# Pengujian Basis Path Dengan Metode White Box Pada Portal Rosalia Indah

Kelfin Antonius[-1], Naufal Rafif[-2], Sri Rezeki Candra Nursari[-3]
Program Studi Teknik Informatika
Fakultas Teknik Universitas Pancasila
Jakarta, Indonesia
4521210018@univpancasila.ac.id[1], 4521210036@univpancasila.ac.id[2], sri.rezeki.candra.n@univpancasila.ac.id[3]

***Abstract*** — Dalam pembuatan perangkat lunak, pengembangan tidak dapat dilakukan begitu saja tanpa dilakukan pengujian oleh pengembang aplikasi. Pengujian pada perangkat lunak adalah salah satu cara mengidentifikasi *bug* dan *error*. Terdapat dua metode pengujian yang berbeda yaitu *black box* dan *white box*. Studi ini bertujuan untuk menguji portal Rosalia Indah pada fungsi masuk (login) dan fungsi pendaftaran. Pengujian ini dilakukan dengan teknik *basis path* yang termasuk dalam metode pengujian *white box*, pengujian akan memeriksa kemungkinan kesalahan yang terdapat pada portal Rosalia Indah. Untuk melakukan pengujian dengan teknik *basis path*, perlu dilakukan analisis dari alur program, lalu membuat *flowchart, flowgraph*, perhitungan *cyclometic complexity,* dan pembuatan tabel *test case.* Kesimpulan dari studi ini adalah fungsi masuk(login) dan fungsi pendaftaran berjalan sesuai yang diharapkan atau berhasil, artinya pengguna dapat dengan mudah mendaftar akun serta login ke akun tersebut untuk memesan tiket bis secara online tanpa adanya kendala pada website Rosalia Indah.

***Kata Kunci:*** *Pengujian, White Box, Basis Path, Pengujian Perangkat Lunak*

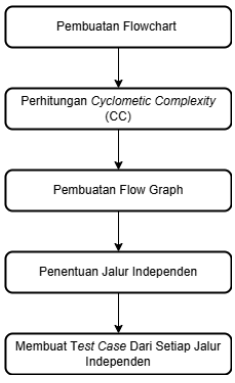
## 1. PENDAHULUAN

Pada era perkembangan teknologi yang pesat sekarang ini, banyak perangkat lunak yang dikembangkan dengan berbagai macam jenis sesuai kebutuhannya. Dalam pembuatan perangkat lunak, pengembangan tidak dapat dilakukan begitu saja tanpa dilakukan pengujian oleh pengembang aplikasi. Perangkat lunak perlu dilakukan pengujian sebelum di-*delivery* kepada pengguna. Pengujian pada perangkat lunak merupakan tahapan penting untuk memastikan kualitasnya sehingga terhindar dari berbagai masalah, contohnya adalah mencegah terlalu banyaknya *bug* pada sistem yang dibuat, *bug* yang terlalu banyak akan mengurangi efektivitas aplikasi perangkat lunak yang digunakan pengguna [1] dan kegagalan sistem yang mengakibatkan sebuah sistem tidak dapat berjalan sesuai fungsinya [2]. Walaupun demikian, pengujian pada perangkat lunak tidak dapat sepenuhnya mencegah resiko *error* dan *bug* dalam system perangkat lunak [2].

Salah satu metode pengujian adalah *white box*, pengujian ini berfokus untuk mengidentifikasi masalah atau kekurangan secara langsung dari struktur internal program yakni alur logika, input output program, alur program dan kode program [3], [4]. Terdapat beberapa teknik pengujian *white box* yang berbeda-beda yaitu data flow testing, control flow testing, basis path, dan loop testing [5]. Dalam pengujian ini, metode yang digunakan adalah basis path, basis path adalah jalur unik pada program dan pengulangan pada jalur yang sama tidak diperbolehkan [6]. Teknik ini terdiri dari penggambaran alur logika program dalam bentuk flowchart, kemudian *flowgraph*, lalu dilanjutkan dengan perhitungan *cyclometic complexity* dari jumlah *node* dan *edge* untuk menentukan jalur independen*, independent path* yang merupakan jalur yang dilalui oleh program setidaknya sekali, dan penentuan *test case* pada tahap akhir pengujian [7].

