 1
A7								1	1-1 = 0
A8- A16					1				1-1 = 0
A17						1	1		2-1 = 1
A18								1	1-1 = 0
A19								1	1-1 = 0
A20									0
Total Connection									2

Tabel 4.1 : Conection Graph Matrix

4.5. Test Case 5

Pengujian : simpan_pengguna.php

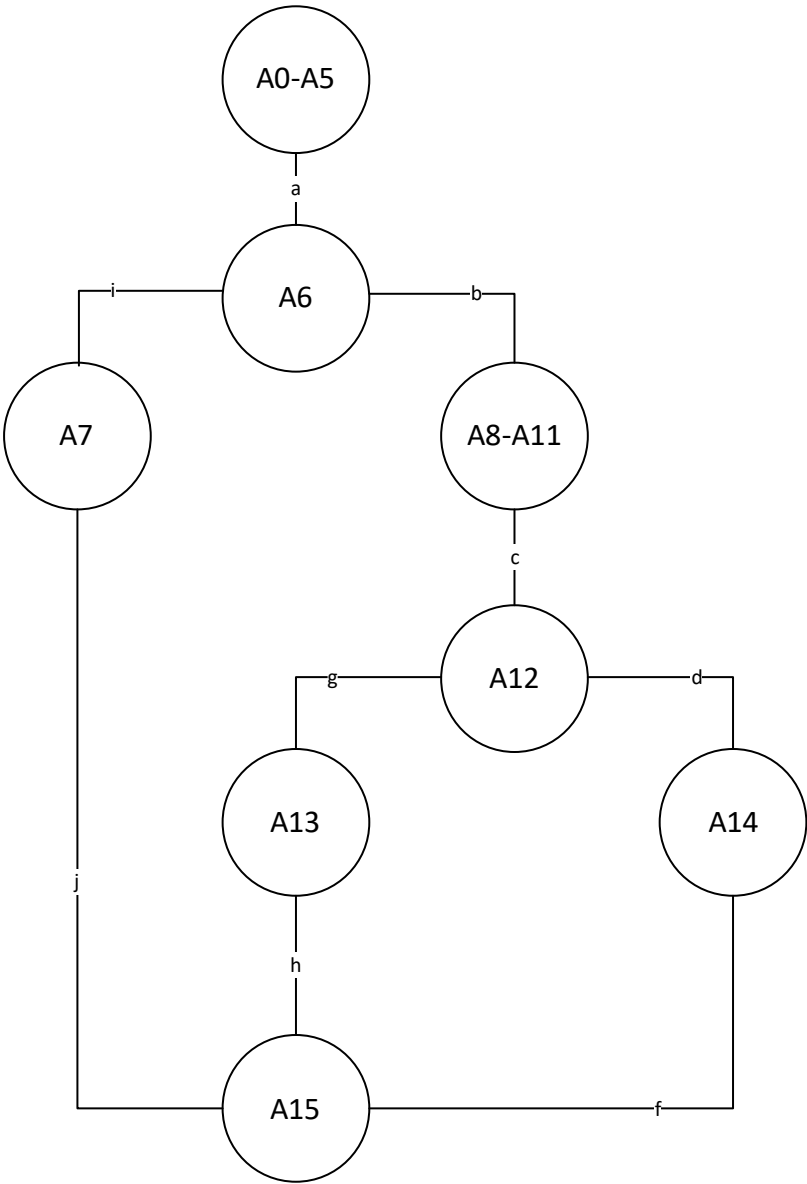
Jenis Pengujian : Basis path, Complexity Cyclometric dan Graph Matrix.

