

PENERAPAN TEKNIK *BASIS PATH* PADA PENGUJIAN *WHITE BOX* SISTEM INFORMASI PERENCANAAN DAN PENGANGGARAN RESPONSIVE GENDER DI DISKOMINFO KABUPATEN JOMBANG

Della Atika Putri*¹⁾, Henni Endah Wahanani*²⁾, Afina Lina Nurlaili³⁾

1. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia
2. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia
3. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia

Article Info

Kata Kunci: *Basis Path*; Sistem Informasi; *System Usability Scale*; *Usability*; *White box*

Keywords: *Basis Path*; *Information System*; *System Usability Scale*; *Usability*; *White box*

Article history:

Received 6 August 2024

Revised 30 August 2024

Accepted 2 September 2024

Available online 1 September 2025

DOI :

<https://doi.org/10.29100/jipi.v10i3.6338>

* Corresponding author.

Henni Endah Wahanani

E-mail address:

henniendah.if@upnjatim.ac.id

ABSTRAK

Sistem informasi merupakan media penting yang digunakan untuk menyediakan informasi secara akurat dan tepat waktu. Mengingat signifikansi sistem ini bagi organisasi, pengujian kualitas dan keandalan sistem menjadi krusial. Penelitian ini mengkaji sistem informasi perencanaan dan penganggaran responsif gender yang dikelola oleh Diskominfo Jombang dengan menerapkan teknik *basis path* dalam metode pengujian *white box*. Pengujian melibatkan pembuatan *flowgraph*, perhitungan *cyclomatic complexity (CC)*, penentuan jalur independen, dan pembuatan *test case*. Teknik *basis path* digunakan untuk memastikan bahwa setiap jalur dalam program dapat dilalui sekali tanpa adanya jalan pintas atau perulangan, melalui analisis kode program sistem. Hasil pengujian menunjukkan bahwa dari empat fungsi yang diuji, satu fungsi memiliki prosedur yang terstruktur dengan baik dan konsisten, sedangkan tiga fungsi lainnya sederhana dan memiliki risiko rendah. Secara keseluruhan, sistem ini dinilai memiliki risiko rendah. Namun, evaluasi *usability* menggunakan metode SUS menunjukkan bahwa, meskipun sistem berfungsi dengan baik dari segi logika internal, antarmuka yang rumit, serta navigasi yang membingungkan menyebabkan skor SUS yang rendah. Hal ini menunjukkan bahwa sistem belum sepenuhnya ramah pengguna dan memerlukan perbaikan.

ABSTRACT

Information systems are crucial media used to provide information accurately and timely. Given the importance of these systems to organizations, testing their quality and reliability is essential. This study examines the gender-responsive planning and budgeting information system managed by Diskominfo Jombang, using the path testing technique within the white box testing method. The testing involves creating flowgraphs, calculating cyclomatic complexity (CC), determining independent paths, and developing test cases. The path testing technique is used to ensure that each path in the program is traversed only once without shortcuts or repetitions, through an analysis of the system's code. The test results indicate that out of the four functions tested, one function has a well-structured and consistent procedure, while the remaining three functions are simple and have low risk. Overall, the system is assessed to have low risk. However, usability evaluation using the SUS method shows that, although the system performs well in terms of internal logic, the complex interface and confusing navigation result in a low SUS score. This indicates that the system is not yet fully user-friendly and requires improvement.

I. PENDAHULUAN

Teknologi informasi dan komunikasi saat ini berkembang pesat. Manusia mulai memanfaatkan teknologi ini untuk mempermudah pekerjaan, seperti mengelola, mengakses, serta menyebarkan informasi. Teknologi informasi yang sekarang ini sudah sangat maju dan banyak digunakan, salah satunya adalah sistem informasi. Sistem informasi adalah sebuah sistem yang terdiri dari berbagai komponen, seperti manusia, proses, teknologi, dan data yang bekerja sama untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi. Sistem

ini dapat membantu perusahaan dalam mengelola pekerjaan sehari-hari, membuat keputusan bisnis, dan memberikan laporan kepada pihak luar [1].