Rosalia indah adalah sebuah perusahaan yang sedang berkembang pesat yang berbisnis di bidang transportasi darat tepatnya bis antarkota, bis pariwisata, dan layanan carter bis. Tiket bis antarkota ini dapat dipesan secara langsung melalui website resminya. Untuk memastikan situs web Rosalia Indah telah bekerja dengan baik, dilakukanlah pengujian *white box,* salah satu teknik yang dapat dilakukan adalah menggunakan teknik basis *path*.

## 2. METODOLOGI PENELITIAN




Gambar 1. Metodologi Penelitian

Pengujian pada portal Rosalia Indah dimulai dengan menganalisis system yang berjalan serta menganalisis data berupa *source code* yang diambil dari portal Rosalia Indah.

Tahap selanjutnya adalah pembuatan flowchart dan flowgraph, kemudian dilanjutkan dengan perhitungan cyclomatic complexity (CC) untuk mengidentifikasi jumlah jalur independen yang dapat dibuat. Tahap berikutnya adalah penentuan *test case* untuk menguji aspek fungsional dari sistem [5] [8] [9].

Tahap pertama, pengujian *white box* dengan *basis path* diperlukan analisis dari kode program [10]. Oleh karena itu, pada tahap analisis data ini, kode program diambil dari website Rosalia Indah melalui inspeksi web atau sisi *frontend*, hal ini dikarenakan kode program yang terdapat pada sisi *backend* tidak memungkinkan untuk dianalisa.

Pada tahap kedua, pembuatan flowchart atau diagram alir. Flowchart adalah sebuah diagram yang digunakan untuk merepresentasikan alur algoritma pada kode program [11]. Diagram ini menggunakan simbol-simbol standar yang telah ditetapkan untuk mewakili berbagai jenis aktivitas, kondisi, dan arah aliran data dalam suatu sistem [4]. Tahap ini perlu dibuat sebelum pembuatan *flowgraph*.

Dilanjutkan dengan tahap ketiga setelah flowchart dibuat, maka flowgraph baru dapat ditentukan [12]. Flowgraph mengubah alur *sequence*, alur logika if else, atau *looping* dalam bentuk notasi berbentuk *node* dan *edge*. *Node* menunjukkan pernyataan prosedur, sedangkan *edge* merupakan representasi alur dari logika program [13]. Flowgraph dibuat berdasarkan diagram alir yang telah dibuat sebelumnya untuk form masuk (login) dan form daftar.

Perhitungan *cyclometic complexity* ditujukan untuk menentukan jalur independen. Adapun rumus yang digunakan untuk menghitung *cyclometic complexity* [14]: V(G) = E – N + 2; dimana E adalah jumlah edges sedangkan N adalah jumlah node dari flowgraph.

Tahap kelima adalah membuat jalur independen. Jumlah jalur didapatkan dari hasil perhitungan *cyclometic complexity*. Jalur independen merupakan jalur (*edge*) pada program yang berjalan dari *node* awal sampai *node* akhir [9]. Setiap jalur independen yang teridentifikasi perlu dilewati setidaknya satu kali selama pengujian untuk memastikan semua logika program telah teruji secara menyeluruh.

Setelah melalui semua tahapan, tahap terakhir adalah pembuatan tabel *test case*. *Test case* merupakan produk dari penentuan setiap jalur independen yang sudah ditentukan sebelumnya. Jumlah *test case* yang dibuat adalah minimal sebanyak jumlah jalur independen [15]. Pada setiap *test case* perlu didefinisikan output yang diharapkan atau *expected output* [7]. Dalam pemberian keterangan *test case* umumnya ialah keterangan berhasil dan keterangan gagal. Tabel *test case* dapat dilihat pada Tabel 1.

| No. | Jalur Independen | Rincian Pengujian | *Expected Output* | *Actual Output* | Keterangan |
|---|---|---|---|---|---|
| - | - | - | - | - | - |

Tabel 1. Tabel Pengujian *test case* [8].