A0	<?php
A1	\$id_pengguna = \$_POST['id_pengguna'];
A2	\$login = \$_POST['login'];
A3	\$sandi = \$_POST['sandi'];
A4	\$optradio = \$_POST['optradio'];
A5	\$conn = new mysqli('localhost', 'root', '', 'aset_kantor');
A6	if (\$conn->connect_error) {
A7	die("Connection failed: " . \$conn->connect_error);
A8	}
A9	\$sql = "INSERT INTO `pengguna`(`id_pengguna`, `login`, `sandi`, `peran`)
A10	VALUES ('\$id_pengguna','\$login','\$sandi','\$optradio') ";
A11	\$result = \$conn->query(\$sql);
A12	if(\$result){
A13	echo "Data Berhasil Ditambah";

A14	}
A15	?>

Tabel 5 : Code simpan_pengguna.php

4.5.1. Basis Path



Gambar 5 : Basis Path simpan _pengguna.php

4.5.2. Complexity Cyclometric

$$\begin{aligned}
 V(G) &\rightarrow Region = V(3) \\
 V(G) &= P + 1 \\
 V(G) &= 2 + 1 \\
 V(G) &= 3
 \end{aligned}$$

4.5.3. Independent Path

1. A0-A5 – A6 – A9-A11 – A12 – A15
2. A0-A5 – A6 – A7 – A8 – A9-A11– A12 – A15
3. A0-A5 – A6 – A7 – A8 – A9-A11– A12 – A13 – A14 – A15

4.5.4. Test Case

I.

1. A(6) = False
2. A(12) = False

II.

1. A(6) = False
2. A(12) = True

III.

1. A(6) = True
2. A(12) = False

IV.

1. A(6) = True
2. A(12) = True

4.5.5. Graph Matrix

4.5.5.1. Tabel Graph Matrix

To From	A0-A5	A6	A7	A8- A11	A12	A13	A14	A15
A0- A5		a						
A6			i	b				
A7								j
A8- A11					c			
A12						g	d	
A13								h
A14								f
A15								

Tabel 5.1 : Graph Matrix

4.5.5.2. Tabel Conection Matrix

To From	A0-A5	A6	A7	A8- A11	A12	A13	A14	A15	Connection
A0- A5		1							1-1 = 0
A6			1	1					2-1 = 1
A7								1	1-1 = 0
A8- A11					1				1-1 = 0
A12						1	1		2-1 = 1
A13								1	1-1 = 0
A14								1	1-1 = 0
A15									
Total Connection									2

Tabel 5.2 : Conection Graph Matrix

4.6. Test Case 6

Pengujian : `simpan_perawatan.php`

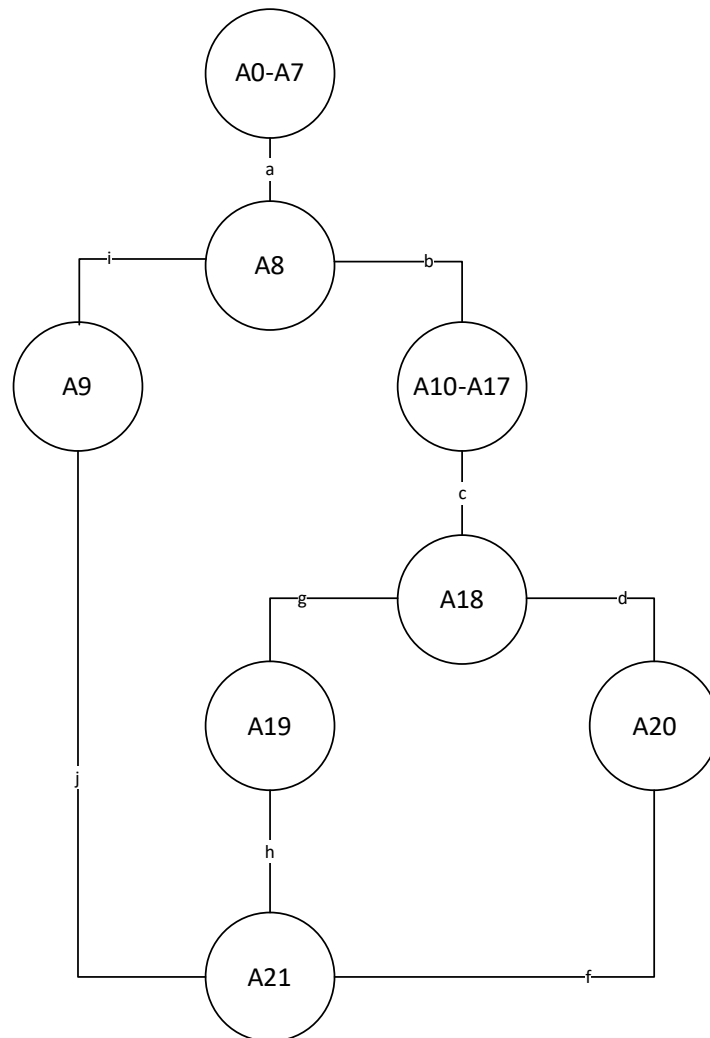
Jenis Pengujian : Basis path, Complexity Cyclometric dan Graph Matrix.