Sistem informasi menjadi salah satu media internet populer yang digunakan untuk memberikan informasi yang sesuai secara akurat, dan tepat waktu bagi pengguna. Selain itu, sistem informasi juga mendukung pengambilan keputusan, pengendalian, dan operasional dalam suatu organisasi. Selain perusahaan besar, sistem informasi juga menjadi salah satu kunci dalam operasional lembaga atau organisasi di tingkat daerah. Salah satu contoh sistem informasi yang dibuat oleh lembaga pemerintahan adalah Sistem informasi perencanaan dan penganggaran responsive gender (Siperansinder).

Sistem informasi perencanaan dan penganggaran responsive gender (Siperansinder) merupakan sebuah sistem informasi yang belgrelrak dibawah naungan Dinas Komunikasi dan Informatika Kabupaten Jombang. Siperansinder digunakan untuk mengoptimalkan fungsi koordinasi perencanaan dan penganggaran berbasis gender di Pemerintahan Kabupaten Jombang. Diharapkan, dengan adanya sistem ini, posisi, peran, dan kualitas perempuan akan meningkat, serta akan mendorong terciptanya kesetaraan dan keadilan gender dalam lingkungan keluarga, masyarakat, bangsa, dan negara.

Melihat betapa pentingnya sistem ini, peneliti akan melakukan pengujian untuk memvalidasi kinerja sistem tersebut dapat berjalan dengan baik tanpa ada kesalahan. Pengujian merupakan tahapan penting dalam pengembangan dan perancangan, menjadi dasar untuk memverifikasi sistem yang dikembangkan dan dirancang ini apakah dapat berfungsi relevan dengan kebutuhan yang telah ditentukan [2][3][4]. Tujuan utama pengujian adalah menemukan *bug* atau kesalahan dalam sistem agar sesuai dengan harapan, serta memastikan kualitas, keandalan, validasi, dan verifikasi sistem [9][10].

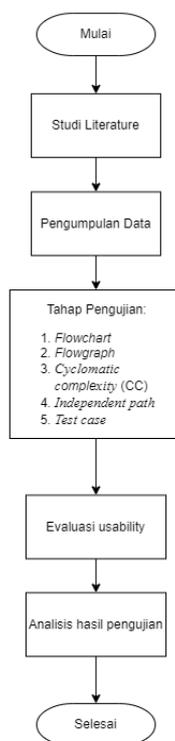
Secara umum, pengujian memiliki tiga metode yang terdiri dari: *black box*, *white box*, serta *grey box*. Metode pengujian *black box* adalah teknik pengujian yang memfokuskan pada pemeriksaan *input* dan *output* tanpa memerlukan pengetahuan tentang kode sumber aplikasi [3][4]. Metode *grey box* adalah kombinasi antara *black box* dan *white box*, di mana penguji harus memiliki pemahaman parsial terhadap struktur internal [5][6]. Sementara itu, metode pengujian *white box* adalah metode yang digunakan untuk menguji logika internal dan struktur kode aplikasi [4][5][11]. Metode *white box testing* merupakan pengujian dengan melakukan unit test yang didasarkan pada penelitian serta analisis kode program untuk memeriksa kesalahan yang terjadi [12]. Pada metode ini, penguji perlu memiliki pengetahuan menyeluruh tentang kode sumber, menganalisis kode program, dan merancang kasus uji berdasarkan struktur kode program tersebut [6].

Pengujian kotak putih atau *white box* memiliki beragam teknik, seperti pengujian loop, aliran data, dan aliran kontrol [6]. Teknik *basis path testing* menghasilkan jumlah kasus uji yang memberikan cakupan uji yang lebih komprehensif [5][15]. Teknik ini membantu perancangan kasus uji untuk mengukur kompleksitas logika dalam desain prosedural dan menggunakan ukuran ini untuk memperkirakan jalur eksekusi dasar. *Basis path testing* melibatkan penggunaan notasi *flowgraph*, *cyclomatic complexity* (CC), *independent path*, dan *test case* untuk menghasilkan urutan pengujian berdasarkan jalur independen yang telah diidentifikasi. Setiap *test case* membutuhkan definisi output yang diharapkan [7].