## 3. HASIL DAN PEMBAHASAN

**Form Masuk (login)**

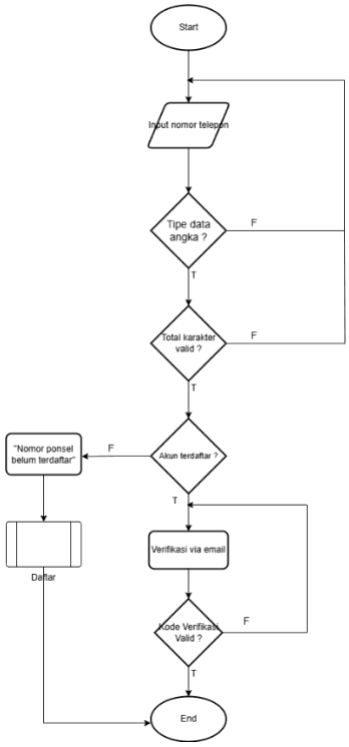
a. Source Code Form Masuk



Gambar 2. Source code pada form masuk

Pada gambar 2 diatas, merupakan HTML yang diambil dari inspeksi website melalui browser pada website rosalia indah.

b. *Flowchart* Form Masuk

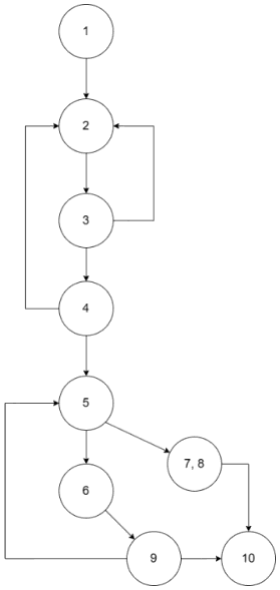



Gambar 3. Flowchart Login

Alur login pada portal Rosalia Indah ini dimulai dengan pengguna yang memasukkan nomor telepon yang telah didaftarkan, setelah itu form input akan mengecek apakah tipe data nya berupa angka, jika bukan angka maka input tidak akan tampil. Form input juga akan diperiksa bila total karakter sudah mencukupi atau belum. Alur login dilanjutkan dengan pengecekan apakah pengguna telah melakukan registrasi sebelumnya atau belum, jika belum terdaftar maka akan muncul notifikasi "Nomor ponsel belum terdaftar" dan diarahkan ke halaman daftar yang dilambangkan oleh *predefined process* seperti di gambar 2. Setelah dipastikan akun valid, maka kode verifikasi akan dikirimkan ke email

pengguna, bila kode tersebut valid maka pengguna dapat berhasil login.

c. *Flowgraph* Form Masuk




Gambar 4. Flowgraph Login

Pada tabel 2, terdapat action node dari flowgraph login diatas.

| Node | Action |
|---|---|
| Node 1 (1) | Start |
| Node 2 (2) | Masukkan nomor telpon<br>"Nomor Ponsel Belum Terdaftar" -> Perlu Register |
| Node 3 (3) | Nomor berupa angka? |
| Node 4 (4) | Nomor adalah angka -> Total karakter valid? |
| Node 5 (5) | Total karakter valid -> akun terdaftar? |
| Node 6 (6) | Akun terdaftar -> kode verifikasi terkirim melalui email |
| Node 7 (7, 8) | Total karakter valid -> akun tidak terdaftar -> muncul notifikasi "Nomor ponsel belum terdaftar" -> Daftarkan akun terlebih dahulu |
| Node 8 (9) | Kode verifikasi valid? |
| Node 9 (10) | Kode verifikasi valid -> End |

Tabel 2. Deskripsi setiap node

d. Perhitungan *Cyclometic Complexity (CC)*

Pada gambar 4 flowgraph form masuk (login), jumlah *edge* yang diketahui adalah 12, sedangkan *node* nya berjumlah sembilan. Dengan diketahui demikian, maka perhitungan *cyclometic complexity* nya adalah

$$V(G) = E - N + 2$$
$$V(G) = 12 - 9 + 2$$
$$V(G) = 5$$

e. Jalur Independen

Berdasarkan perhitungan cyclometic complexity sebelumnya, maka didapatkan lima jalur independen yang dapat terbentuk pada fungsi login.

| No. | Jalur Independen |
|---|---|
| 1 | 1 -> 2 -> 3 -> 2 |
| 2 | 1 -> 2 -> 3 -> 4 -> 2 |
| 3 | 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 9 -> 5 |
| 4 | 1 -> 2 -> 3 -> 4 -> 5 -> 7,8 -> 10 |
| 5 | 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 9 -> 10 |

Tabel 3. Jalur Independen form login.

