

## IMPLEMENTASI WHITE BOX TESTING DENGAN TEKNIK BASIS PATH PADA PENGUJIAN FORM LOGIN

Muammar Farhan Londjo

Program Studi Informatika Fakultas Teknik Universitas Siliwangi Tasikmalaya  
e-mail: 197006024@student.unsil.ac.id

### Abstrak

Form login dalam sebuah aplikasi dapat digunakan sebagai autentikasi pengguna dengan hak akses yang berbeda. Form login yang dibuat dengan buruk dapat menyebabkan bahaya kredensial yang dapat disalahgunakan untuk mendapatkan akses oleh yang tidak berwenang. Tujuan penelitian ini untuk melakukan pengujian terhadap sebuah form login menggunakan metode *White Box* dengan teknik *basis path*. Teknik ini menguji aplikasi dengan mencari jalur independen dengan cara membuat *flowgraph* dan menghitung *Cyclomatic Complexity* untuk menentukan jumlah jalur independen yang ada pada alur aplikasi. Apabila sebuah aplikasi memiliki tingkat *Cyclomatic Complexity* yang rendah maka tingkat kemungkinan terjadinya *error* pun sangat rendah. Hasil luaran yang diujikan pun dibandingkan dengan hasil yang diharapkan dalam tabel *test case*. Setelah dilakukan uji pada Aplikasi Form Login dan tidak ditemukan *error* dalam pengujian, maka semua tes berhasil. Hasil pengujian ini dapat dijadikan acuan untuk memperbaiki aplikasi sesuai yang diharapkan, dan menjamin bahwa aplikasi yang dibuat dapat berfungsi sesuai dengan harapan.

**Kata Kunci:** *White Box*, *Basis Path*, Form Login

### Abstract

*The login form in an application can be used to authenticate users with different access rights. A poorly crafted login form can lead to dangerous credentials that could be misused to gain access by unauthorized persons. The purpose of this study is to test a login form using the White Box method with the base path technique. This technique tests the application by finding independent paths by creating a flowgraph and calculating Cyclomatic Complexity to determine the number of independent paths in the application flow. If an application has a low level of Cyclomatic Complexity, the probability of error is very low. The results tested are also compared with the expected results in the test case table. After testing on the Login Form Application and no errors were found in the test, then all tests were successful. The results of this test can be used as a reference to improve the application as expected, and ensure that the application made can function as expected..*

**Keywords:** *White Box*, *Basis Path*, Login Form

### I. PENDAHULUAN

Form login merupakan salah satu komponen aplikasi yang sangat penting. Sebuah form login dibutuhkan dalam suatu aplikasi agar pengguna dapat memiliki akses ke berbagai fitur yang disediakan oleh aplikasi. Form login juga merupakan salah satu cara untuk menambah lapisan keamanan dalam suatu aplikasi yang memiliki beberapa tingkatan *user*. Misalkan dalam suatu aplikasi ada 2 tipe akun yaitu, akun *user* dan akun *administrator*. Agar dapat membedakan akses *administrator* dan akses pengguna biasa (*user*) digunakan form login, sehingga hak akses pengguna aplikasi dapat ditentukan berdasarkan tipe akun yang ia miliki.

Apabila suatu aplikasi memiliki form login yang cacat atau tidak berfungsi sesuai dengan harapan, maka keamanan aplikasi tersebut menjadi berkurang. Akibat paling fatal adalah bocornya hak akses akun

*administrator* karena form login memiliki kesalahan dalam fungsi verifikasi akun.

Agar suatu aplikasi dapat berfungsi dengan baik dan berjalan sesuai dengan harapan maka harus dilakukan pengujian. Dalam pengujian suatu aplikasi ada dua metode yang paling umum digunakan yaitu Pengujian *White Box* (pengujian secara sistematis) dan Pengujian *Black Box* (pengujian secara fungsional) [1]. Tujuan penelitian ini melakukan pengujian sebuah form login menggunakan metode *White Box* Testing dengan teknik *Basis Path*. Teknik *Basis Path* ini digunakan untuk mengetahui kompleksitas logika aplikasi [2].