A0	<code><?php</code>
A1	<code>\$id_perawatan = \$_POST['id_perawatan'];</code>
A2	<code>\$barang = \$_POST['barang'];</code>
A3	<code>\$tgl_perawatan = \$_POST['tgl_perawatan'];</code>
A4	<code>\$keterangan = \$_POST['keterangan'];</code>
A5	<code>\$biaya_perawatan = \$_POST['biaya_perawatan'];</code>
A6	<code>\$opstatus = \$_POST['opstatus'];</code>
A7	<code>\$conn = new mysqli('localhost', 'root', '', 'aset_kantor');</code>
A8	<code>if (\$conn->connect_error) {</code>
A9	<code>die("Connection failed: " . \$conn->connect_error);</code>
A10	<code>}</code>
A11	<code>session_start();</code>
A12	<code>\$pengguna = \$_SESSION['id_pengguna'];</code>
A13	<code>\$sql = "INSERT INTO `perawatan`</code>
A14	<code>(`id_perawatan`, `id_barang`, `tgl_perawatan`, `keterangan`,</code> <code>`biaya_perawatan`, `status`, `id_pengguna`)</code>
A15	<code>VALUES</code>
A16	<code>(`\$id_perawatan`, `\$barang`, `\$tgl_perawatan`, `\$keterangan`, `\$biaya_per` <code>awatan`, `\$opstatus`, `\$pengguna`) ";</code></code>
A17	<code>\$result = \$conn->query(\$sql);</code>
A18	<code>if(\$result){</code>
A19	<code>echo "Data Berhasil di Simpan";</code>
A20	<code>}</code>

A21	>?
-----	----

Tabel 6 : *Code* simpan_perawatan.php

4.6.1. Basis Path



Gambar 6 : *Basis Path* simpan_perawatan.php

4.6.2. Complexity Cyclometric

$$V(G) \rightarrow Region = V(3)$$

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

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

$$V(G) = 3$$

4.6.3. Independent Path

1. A0-A7 – A8 – A11-A17 – A18 – A21
2. A0-A7 – A8 – A9 – A10 – A11-A17 – A18 – A21

3. A0-A7 – A8 – A9 – A10 – A11-A17 – A18 – A19 – A20 – A21

4.6.4. Test Case

I.

1. A(8) = False
2. A(18) = False

II.

1. A(8) = False
2. A(18) = True

III.

1. A(8) = True
2. A(18) = False

IV.

1. A(8) = True
2. A(18) = True

4.6.5. Graph Matrix

4.6.5.1. Tabel Graph Matrix

To From	A0-A7	A8	A9	A10- A17	A18	A19	A20	A21
A0- A7		a						
A8			i	b				
A9								j
A10- A17					c			
A18						g	d	
A19								h
A20								f
A21								

Tabel 6.1 : Graph Matrix

4.6.5.2. Tabel Conection Matrix

To From	A0-A5	A6	A7	A8- A16	A17	A18	A19	A20	Connection
A0- A5		1							1-1 = 0
A6			1	1					2-1 = 1
A7								1	1-1 = 0
A8- A16					1				1-1 = 0
A17						1	1		2-1 = 1
A18								1	1-1 = 0
A19								1	1-1 = 0
A20									
Total Connection									2

Tabel 6.2 : Conection Graph Matrix