f. Tabel *Test Case*

| No. | Jalur Independen | Rincian Pengujian | *Expected Output* | Actual Output | Keterangan |
|---|---|---|---|---|---|
| 1 | 1-2-3-4-5-7,8-10 | Pengguna memasukkan email dan nomor telepon yang belum terdaftar | Sistem akan mengarahkan pengguna ke halaman register | Sistem mengingatkan pengguna bahwa akun belum terdaftar dan harus menyelesaikan tahap registrasi | Berhasil |
| 2 | 1-2-3-2 | Pengguna memasukkan kolom nomor telepon bukan angka | Input tidak akan valid sampai pengguna memasukkan nomor angka | Kolom input tidak terisi ketika inputan bukan angka | Berhasil |
| 3 | 1-2-3-4-2 | Pengguna memasukkan nomor telepon kurang dari format nomor telepon Indonesia | Input tidak akan valid sampai pengguna memasukkan jumlah karakter sesuai | Kolom input mengeluarkan peringatan, dan form tidak bisa disubmit | Berhasil |
| 4 | 1-2-3-4-5-6-9-5 | Pengguna memasukkan nomor telepon yang valid, namun kode verifikasi tidak valid | Proses tidak akan berhasil sampai kode verifikasi dimasukkan dan valid | Proses login tidak berhasil ketika kode verifikasi salah | Berhasil |
| 5 | 1-2-3-4-5-6-9-10 | Pengguna memasukkan nomor telepon yang telah terdaftar dan valid | Sistem dapat memproses kode verifikasi yang valid dan proses login berhasil | Sistem dapat dengan cepat mengirimkan kode verifikasi dan pengguna dapat login | Berhasil |

Tabel 4. Tabel *test case* pada form masuk

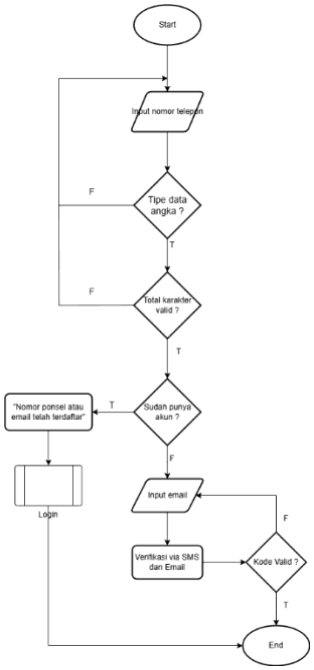
**Form Daftar**

a. *Source Code* Form Daftar



Gambar 5. *Source Code* Form Daftar

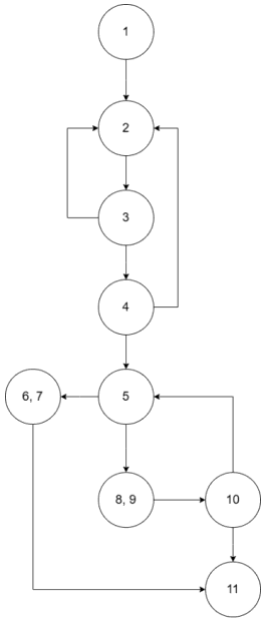
b. *Flowchart* Form Daftar




Gambar 6. Flowchart Form Daftar

Alur register pada portal Rosalia Indah ini dimulai dengan pengguna memasukkan nomor telepon dan email yang akan didaftarkan. Setelah itu form input akan mengecek apakah tipe data nya berupa angka, jika bukan angka maka input tidak akan tampil. Setelah dipastikan inputan angka, maka akan dicek kembali apakah total karakter valid atau tidak. Setelah jumlah karakter valid, pengecekan dilanjutkan dengan apakah pengguna telah melakukan registrasi sebelumnya atau belum, jika sudah terdaftar maka akan muncul notifikasi "Nomor ponsel atau email sudah terdaftar" dan akan diarahkan ke halaman login pada predefined process seperti di gambar 5. Jika akun belum terdaftar, maka pengguna dapat memasukkan email dan dapat melanjutkan ke tahap verifikasi. Pada tahap verifikasi, kode verifikasi akan dikirimkan ke nomor telepon dan email secara bersamaan, jika kedua kode verifikasi valid maka pengguna dapat melakukan daftar akun.

c. *Flowgraph* Form Daftar




Gambar 7. Flowgraph Form Daftar

Pada tabel 5 di bawah, merupakan deskripsi dari setiap jalur yang dilewati *flowgraph* form daftar pada gambar 7.