Form login ini memiliki sistem yang sederhana karena hanya membaca nama sebuah file sebagai validasi nama pengguna dan membaca isi file tersebut untuk validasi kata sandi. Form login yang telah dibuat juga akan diuji beberapa skenario terjadinya

kesalahan masukan atau kolom nama pengguna dan kata sandi tidak diisi.

## II. METODOLOGI

### Python

Python merupakan suatu bahasa pemrograman tingkat tinggi yang berorientasi objek dan dapat dijalankan di berbagai sistem operasi karena bersifat *multi-platform*. Python dibandingkan bahasa pemrograman lain lebih mudah dipelajari karena sintak-sintaknya lebih mirip dengan bahasa Inggris yang jelas. Python banyak digunakan untuk membuat berbagai aplikasi dan *script* untuk otomatisasi. Python sekarang sudah sangat populer dan komunitas pengguna bahasa pemrograman ini sudah tersebar di seluruh dunia [3].

Python juga bersifat *open source* yang berarti *source code* nya terbuka untuk umum sehingga semua orang dapat berkontribusi dalam mengembangkan bahasa Python. Kode Python dieksekusi menggunakan sebuah *interpreter*. Dalam pemanggilan kode dapat menggunakan CLI (*Command Line Interface*) atau dieksekusi sebagai *script*. Kedua metode pemanggilan kode di atas ada kelebihan dan kekurangannya masing-masing di mana menggunakan CLI lebih baik dalam pengujian satu per satu sedangkan *script* dapat lebih besar skala *deploy* nya [3].

### Pengujian White Box

Pengujian *White Box*, adalah suatu metode pengujian aplikasi yang menggunakan penjelasan struktur kontrol sebagai bagian dari *component-level design* untuk membuat *test cases*. *White Box* sendiri mempunyai beberapa teknik di dalam pengujiannya, seperti : *Data Flow Testing*, *Control Flow Testing*, *Basic Path / Path Testing*, dan *Loop Testing* [4]. Dalam Pengujian *White Box* para penguji perlu mengetahui secara dalam *source code* yang akan diuji. Pengujian *White Box* dapat mengungkap kesalahan implementasi dari sebuah aplikasi. Pengujian ini dapat diterapkan pada tingkatan integrasi, unit dan sistem [5].

Ada beberapa kelebihan dan kekurangan dalam pengujian menggunakan metode *White Box* antara lain: [5]

#### Kelebihan

- Metode *White Box* dapat memperlihatkan galat pada kode yang dibuat dengan menghapus baris yang tidak diperlukan.

- Maksimalnya cakupan pengujian aplikasi saat uji coba sebuah skenario.

#### Kekurangan

- Biaya pengujian menggunakan metode *White Box* sangatlah tinggi karena dibutuhkan penguji yang berpengalaman dalam bidang ini
- Beberapa alur program akan dibiarkan tidak diuji karena tidak memungkinkan untuk menguji setiap baris kode untuk menemukan kesalahan

### Teknik Basis Path

Teknik *Basis Path* merupakan salah satu teknik Pengujian *White Box* yang pertama kali dikemukakan oleh Tom McCabe. Teknik ini memungkinkan penguji untuk mengukur kompleksitas logika dari rancangan prosedural. Skenario uji coba yang dibuat untuk menguji Teknik *Basis Path* ini dijamin akan mengeksekusi setiap *statement* dari aplikasi yang diujikan setidaknya satu kali saat tahap pengujian [6].