Meskipun teknik *basis path* sangat kuat dalam mendeteksi kesalahan logika di jalur yang dipilih untuk diuji, teknik ini memiliki keterbatasan dalam mengatasi bug atau anomali yang muncul di luar jalur-jalur yang diuji atau yang terjadi akibat interaksi kompleks antar-modul dalam sistem. Oleh karena itu, teknik ini mungkin tidak selalu memadai untuk memastikan kualitas dan kestabilan seluruh sistem, terutama dalam konteks pengujian integrasi dan sistem secara keseluruhan. Namun, kelebihan dari penggunaan teknik ini adalah kemampuannya untuk menghapus bagian kode tersembunyi yang tidak diperlukan, melakukan pengujian mendalam dengan mengeksplorasi semua bagian dari struktur, sehingga dapat mengoptimalkan kode program agar sistem dapat berjalan lancar dan terstruktur [17].

Berdasarkan penjelasan tersebut, penulis memilih teknik *basis path* dari metode *white box testing* sebagai pendekatan utama yang akan digunakan untuk menguji Sistem Informasi Perencanaan dan Penganggaran Responsif (Siperansinder). Tujuan penelitian ini adalah untuk menilai keberhasilan pengembangan aplikasi dan mengidentifikasi potensi kesalahan, error, serta ketidaksesuaian dalam struktur internal dan kode programnya. Penelitian ini juga akan memberikan rekomendasi perbaikan berdasarkan hasil yang telah diperoleh serta mengevaluasi sejauh mana suatu produk, aplikasi, atau sistem mudah digunakan oleh pengguna akhir yang belum pernah dilakukan oleh penelitian lain yang hanya menggunakan metode *white box* tanpa melihat interaksi pengguna dengan sistem. Dengan demikian, Penelitian ini tidak hanya mengkonfirmasi efektivitas *white box testing*, tetapi juga menyoroti pentingnya mengevaluasi aspek *usability* untuk meningkatkan kualitas perangkat lunak secara keseluruhan.

II. METODE PENELITIAN



Gambar. 1. Tahapan penelitian

A. Studi Literatur

Tahap awal sebelum penelitian ini dilakukan adalah melakukan studi literature untuk melakukan pemahaman mendalam terkait metode pengujian yang dilakukan. Jurnal ilmiah, makalah, artikel, dan buku menjadi rujukan utama dalam studi literatur untuk memahami metode pengujian *white box* dan *teknik basis path*.

B. Pengumpulan Data

Kemudian melakukan pengumpulan data untuk bahan penelitian. Dalam penelitian ini, menggunakan data kualitatif, yaitu berupa kode-kode program dari sistem informasi perencanaan dan penganggaran responsive gender yang diperoleh dari hasil wawancara mendalam bersama pengembang sistem di Diskominfo Kabupaten Jombang dan akan digunakan sebagai bahan untuk dianalisis. Pada penelitian yang dilakukan ini, pengujian dilakukan pada fungsi – fungsi yang berekstensi php menerapkan CRUD. Fungsi – fungsi yang akan diuji tersebut berupa tambah berkas, lihat berkas, edit berkas, dan hapus berkas yang beralokasi di file *controller*.

C. Melakukan Pengujian

Pada tahap ini merupakan tahapan dilakukannya pengujian, pengujian dilakukan dengan menggunakan metode *white box testing* dengan tekniknya yaitu *basis path*. Teknik ini berfokus pada pengujian jalur independen atau urutan langkah – langkah eksekusi yang unik dalam sebuah program, di mana urutan tersebut tidak sepenuhnya sama dengan urutan langkah-langkah pada jalur lainnya. Hal ini untuk memastikan bahwa setiap jalur yang diuji yang mungkin dilalui program selama eksekusi diuji setidaknya sekali berdasarkan jalur yang ditentukan [13].

Tahapan dari teknik yang digunakan ini melibatkan penggunaan notasi *flowgraph*, *cyclomatic complexity (CC)*, *independent path*, dan *test case* untuk menentukan urutan pengujian berdasarkan jalur independen yang telah diidentifikasi.

1. Flowchart

Kode program yang telah didapatkan, selanjutnya dianalisis dan digambarkan kedalam bentuk *flowchart*. *Flowchart* adalah gambar yang menunjukkan langkah-langkah untuk menyelesaikan suatu masalah, menggunakan simbol-simbol khusus. Gambar ini membantu dalam memahami alur berpikir dalam sebuah program [16].

