Jurnal Siliwangi Vol. 9.No.1, 2023 **P-ISSN 2477-3891**
Seri Sains dan Teknologi **E-ISSN 2615-4765**

IMPLEMENTASI WHITE BOX TESTING BERBASIS PATH PADA APLIKASI BERBASIS WEB

Muhammad Ghibran AL Khamaeni
Program Studi Informatika Fakultas Teknik Universitas Siliwangi
197006056@student.unsil.ac.id

Abstrak

Pengujian perangkat lunak merupakan salah satu proses penting dalam tahapan pengembangan perangkat lunak. Pengujian perangkat lunak dilakukan untuk mengetahui jika terdapat kesalahan, serta menghasilkan perangkat lunak dengan kualitas baik. Pengujian perangkat lunak terbagi menjadi dua jenis, salah satunya metode *white box*. *White box testing* merupakan pengujian perangkat lunak yang berfokus pada desain dan kode program. Terdapat beberapa teknik pengujian pada *white box testing*, salah satunya adalah *basic path.* Teknik *basic path* merupakan teknik yang dapat mengukur tingkat kompleksitas dari kode program dan mendefinsikan alur yang akan dieksekusi. Tujuan dari penelitian ini, menerapkan *white box testing* dengan teknik *berbasis path*. Sehingg dapat diketahui setiap jalur eksekusi kode program, serta memastikan bahwa setiap jalur yang ada dieksekusi minimal satu kali. Tahapan yang dilakukan dalam pengujian *white box* dengan menggunakan teknik *basic path* diantaranya adalah: membuat diagram alir (*fowchart*), membuat grafik alir (*flowgraph*), menghitung *Cyclomatic Compelxity* (CC), menentukan jalur independen, dan melakukan uji kasus (*test case*). Hasil percobaan pada penelitian ini diketahui CC dengan jumlah jalur independen dua jalur yang berarti risiko error dari aplikasi tersebut cukup rendah. *Test case* yang dibuat diperoleh hasil valid menunjukkan bahwa sistem login pada aplikasi sederhana yang dibuat dapat berjalan tanpa adanya kesalahan.
Kata Kunci: white box, basis path, grafik alir, cyclomatic complexity

Abstract

Software testing is one of the important processes in the software development stage. Software testing is carried out to find out if there are errors, and to produce software of good quality. Software testing is divided into two types, one of which is the white box method. White box testing is software testing that focuses on design and program code. There are several testing techniques in white box testing, one of which is basic path. The basic path technique is a technique that can measure the level of complexity of program code and define the flow to be executed. The aim of this research is to apply white box testing with path-based techniques. So you can know every program code execution path, and ensure that each existing path is executed at least once. The stages carried out in white box testing using basic path techniques include: making a flowchart, making a flowgraph, calculating Cyclomatic Compelxity (CC), determining independent paths, and carrying out test cases. The experimental results in this research show that CC has two independent paths, which means the risk of error from the application is quite low. The test case created obtained valid results showing that the login system in the simple application created can run without any errors.

Keywords*: white box, basis path, flow graph, cyclomatic complexity*

I. PENDAHULUAN

Pada proses pengembangan sebuah sistem atau aplikasi terdapat beberapa tahapan, diantaranya proses analisa, perancangan, implementasi, uji coba, dan pengelolaan. Dari kelima proses ini, proses uji coba adalah proses yang memerlukan waktu yang cukup lama. Untuk menjamin kualitas sebuah sistem atau aplikasi harus melalui tahap uji coba [1].

Ada dua metode yang digunakan dalam pengujian, yaitu secara fungsional (*Black box*) dan secara sistematik (*White box*) [2]. Pengujian perangkat lunak merupakan sebuah proses menelusuri dan mempelajari sebuah program dalam rangka menemukan kesalahan pada perangkat lunak sebelum diserahkan kepada pengguna. Namun, pengujian perangkat lunak menjadi tugas yang memakan waktu dan mahal. Tahap pengujian mampu menghabiskan hampir 50% sumber daya pengembangan perangkat lunak.

Pengujian perangkat lunak juga dapat didefinisikan sebagai proses verifikasi dan validasi pada perangkat lunak untuk memastikan bahwa aplikasi telah memenuhi persyaratan teknis dan bisnis seperti yang diharapkan [3].

Pada pengujian *white box* terdapat beberapa teknik, salah satunya adalah *basic path / path testing*. Basis *path* adalah suatu jalur unik yang melintasi alur program dan tidak diperbolehkan terjadinya perulangan lintasan yang sama [1].