### Cyclomatic Complexity

*Cyclomatic Complexity* adalah besaran perangkat lunak yang menyediakan acuan kuantitatif kompleksitas suatu logika dalam program. Rumus menghitung *Cyclomatic Complexity* adalah sebagai berikut: [7]

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

Keterangan :

E = jumlah *edges* pada *flowgraph*

N = jumlah *nodes* pada *flowgraph*

P = jumlah *predicates nodes* pada *flowgraph*

### Flowchart

*Flowchart* adalah gambaran langkah-langkah dan urutan-urutan prosedur dari suatu aplikasi. *Flowchart* membantu para analis dan *programmer* untuk membuat suatu gambaran aplikasi yang dibuat dan memecahkannya ke dalam segmen yang lebih kecil agar lebih mudah dianalisis dengan ini *flowchart* mempermudah penyelesaian suatu masalah [8]. *Flowchart* digunakan pada perancangan sebuah aplikasi untuk menggambarkan logika aplikasi tersebut [9].

### Flowgraph

*Flowgraph* adalah grafik program yang dihasilkan dari pemetaan *flowchart* program yang ada untuk merepresentasikan aliran kontrol logika program yang ada [10]. *Flowgraph* digunakan pada tahapan pengujian yang berfokus pada penggambaran aliran dari sebuah program [9].

Penelitian ini dilakukan dengan pengujian secara langsung terhadap aplikasi yang telah dirancang dengan pendekatan kuantitatif. Oleh karena itu hasil dari penelitian ini sudah dilakukan hingga akhir untuk mengumpulkan fakta sebagai bukti penelitian. Penelitian ini dibagi menjadi 3 tahapan yaitu:

### Pembuatan Form Login

Langkah pertama dalam penelitian ini yaitu membuat sebuah aplikasi form login sederhana menggunakan bahasa pemrograman *Python*. Form Login ini menggunakan *library tkinter* untuk membuat *screen* agar program dapat berjalan menggunakan GUI.

Pada aplikasi form login ini terdapat 15 fungsi yang digunakan agar aplikasi dapat berjalan. Beberapa fungsi yang butuh tampilan GUI akan memanggil *screen* agar dapat muncul pada layar monitor (contoh : menu utama dan pop up error). Untuk fungsi yang tidak memerlukan GUI seperti fungsi verifikasi login maka tidak akan memanggil *screen* untuk menampilkan GUI. Daftar fungsi di dalam aplikasi Form Login adalah sebagai berikut:

- a. *Function delete\_screen2*
- b. *Function delete\_screen3*
- c. *Function delete\_screen4*
- d. *Function delete\_screen5*
- e. *Function delete\_screen6*
- f. *Function logout*
- g. *Function session*
- h. *Function login\_success*
- i. *Function user\_not\_filled*
- j. *Function password\_not\_recognised*
- k. *Function user\_not\_found*
- l. *Function form\_not\_filled*
- m. *Function login\_verify*
- n. *Function login*

### Pengujian Form Login

Pengujian aplikasi ini dimulai dengan menghitung jumlah skenario yang akan diuji dengan menggunakan *Cyclomatic Complexity* dengan rumus  $V(G) = E - N + 2$  dengan melihat hasil pembuatan *flowgraph*. Setelah mendapatkan nilai *Cyclomatic Complexity* maka dibuat skenario uji. Pengujian menggunakan dilakukan secara manual dengan melakukan pengujian satu per satu dan melihat hasil keluaran yang muncul dari setiap jalur independen tersebut.

### Analisis Form Login

Selanjutnya untuk tahap analisis dibuat sebuah

tabel *test case* lalu membandingkan hasil keluaran setelah percobaan dengan tabel *test case*. Bentuk tabel *test case* ditampilkan pada tabel 1.

Tabel 1. Format *test case*

No	Kegiatan	Hasil yang diharapkan	Hasil	Keterangan
1	-	-	-	-
2	-	-	-	-

Setelah selesai maka dapat ditarik kesimpulan dari penelitian ini dalam pengujian aplikasi form login menggunakan metode *White Box*.