2. Flowgraph

Flowgraph (grafik alir) adalah notasi sederhana yang menggambarkan aliran kontrol dalam struktur program

[14][15]. Proses Pembuatan flowgraph dapat dilakukan melalui dua pendekatan, yaitu dengan menganalisis kode sumber atau dengan memodifikasi flowchart sistem yang sudah ada. Notasi yang digunakan dalam flowgraph terdiri dari *node* yang diwakili oleh lingkaran dan *edge* yang diwakili oleh panah. [14]

3. Perhitungan *cyclomatic complexity* (CC)

Nilai dari *Cyclomatic complexity* (CC) digunakan untuk menunjukkan berapa banyak jalur berbeda yang bisa dilalui program. *Cyclomatic Complexity* (CC) adalah metrik yang dipakai untuk mengukur kompleksitas logika kontrol suatu program. Berikut cara menghitung *Cyclomatic Complexity* (CC):

$$VG = (E - N) + 2 \quad (1)$$

Keterangan :

$V(G)$ = *Cyclomatic Complexity* (CC)

E = Total *edge* (garis) yang ada di *flowgraph*

N = Total *nodes* (lingkaran) yang ada di *flowgraph*

Tingkat kompleksitas suatu program dapat diukur dengan nilai *Cyclomatic Complexity* (CC). Semakin banyak jalur eksekusi yang mungkin dalam sebuah program, semakin tinggi nilai CC-nya. Di bawah ini merupakan tabel 1 Hubungan antara CC dengan tipe prosedur dan tingkat resiko di bawah ini.

TABEL I
 HUBUNGAN ANTARA CC DENGAN TIPE PROSEDUR DAN TINGKAT RESIKO

Nilai CC	Tipe Prosedur	Tingkat Resiko
1 - 4	Prosedurnya Sederhana	Rendah
5 - 10	Prosedurnya terstruktur dengan baik serta konsisten	Rendah
11 - 20	Prosedur yang memiliki tingkat kompleksitas lebih tinggi	Menengah
21 - 50	Prosedur yang rumit dan kritis	Tinggi
>50	Prosedur yang mudah terjadi kesalahan, sangat merepotkan, dan sulit divalidasi.	Sangat Tinggi

Prosedur dengan nilai CC tinggi memiliki kemungkinan terjadinya tingkat resiko kesalahan tinggi juga. Sebaliknya prosedur dengan nilai CC rendah mengidentifikasi bahwa prosedur memiliki tingkat resiko kesalahan yang rendah. Memahami dan mengelola nilai CC termasuk bagian penting karena dapat membantu dalam mengidentifikasi bagian kode yang mungkin memerlukan perhatian ekstra atau penyederhanaan untuk meningkatkan kualitas perangkat lunak secara keseluruhan.

4. *Test case*

Setelah mendapatkan jalur independen dari hasil perhitungan CC yang didapat, jalur independen ini adalah sekumpulan perintah dalam kode yang bisa dijalankan dengan satu cara tanpa ada jalur lain yang bisa diwakili atau dicapai dengan menggabungkan jalur lainnya. Selanjutnya menyusun skenario *test case*/ kasus uji untuk memastikan apakah jalur dari *source code* tersebut valid atau sesuai. Skenario pengujian yang telah disusun, selanjutnya dijalankan untuk melakukan pengujian. Berikut merupakan *test case* yang dibuat untuk menjalankan pengujian.

D. *Evaluasi Usability*

Setelah melakukan pengujian *white box* kemudian dianalisis hasilnya. Kemudian untuk memastikan sistem ini dapat berjalan sesuai dengan keinginan pengguna, maka dilakukan pengujian dengan menggunakan *usability testing*. *Usability testing* merupakan metode pengujian yang digunakan untuk mengevaluasi sejauh mana suatu produk, aplikasi, atau sistem mudah digunakan oleh pengguna akhir. Pada tahap ini menggunakan metode *system usability scale* (SUS). *System Usability Scale* (SUS) adalah metode yang digunakan untuk mengukur kegunaan (*usability*) suatu produk atau sistem.

Berikut ini adalah sepuluh pertanyaan yang akan diberikan pada responden yang disajikan pada Tabel II daftar pertanyaan *usability*. Setiap pertanyaan dalam instrumen memiliki skala jawaban dari 1 hingga 5. Skala 1 menunjukkan sangat tidak setuju (STS), skala 2 berarti tidak setuju (TS), skala 3 menandakan netral (N), skala 4 berarti setuju (S), dan skala 5 menunjukkan sangat setuju (SS).