Penelitian ini melakukan pengujian *white box* terhadap sistem *login* pada aplikasi sederhana yang dibangun dengan menggunakan bahasa pemrograman HTML, CSS, dan juga PHP. Sistem *login* tersebut bertujuan agar pengguna dapat mengakses atau masuk ke dalam aplikasi yang dibuat. Pada penelitian ini akan menguji proses *login* tersebut, apakah yang akan terjadi jika pengguna memasukkan *email* dan kata sandi yang telah terdaftar dengan benar serta memasukkan *email* dan kata sandi yang salah. Pada penelitian ini menggunakan pengujian *white box* teknik basis *path*.

II. METODOLOGI PENELITIAN

Pada penelitian ini menggunakan metode penelitian kuantitatif, dimana mengumpulkan data-data hasil pengujian *white box* dengan teknik basis *path*.

Pengujian White Box

Pengujian *white* box adalah pengujian perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan [4].

Teknik Basic Path

Metode jalur dasar (*basic path*) adalah salah satu metode *white box testing*, di mana dalam proses pengujian diperlukan untuk membuat *flow graph* dari program skrip dan juga menentukan nilai kompleksitas siklomatik. Tes ini bertujuan untuk menganalisis kebenaran struktur program yang dibuat dan kinerja program [5].

Langkah-langkah untuk melakukan pengujian jalur dasar adalah [6]:

- Menggambarkan grafik alir (*flowgraph*) berdasarkan algoritma perancangan prosedur/fungsi.
- Menentukan *cyclomatic complexity*.
- Menentukan jalur-jalur dasar sesuai dengan jumlah dari *cyclomatic complexity*.
- Mendefinisikan kasus-kasus uji untuk setiap jalur dasar yang telah ditentukan.

Flowgraph

Grafik alir (*flowgraph*) adalah sebuah notasi sederhana yang merepresentasikan aliran kontrol dari sebuah struktur program [7].

Dalam sebuah grafik alir, anak panah disebut sebagai sisi (*edge*, E) merepresentasikan aliran kontrol. Lingkaran disebut sebagai simpul (*node*, N) merepresentasikan satu atau lebih aksi/pernyataan logis. Daerah yang dibatasi oleh sisi dan simpul disebut area (*region*, R). Simpul yang mengandung keputusan disebut sebagai (*predicate node*, P) yaitu simpul yang mengeluarkan lebih dari satu sisi [7].

Cyclomatic Compelxity

Cyclomatic Complexity adalah sebuah satuan perangkat lunak yang memberikan ukuran kuantitatif dari logika kompleksitas sebuah program. Saat digunakan dalam konteks metode pengujian basis path, satuan ini menunjukkan alur independen pada progam yang diuji [8]. Pendekatan ukuran kompleksitas dilakukan untuk mengukur dan mengontrol jumlah alur melalui program [9]. Nilai *cyclomatic complexity* dapat ditentukan dengan menggunakan salah satu rumus berikut.

- $V(G) = jumlah\ area\ (R)$
- $V(G) = E - N + 2$
- $V(G) = P + 1$

Dalam melakukan penelitian ini, terdapat beberapa tahapan yang dilalui, diantaranya adalah analisis data, proses pengerjaan, prosedur cara kerja sistem, dan skenario pengujian sistem [10].

Analisis Data

Analisis data, data-data yang digunakan pada penelitian merupakan data dari *source code* yang digunakan dalam membangun aplikasi sederhana berbasis *web*. Data yang dujikan adalah index.php yang berisi *source code* sistem *login* aplikasi tersebut.

Proses Pengerjaan

Proses pengerjaan penelitian ini adalah menguji fungsi-fungsi sistem *login* pada file index.php. Tahapan pengujiannya yaitu, membuat *flowchart*, membuat *flowgraph*, menentukan nilai dari *flowgraph*, menghitung nilai *cyclomatic complexity* (CC), menentukan jalur independen,

dan membuat uji kasus dari skenario. Alur proses tersebut sesuai dengan teknik *basic path* pada *white box testing*.

Prosedur Cara Kerja Sistem

Prosedur cara kerja sistem diperoleh dengan panduan dari pembuatan program aplikasi yang akan diujikan.