| I. Node | Action |
|---|---|
| Node 1 (1) | Start |
| Node 2 (2) | Input nomor telepon |
| Node 3 (3) | Nomor berupa angka? |
| Node 4 (4) | Nomor adalah angka -> Total karakter valid ? |
| Node 5 (5) | Total karakter valid -> Sudah punya akun ? |
| Node 6 (6,7) | Terdapat akun yang terdaftar -> diarahkan ke halaman login |
| Node 7 (8, 9) | Pengguna memasukkan email -> verifikasi akun melalui SMS dan Email |
| Node 8 (10) | Kode verifikasi valid? |
| Node 9 (11) | Kode verifikasi valid -> end |

Tabel 5. Deskripsi *flowgraph* Daftar.

d. Perhitungan *Cyclometic Complexity* (CC)

Pada gambar 7 *flowgraph* daftar, jumlah edge yang diketahui adalah 12, sedangkan node nya berjumlah sembilan. Dengan diketahui demikian, maka perhitungan cyclometic complexity nya adalah

$$V(G) = E - N + 2$$
$$V(G) = 12 - 9 + 2$$
$$V(G) = 5$$

e. Jalur Independen Form Daftar

Berdasarkan perhitungan cyclometic complexity sebelumnya, maka didapatkan lima jalur independen yang dapat terbentuk pada fungsi daftar di portal Rosalia Indah.

| No. | Jalur Independen |
|---|---|
| 1 | 1 -> 2 -> 3 -> 4 -> 5 -> 6,7 -> 11 |
| 2 | 1 -> 2 -> 3 -> 2 |
| 3 | 1 -> 2 -> 3 -> 4 -> 2 |
| 4 | 1 -> 2 -> 3 -> 4 -> 5 -> 8,9 -> 10 -> 5 |
| 5 | 1 -> 2 -> 3 -> 4 -> 5 -> 8,9 -> 10 -> 11 |

Tabel 6. Jalur Independen Form Daftar Rosalia Indah.

f. Tabel *Test Case* Pada Form Daftar

| No. | Jalur Independen | Rincian Pengujian | *Expected Output* | Actual Output | Keterangan |
|---|---|---|---|---|---|
| 1 | 1-2-3-4-5-6-7-11 | Pengguna memasukkan email dan nomor telepon yang sudah terdaftar | Sistem akan mengarahkan pengguna ke halaman login | Sistem mengingatkan pengguna bahwa akun sudah terdaftar dan harus login | Berhasil |
| 2 | 1-2-3-2 | Pengguna memasukkan kolom nomor telepon bukan angka | Input tidak akan valid sampai pengguna memasukkan nomor angka | Kolom input tidak terisi ketika inputan bukan angka | Berhasil |
| 3 | 1-2-3-4-2 | Pengguna memasukkan nomor telepon kurang dari format nomor telepon Indonesia | Input tidak akan valid sampai pengguna memasukkan jumlah karakter sesuai | Kolom input mengeluarkan peringatan, dan form tidak bisa disubmit | Berhasil |
| 4 | 1-2-3-4-5-8-9-10-5 | Pengguna memasukkan nomor telepon dan email yang valid namun kode verifikasi tidak sesuai | Proses registrasi tidak berhasil karena kode verifikasi salah | Sistem mengirimkan pesan error bahwa kode verifikasi berhasil dan pengguna tidak dapat membuat akun | Berhasil |
| 5 | 1-2-3-4-5-8-9-10-5 | Pengguna memasukkan nomor telepon, email yang valid | Sistem dapat memproses kode verifikasi yang valid dan pembuatan akun berhasil | Sistem dapat dengan cepat mengirimkan kode verifikasi dan akun pengguna dapat terdaftar | Berhasil |

Tabel 7. *Test case* form Daftar

## 4. KESIMPULAN

Pengujian menggunakan white box dengan teknik basis path pada form login dan form registrasi di website pemesanan tiket bis Rosalia Indah berjalan dengan baik, tanpa adanya identifikasi sistem yang mempunyai bug ataupun error. Tahapan yang dilakukan dalam pengujian ini dimulai dari tahap pemetaan source code, pembuatan

flowchart lalu diubah ke bentuk flowgraph, perhitungan cyclometic complexity, dan terakhir penentuan jalur independen. Semua fungsi test case diuji coba dan hasilnya berjalan sesuai yang diharapkan atau berhasil, artinya pengguna dapat dengan mudah mendaftar akun serta login ke akun tersebut untuk memesan tiket bis secara online tanpa adanya kendala pada website Rosalia Indah.