TABEL II
 DAFTAR PERTANYAAN *USABILITY* MENGGUNAKAN METODE SUS

No	Pertanyaan
1	Saya berpikir bahwa saya akan menggunakan sistem ini lebih sering.
2	Saya merasa sistem ini tidak perlu dibuat semenarik atau serumit itu.
3	Menurut saya, Sistem ini cukup mudah untuk digunakan.
4	Saya pikir saya akan membutuhkan bantuan teknis untuk bisa menggunakan sistem ini.
5	Saya menemukan bahwa fungsi-fungsi di sistem ini terintegrasi dengan baik.
6	Saya merasa sistem ini memiliki banyak ketidaksesuaian.
7	Saya rasa kebanyakan orang akan belajar menggunakan aplikasi ini dengan cepat.
8	Saya merasa sistem ini membingungkan
9	Saya merasa tidak ada kendala ketika menggunakan sistem ini.
10	Saya harus mempelajari banyak hal sebelum saya bisa mulai menggunakan aplikasi ini.

E. Analisis Hasil Pengujian

Hasil pengujian *white box* dan *usability* akan dikumpulkan dan dianalisis. Rekomendasi perbaikan akan disusun berdasarkan temuan masalah yang terungkap melalui kedua metode pengujian ini.

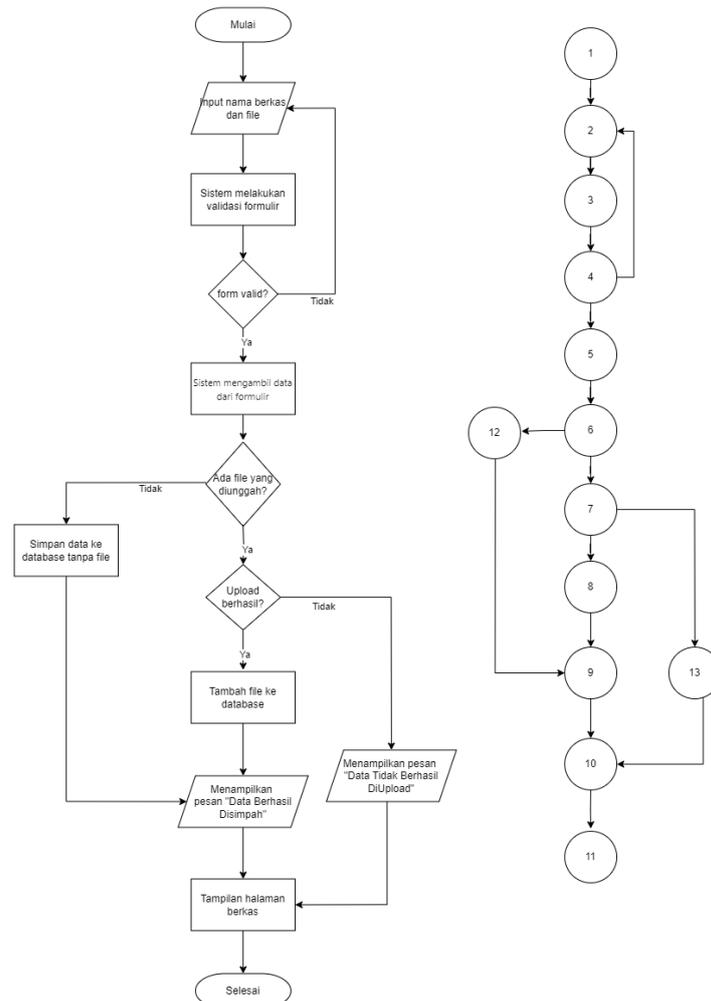
III. HASIL DAN PEMBAHASAN

A. Pengujian *white box*

Pada penelitian yang dilakukan ini, fungsi – fungsi yang akan diuji tersebut berupa tambah berkas, lihat berkas, edit berkas, dan hapus berkas yang beralokasi di file *controller*. Berikut merupakan penelitian yang dilakukan:

1) Fungsi Tambah Berkas

Fungsi tambah berkas digunakan untuk fitur menambahkan berkas yang dibutuhkan pada masing-masing dinas. Berikut merupakan alur dari fitur tambah berkas.