Skenario Pengujian Sistem

Skenario pengujian sistem diambil dari hasil jalur independen dari perhitungan *cyclomatic complexity*, format tabel dari skenario pengujian sistem ditampilkan pada tabel 1.

Tabel 1. Format tabel *test case*

No	Kegiatan	Hasil Yang Diharapkan	Hasil	Ket
1	....	....	....	....
2	....	....	....	....

III. HASIL DAN PEMBAHASAN

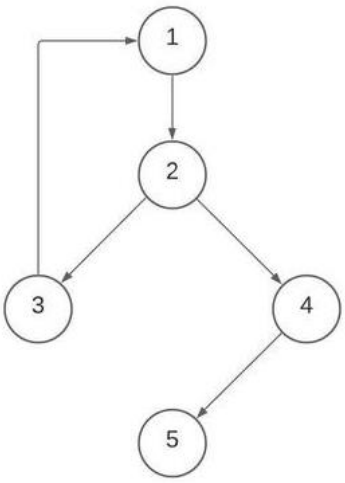
A. Source Code Sistem Login Pada Index.php

```
<?php

$server = "localhost";
$user = "root";
$pass = "";
$database = "pis_login2";

$conn = mysqli_connect($server, $user,
$pass, $database);

if (!$conn) {
    die("<script>alert('Gagal tersambung ke
database.')</script>");
}

error_reporting(0);

session_start();

if (isset($_SESSION['username'])) {
    header("Location: berhasil_login.php");
}

if (isset($_POST['submit'])) {
    $email = $_POST['email'];
    $password = $_POST['password'];

    $sql = "SELECT * FROM users WHERE
email='$email' AND password='$password'";
    $result = mysqli_query($conn, $sql);
    if ($result->num_rows > 0) {
        $row = mysqli_fetch_assoc($result);
        $_SESSION['username'] =
$row['username'];
        header("Location:
berhasil_login.php");
    } else {
        echo "<script>alert('Email atau
password Anda salah. Silahkan coba
lagi!')</script>";
    }
}
?>
```

B. Flowchart Fungsi Login

Gambar 1. Flowchart Sistem Login

C. Flowgraph



Gambar 2. Flowgraph Sistem Login

Dari gambar 2, diketahui sebagai berikut, *node* = 5; *edge* = 5; *region* = 2; *predicate* = 1. Sehingga dapat diperoleh perhitungan *cyclomatic compelxity* sebagai berikut:

- $V(G) = jumlah\ area\ (R)$
  $V(G) = 2$
- $V(G) = E - N + 2$
  $V(G) = 5 - 5 \mp 2$
  $V(G) = 2$
- $V(G) = P + 1$
  $V(G) = 1 + 1$
  $V(G) = 2$

Dari hasil perhitungan di atas, diperoleh jalur independen sebagai berikut:

1. 1-2-4-5 (if = true)
2. 1-2-3-1-2-4-5 (if = false)

Dari jalur independen yang diperoleh, dapat dibuat tabel *test case* (kasus uji) dengan hasil seperti ditampilkan pada tabel 2.

Tabel 2. Hasil Test Case

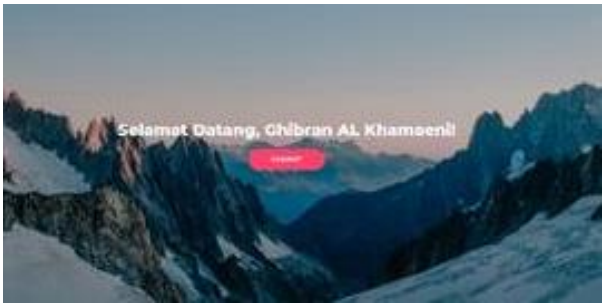
No	Kegiatan	Hasil Yang Diharapkan	Hasil	Ket
1	Mengetikkan email dan password yang sesuai (terdaftar pada database)	Berhasil masuk ke beranda aplikasi	Berhasil masuk ke beranda aplikasi	Valid
2	Mengetikkan email dan password yang tidak sesuai (email salah, password benar), (email benar, password salah), (email dan password salah), (tidak mengisi email dan password atau salah satunya)	Tidak dapat masuk ke beranda aplikasi dan muncul notifikasi email/password salah, atau email dan password harus diisi	Tidak dapat masuk ke beranda aplikasi dan muncul notifikasi email/password salah, atau email dan password harus diisi	Valid

Berdasarkan tabel 2 dapat diperoleh hasil bahwa keduanya adalah valid yang berarti tidak ditemukannya eror pada sistem login yang diuji.