## III. HASIL DAN PEMBAHASAN

### Source Code Validasi Login

Dalam pembuatan aplikasi form login ini, aplikasi melakukan validasi dengan memeriksa file yang berada di dalam folder yang sama di mana aplikasi *python* dijalankan. Nama pengguna diambil dari nama file yang dibuat sedangkan kata sandinya diambil dari isi file yang ada di dalam file tersebut. Dalam pengujian aplikasi ini telah dibuat sebuah file dengan nama "farhan" yang dijadikan nama pengguna dan didalamnya berisi kata sandi "197006024". Secara umum source code function login ditampilkan pada tabel 2.

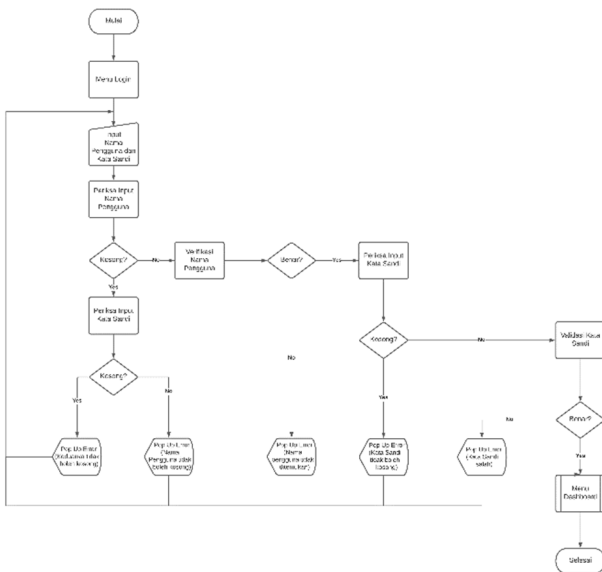
Tabel 2. *Function login\_verify*

```
def login_verify():
    username1 = username_verify.get()
    password1 = password_verify.get()

    list_of_files = os.listdir()
    if not username1:
        if not password1:
            form_not_filled()
        else:
            user_not_filled()
    elif username1 not in list_of_files:
        user_not_found()
    elif username1 in list_of_files:
        file1 = open(username1, "r")
        verify = file1.read().splitlines()
        if password1 in verify:
            login_success()
        elif not password1:
            password_not_filled()
    else:
        password_not_recognised()
```

### Flowchart Aplikasi Form Login

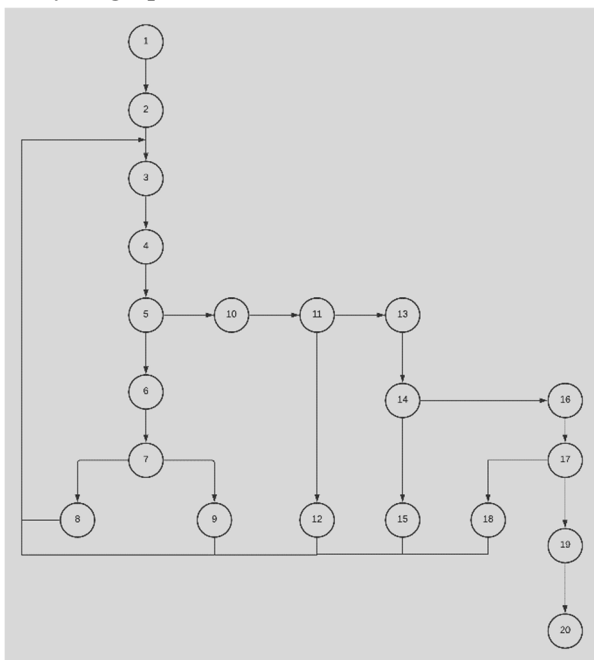
*Flowchart* ini dibuat agar dapat mengonversikannya ke dalam bentuk *flowgraph*. Secara umum *Flowchart Form Login* yang telah dibuat ditampilkan pada gambar 1.