Gambar. 2. Flowchart dan flowraph fungsi tambah berkas

Pada gambar 2. *Flowchart* dan *flowgraph* merupakan fungsi tambah berkas gambar 2 *flowgraph* tambah berkas adalah hasil konversi gambar 2. *Flowchart* tambah berkas. Berdasarkan gambar 2. *Flowchart* dan *flowgraph* fungsi tambah berkas didapatkan *node* berjumlah 13 dan *edge* berjumlah 15. Setelah diketahui jumlah *edge* dan *node* dari *flowgraph* tersebut, maka dilakukan perhitungan *cyclomatic complexity* (CC) sebagai berikut:

$$\begin{aligned} V(G) &= (E - N) + 2 \\ &= (15 - 13) + 2 \\ &= 4 \end{aligned}$$

Dari hasil perhitungan *cyclomatic complexity* (CC) di atas, didapatkan 4 *independent path* atau jalur independen, yakni sebagai berikut:

1. Jalur 1: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11
2. Jalur 2: 1 – 2 – 3 – 4 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11
3. Jalur 3: 1 – 2 – 3 – 4 – 5 – 6 – 12 – 9 – 10 – 11
4. Jalur 4: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 13 – 10 – 11

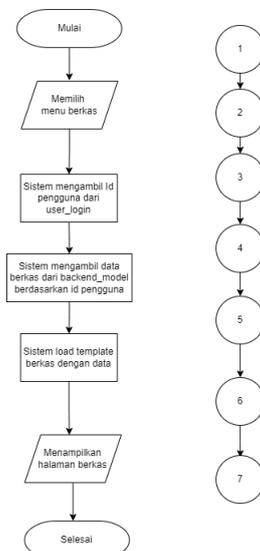
Hasil yang didapat ber-dasarkan tabel 1 Hubungan antara CC dengan tipe prosedur dan tingkat resiko menunjukkan fungsi tambah berkas memiliki tingkat resiko rendah dan tipe prosedurnya sederhana. Langkah selanjutnya adalah menjalankan *test case* untuk mengetahui apakah jalur dari program dari fungsi tambah berkas dapat berjalan dengan sesuai atau valid

TABEL III
 PENGUJIAN FUNGSI TAMBAH BERKAS

Jalur	Kegiatan	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1	Menginput nama berkas/ <i>tittle</i> dan tambahkan berkas yang valid	Sistem akan mengkonfirmasi keberhasilan dengan menampilkan pesan "Data berhasil disimpan" serta mengarahkan pengguna pada halaman berkas	Sistem mengkonfirmasi dan menampilkan pesan "Data berhasil disimpan" serta mengarahkan pengguna pada halaman berkas	Valid
2	Menginput nama berkas/ <i>tittle</i> dan tambahkan berkas yang valid	Sistem akan melakukan validasi dan menampilkan pesan bahwa form harus diisi serta tetap berada di halaman tambah berkas	Sistem menampilkan pesan bahwa form harus diisi serta tetap berada di halaman tambah berkas	Valid
3	<i>Input</i> nama berkas/ <i>tittle</i> yang valid dan kosongkan file	Sistem menyimpan data tanpa file serta menampilkan pesan "Data berhasil disimpan" dan menampilkan halaman berkas	Sistem menyimpan data tanpa file serta menampilkan pesan "Data berhasil disimpan" kemudian menampilkan halaman berkas	Valid
4	<i>Input</i> nama berkas/ <i>tittle</i> valid dan masukkan file yang tidak valid (contoh .txt)	Sistem tidak dapat memproses, serta menampilkan pesan "Data Tidak Berhasil DiUpload" dan diarahkan kembali ke halaman berkas	Sistem tidak dapat memproses.	Valid

2) Fungsi Lihat Berkas

Fungsi lihat berkas OPD digunakan untuk memanggil data berkas lalu menampilkannya pada halaman berkas. Berikut merupakan alur dari fitur lihat berkas