D. Hasil Uji Coba

1. Memasukkan email dan password yang sesuai (terdaftar pada *database*)

Gambar 3. Input email dan password yang sesuai



Gambar 4. Berhasil masuk ke beranda aplikasi

2. Memasukkan email tidak sesuai dan password sesuai

Gambar 5. Input email tidak sesuai dan password sesuai

Gambar 6. Notifikasi email atau password salah

3. Memasukkan email sesuai dan password tidak sesuai

Gambar 7. Input email sesuai dan password tidak sesuai

Gambar 8. Notifkasi email atau password salah

4. Memasukkan email dan password yang tidak sesuai

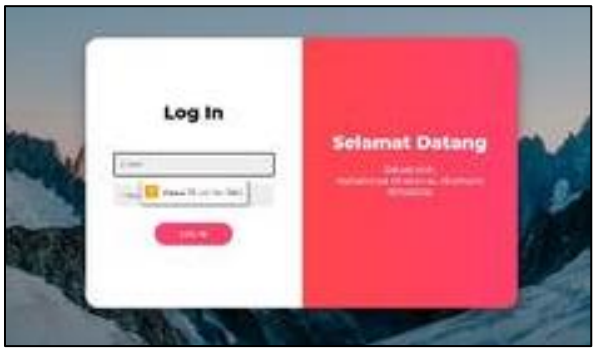
Gambar 9. Input email dan password tidak sesuai

Gambar 10. Notifkasi email atau password salah

5. Tidak mengisi email, password, atau keduanya

Gambar 11. Notifikasi harus mengisi password

Gambar 12. Notifikasi harus mengisi email



Gambar 13. Notifikasi harus mengisi email dan password

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, pada prototipe aplikasi yang dikembangkan, setelah dilakukan pengujian, diperoleh nilai *cyclomatic complexity (CC)* dengan 2 jalur independen. Nilai tersebut menunjukan bahwa risiko *error* dari aplikasi tersebut terbilang cukup rendah. Hasil tabel *test case* yang dibuat, dinyatatakan setiap case dinyatakan valid, hal tersebut menunjukkan bahwa protitipe aplikasi yang dibuat dapat berjalan tanpa adanya kesalahan.

DAFTAR PUSTAKA

[1] C. T. Pratala, E. M. Asyer, I. Prayudi, and A. Saifudin, “Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path,” *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 111, 2020, doi: 10.32493/informatika.v5i2.4713.

[2] Y. Irawan, “Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus dengan Metode

Whitebox Testing," *Sentra Penelit. Eng. dan Edukasi*, vol. 9, no. 3, pp. 59–63, 2017.

[3] C. Sharma, S. Sabharwal, and R. Sibal, "A Survey on Software Testing Techniques using Genetic Algorithm," vol. 10, no. 1, pp. 381–393, 2014, [Online]. Available: http://arxiv.org/abs/1411.1154.

[4] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, 2018, doi: 10.30998/string.v3i2.3048.

[5] I. G. S. Rahayuda and N. P. L. Santiari, "Basis Path Testing of Iterative Deepening Search and Held-Karp on Pathfinding Algorithm," *Kursor*, vol. 9, no. 2, pp. 39–48, 2018, doi: 10.28961/kursor.v9i2.129.

[6] R. S. Pressman, *Software Engineering: A Practitioner's Approach*. Boston, 2005.

[7] T. A. Kurniawan, "Pengujian Struktur Program Dengan Pengujian Jalur Dasar (Basis Path Testing) : Teori Dan Aplikasi," *Eeccis*, vol. 1, no. 1, pp. 29–32, 2007, [Online]. Available: http://jurnaleeccis.ub.ac.id/index.php/eeccis/article/viewFile/357/266.

[8] M. E. Khan, "Different Approaches to White Box Testing Technique for Finding Errors," vol. 5, no. 3, pp. 1–14, 2011.

[9] C. P. C. Munaiseche, G. C. Rorimpandey, T. Informatika, F. Teknik, and U. N. Manado, "Penerapan Metode Basis Path Analysis dalam Pengujian White Box Sistem Pakar," pp. 124–128.

[10] R. Subagia, R. Alit, and F. A. Akbar, "Pengujian white box pada sistem informasi monitoring skripsi program studi informatika," *J. Inform. dan Sist. Inf.*, vol. 01, no. 2, pp. 539–547, 2020.