Gambar 1. Flowchart Form Login

**Flowgraph Aplikasi Form Login**

Setelah pembuatan *flowchart* tahap berikutnya membuat *flowgraph*. Tujuan dibuatnya *flowgraph* ini untuk menghitung *Cyclomatic Complexity* yang membutuhkan jumlah *node* dan jumlah *edge* dari suatu *flowgraph*.



Gambar 2. Flowgraph Form Login

**Menghitung Jalur Independen Menggunakan Cyclomatic Complexity**

Untuk menghitung CC digunakan formula:

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

Dari hasil pembuatan *flowgraph* sebelumnya

dapat diketahui nilai berikut:

$$E = 24$$

$$N = 20$$

Setelah nilai tersebut dimasukkan ke dalam formula akan menghasilkan:

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

$$V(G) = 24 - 20 + 2$$

$$V(G) = 4 + 2$$

$$V(G) = 6$$

Dari hasil perhitungan tersebut didapatkan 6 jalur independen yaitu:

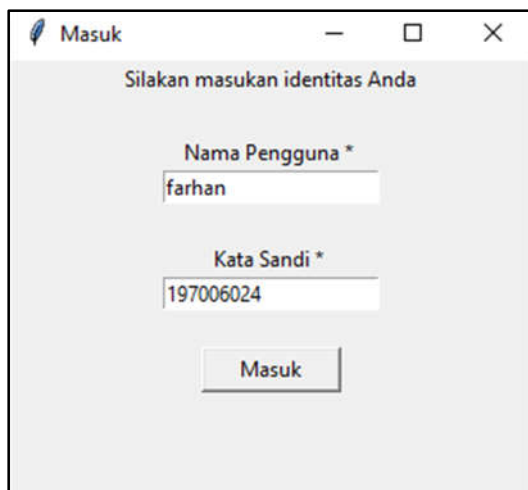
- 1-2-3-4-5-10-11-13-14-16-17-19-20 (Skenario login berhasil)
- 1-2-3-4-5-6-7-8-3-4-5-10-11-13-14-16-17-19-20 (Skenario login gagal karena nama pengguna dan kata sandi kosong)
- 1-2-3-4-5-6-7-9-3-4-5-10-11-13-14-16-17-19-20 (Skenario login gagal karena nama pengguna kosong sedangkan kolom kata sandi terisi)
- 1-2-3-4-5-10-11-12-3-4-5-10-11-13-14-16-17-19-20 (Skenario login gagal karena nama pengguna salah)
- 1-2-3-4-5-10-11-13-14-15-3-4-5-10-11-13-14-16-17-19-20 (Skenario login gagal karena nama pengguna benar tetapi kata sandi kosong)
- 1-2-3-4-5-10-11-13-16-17-18-3-4-5-10-11-13-14-16-17-19-20 (Skenario login gagal karena nama pengguna benar tetapi kata sandi salah)

Setelah selesai menentukan jalur independen tahap berikutnya, membuat tabel test case dan bandingkan hasil luaran nyata dengan hasil yang diharapkan. Tabel test case yang telah dibuat ditampilkan pada tabel 3.

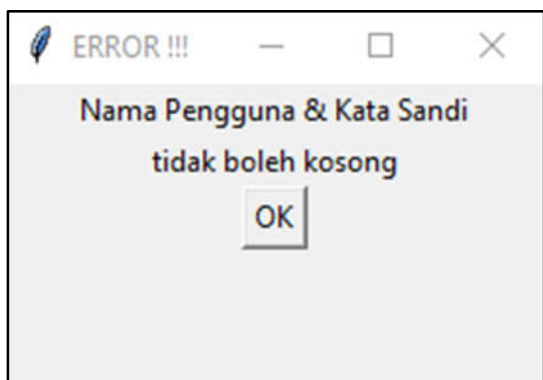
Tabel 3. Test Case

No	Kegiatan	Hasil yang diharapkan	Hasil	Keterangan
1	Input nama pengguna dan kata sandi yang benar	Masuk ke Dashboard	Masuk ke Dashboard	Valid
2	Input nama pengguna dan kata sandi kosong	Muncul Pop Up Error	Muncul Pop Up Error	Valid
3	Input nama pengguna kosong	Muncul Pop Up Error	Muncul Pop Up Error	Valid