Gambar. 3. *Flowchart* dan *flowgraph* fungsi lihat berkas

Pada gambar 3. *Flowchart* dan *flowgraph* merupakan fungsi lihat berkas. Berdasarkan gambar 3. *Flowchart* dan *flowgraph* fungsi lihat berkas, didapatkan *node* berjumlah 7 dan *edge* berjumlah 6. Setelah diketahui jumlah *edge* dan *node* dari *flowgraph* tersebut, maka dilakukan perhitungan *cyclomatic complexity* (CC) sebagai berikut:

$$\begin{aligned} V(G) &= (E - N) + 2 \\ &= (6 - 7) + 2 \\ &= 1 \end{aligned}$$

Dari hasil perhitungan *cyclomatic complexity* (CC) di atas, didapatkan 1 *independent path* atau jalur independen, yakni sebagai berikut:

1. Jalur 1: 1 – 2 – 3 – 4 – 5 – 6 – 7

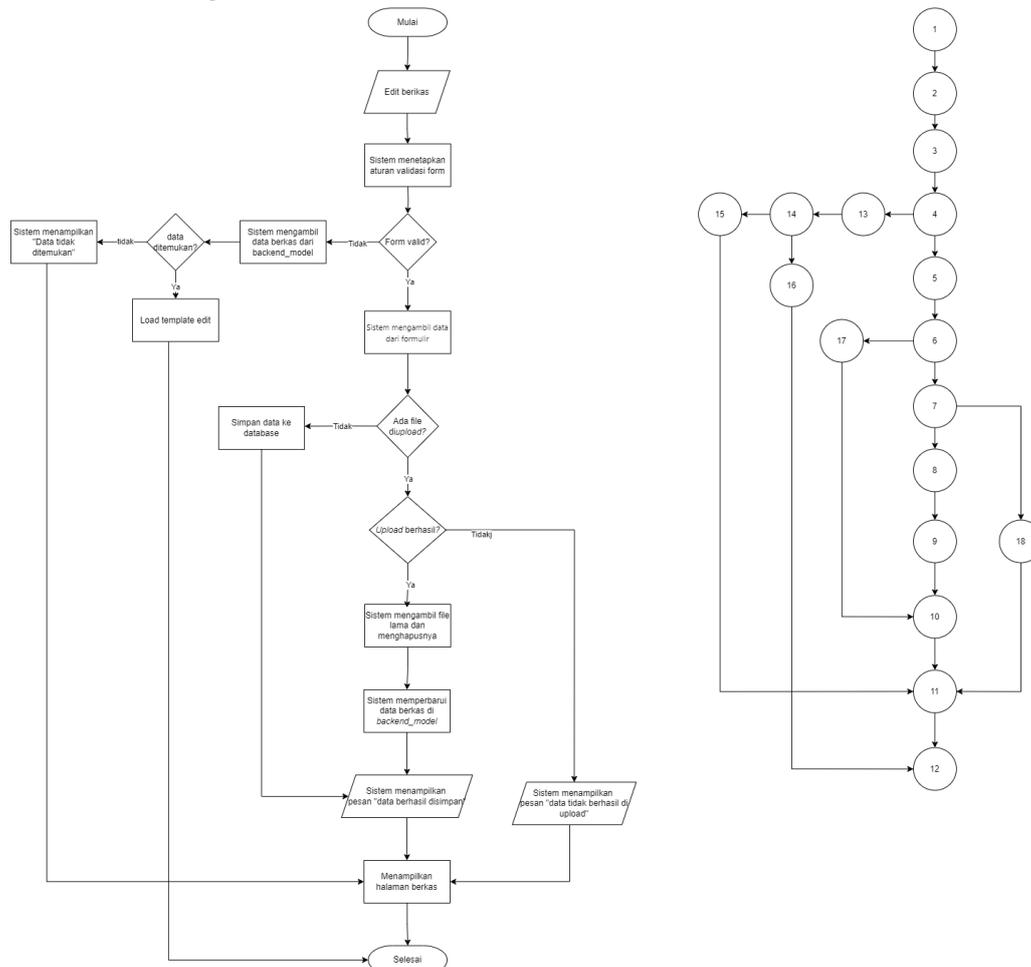
Berdasarkan perhitungan CC, hasil yang didapat berdasarkan tabel 1 Hubungan antara CC dengan tipe prosedur dan tingkat resiko menunjukkan fungsi lihat berkas memiliki tingkat resiko rendah dan tipe prosedurnya sederhana. Langkah selanjutnya adalah menjalankan *test case* untuk mengetahui apakah jalur dari program dari fungsi lihat berkas dapat berjalan dengan sesuai atau valid.