## 5. DAFTAR PUSTAKA

[1] M. F. Ahmadi and S. R. C. Nursari, "Pengujian Kondisi Struktur Kontrol pada Website (Studi Kasus: Erigo Official)," *Journal of Informatics and Advanced Computing*, vol. 3, no. 2, Nov. 2022.

[2] D. I. Pirdaus and R. A. Hidayana, "Analysis Testing Black Box and White Box on Application To-Do List Based Web," *International Journal of Mathematics, Statistics, and Computing*, vol. 2, no. 2, pp. 68–75, 2024.

[3] H. Rafli, R. Zen, and I. Nuryasin, "PENERAPAN WHITEBOX TESTING PADA PENGUJIAN SISTEM MENGGUNAKAN TEKNIK BASIS PATH," *Journal Of Information Systems And Informatics Engineering*, vol. 8, no. 1, pp. 101–111, 2024, doi: 10.35145/joisie.v8i1.4229.

[4] Alfian Permana Putra and Ilyas Nuryasin, "Pengujian sistem informasi monitoring dava kebab menggunakan white box testing dengan teknik basis path," *INFOTECH : Jurnal Informatika & Teknologi*, vol. 5, no. 1, pp. 63–75, Jun. 2024, doi: 10.37373/infotech.v5i1.1147.

[5] Y. J. Solissa, F. Putra, and A. N. Putri, "Pengujian White Box Berbasis Path pada Form Daftar Jobstreet.co.id," vol. 3, Dec. 2023.

[6] C. T. Pratala, E. M. Asyer, I. Prayudi, and A. Saifudin, "Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, p. 111, Jun. 2020, doi: 10.32493/informatika.v5i2.4713.

[7] J. B. L. S. Sie, I. A. Musdar, and S. Bahri, "PENGUJIAN WHITE BOX TESTING TERHADAP WEBSITE ROOM MENGGUNAKAN TEKNIK BASIS PATH," *Ilmu Komputer*, vol. 17, pp. 45–57, Sep. 2022.

[8] R. Subagia, R. Alit, and A. Akbar, "PENGUJIAN WHITE BOX PADA SISTEM INFORMASI MONITORING SKRIPSI PROGRAM STUDI INFORMATIKA," 2020.

[9] R. I. Ndaumanu, "Pengujian Sistem Informasi Perpustakaan Berbasis Website dengan Basis Path Testing," vol. 6, no. 1, pp. 123–134, 2023, doi: 10.31764/justek.vXiY.ZZZ.

[10] M. Farhan Londjo, "IMPLEMENTASI WHITE BOX TESTING DENGAN TEKNIK BASIS PATH PADA PENGUJIAN FORM LOGIN," vol. 7, no. 2, p. 2021, 2021.

[11] R. Andrian Ibrahim and G. Saktian Laksito, "Optimization of White Box Testing by Utilizing Branching and Repeating Structures in Java Programs Using Base Path," *International Journal of Mathematics, Statistics, and Computing*, vol. 2, no. 2, pp. 85–89, 2024.

[12] A. Dimas Saputro and A. Azzahra Narwastika, "IMPLEMENTASI WHITE BOX TESTING DENGAN TEKNIK BASIS PATH PADA PENGUJIAN FORM PEMINJAMAN SISTEM APLIKASI PERPUSTAKAAN," 2023.

[13] M. Y. Rafi, I. Yusuf Arifin, D. Safutri, D. Fadilah, and J. Riyanto, "Pengujian White Box Testing Menggunakan Teknik Loop Testing pada Aplikasi Sistem Informasi Perpustakaan (Studi Kasus SMKN 3 Kota Tangerang Selatan)," 2021. [Online]. Available: http://pijarpemikiran.com/index.php/Scientia

[14] H. Gusdevi *et al.*, "PENGUJIAN WHITE-BOX PADA APLIKASI DEBT MANAGER BERBASIS ANDROID," vol. 04, no. 1, Jun. 2022.

[15] C. Pamela, C. Munaiseche, and G. C. Rorimpandey, "Penerapan Metode Basis Path Analysis dalam Pengujian White Box Sistem Pakar," 2021.