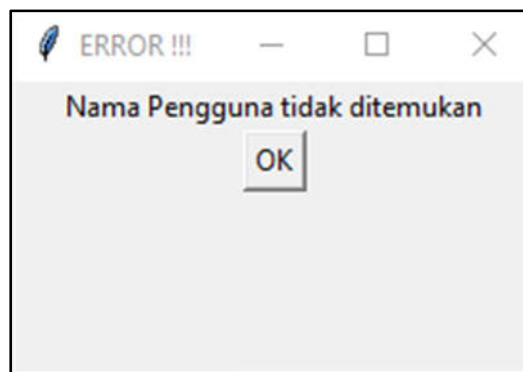
	sedangkan kolom kata sandi terisi			
4	Input nama pengguna salah	Muncul <i>Pop Up</i> <i>Error</i>	Muncul <i>Pop Up</i> <i>Error</i>	Valid
5	Input nama pengguna benar tetapi kata sandi kosong	Muncul <i>Pop Up</i> <i>Error</i>	Muncul <i>Pop Up</i> <i>Error</i>	Valid
6	Input nama pengguna benar tetapi kata sandi salah	Muncul <i>Pop Up</i> <i>Error</i>	Muncul <i>Pop Up</i> <i>Error</i>	Valid



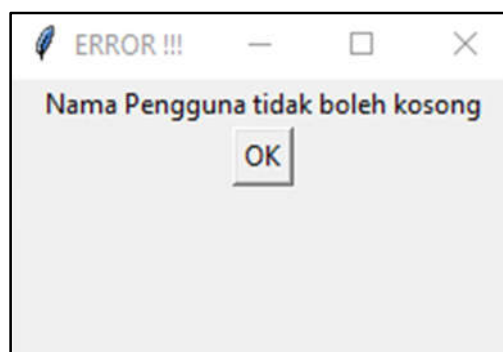
Gambar 3. Tampilan Form Login



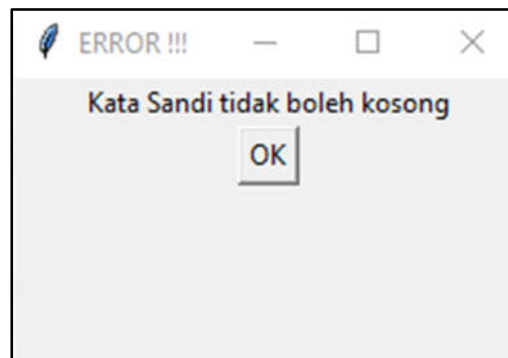
Gambar 4. Tampilan Pop Up Error Saat Nama Pengguna & Kata Sandi Kosong



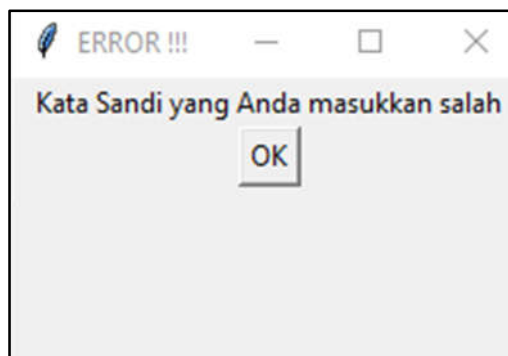
Gambar 5. Tampilan Pop Up Error Saat Nama Pengguna Salah



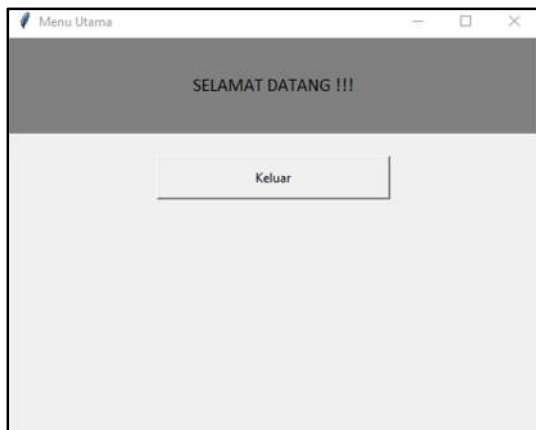
Gambar 6. Tampilan Pop Up Error Saat Nama Pengguna Kosong Sedangkan Kata Sandi Terisi



Gambar 7. Tampilan Pop Up Error Saat Nama Pengguna Benar Sedangkan kata sandi Kosong



Gambar 8. Tampilan Pop Up Error Nama Pengguna Benar Sedangkan Kata Sandi Salah



Gambar 9. Tampilan Menu Dashboard Apabila Berhasil Login

Dari hasil luaran nyata dan hasil luaran yang diharapkan dalam *test case* dan sudah benar semua dan tidak ada kesalahan dalam Form Login yang telah dibuat.

#### IV. KESIMPULAN DAN SARAN

Aplikasi Form Login yang telah diuji dengan metode *White Box - Basis Path* tidak terdapat kesalahan pada proses verifikasi login aplikasi tersebut.

Aplikasi di sini belum maksimal karena dalam menyimpan data nama pengguna dan kata sandi masih menggunakan file teks yang tidak terenkripsi dan lokasinya sama dengan tempat aplikasi dijalankan. Untuk sistem keamanan yang lebih baik form login dapat dihubungkan dengan *database* agar kata sandi dapat terenkripsi dan data pengguna tidak mudah diambil oleh pihak yang tidak berwenang.

#### DAFTAR PUSTAKA

- [1] 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.
- [2] 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.
- [3] I. Ghazali, "Making Game the New Adventurer Using Python," no. 50404370, pp. 4–5, 2020.
- [4] V. P. Katiyar and S. Patel, "White-Box Testing Technique for Finding Defects," *Glob. J. Res. Anal.*, vol. 8, no. 7, pp. 83–85,

- 2019, [Online]. Available: <http://worldwidejournals.co.in/index.php/gjra/article/view/4883>.
- [5] K. Mohd. Ehmer and K. Farneena, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 6, pp. 12–15, 2012.
- [6] Handy and J. Susilo, "Aplikasi Pengujian White-Box Ibi Online Judge," *J. Inform. dan Bisnis*, vol. 3, pp. 56–68, 2014.
- [7] E. sita Eriana, "Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web Dengan White Box Testing," *J. Teknol. Inf. ESIT*, vol. XV, no. 10, pp. 28–33, 2020.
- [8] Adelia and J. Setiawan, "Implementasi Customer Relationship Management ( CRM ) pada Sistem Reservasi Hotel berbasis Website dan Desktop," *Bandung, Univ. Kristen Maranatha*, vol. 6, no. 2, pp. 113–126, 2011.
- [9] D. D. A. Suprpto, F. Fauziah, I. Fitri, and N. Hayati, "Pengembangan Aplikasi Sistem Informasi Smart Register Online Berbasis Android Menggunakan Algoritma BruteForce," *Edumatic J. Pendidik. Inform.*, vol. 4, no. 1, pp. 47–56, 2020, doi: 10.29408/edumatic.v4i1.2106.
- [10] S. Pare, "Desain dan Implementasi E-commerce pada Toko AS88 Celluler Merauke," vol. 7, no. 1, pp. 104–126, 2017.