TABEL IV
 PENGUJIAN FUNGSI LIHAT BERKAS

Jalur	Kegiatan	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1	Klik menu berkas	Sistem menampilkan halaman berkas beserta datanya	Sistem menampilkan halaman berkas beserta datanya	Valid

3) Fungsi Edit Berkas

Fungsi edit berkas OPD digunakan untuk fitur mengedit berkas yang memungkinkan pengguna untuk mengubah data berkas. Berikut merupakan alur dari fitur tambah berkas



Gambar 4. *Flowchart* dan *flowgraph* fungsi edit berkas

Pada gambar 4. *Flowchart* dan *flowgraph* merupakan fungsi edit berkas gambar 4 *flowgraph* edit berkas adalah hasil konversi gambar 4. *Flowchart* edit berkas. Berdasarkan gambar 4. *Flowchart* dan *flowgraph* fungsi edit berkas, didapatkan *node* berjumlah 18 dan *edge* berjumlah 21. Setelah diketahui jumlah *edge* dan *node* dari *flowgraph* tersebut, maka dilakukan perhitungan *cyclomatic complexity* (CC) sebagai berikut:

$$\begin{aligned} V(G) &= (E - N) + 2 \\ &= (21 - 18) + 2 \\ &= 5 \end{aligned}$$

Dari hasil perhitungan *cyclomatic complexity* (CC) di atas, didapatkan 5 *independent path* atau jalur independen, yakni sebagai berikut:

1. Jalur 1: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12
2. Jalur 2: 1 – 2 – 3 – 4 – 13 – 14 – 16 – 12
3. Jalur 3: 1 – 2 – 3 – 4 – 13 – 14 – 15 – 11 – 12
4. Jalur 4: 1 – 2 – 3 – 4 – 5 – 6 – 17 – 10 – 11 – 12
5. Jalur 5: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 18 – 11 – 12

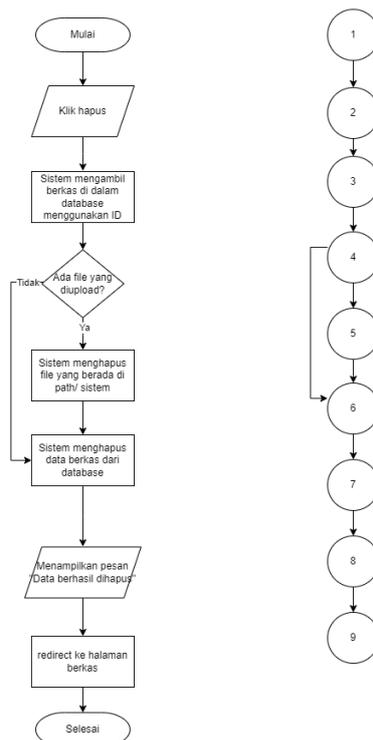
Setelah diketahui *independent path*, selanjutnya melakukan perbandingan dengan tabel 1 Hubungan antara CC dengan tipe prosedur dan tingkat resiko berdasarkan hasil perhitungan *cyclomatic complexity* (CC). Hasil yang didapat berdasarkan tabel 1 Hubungan antara CC dengan tipe prosedur dan tingkat resiko menunjukkan fungsi edit berkas memiliki tingkat resiko rendah dan tipe prosedurnya Prosedurnya terstruktur dengan baik serta konsisten. Langkah selanjutnya adalah menjalankan *test case* untuk mengetahui apakah jalur dari program dari fungsi edit berkas dapat berjalan dengan sesuai atau valid.

TABEL V
 PENGUJIAN FUNGSI EDIT BERKAS

Jalur	Kegiatan	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1	Mengubah nama berkas/ <i>tittle</i> dan tambahkan berkas yang valid	Sistem akan menampilkan pesan “Data berhasil Diedit” serta mengarahkan pengguna pada halaman berkas	Sistem menampilkan pesan “Data berhasil Diedit” serta mengarahkan pengguna pada halaman berkas	Valid
2	Akses halaman edit erkas dengan ID yang valid dan buat form menjadi kosong lalu klik submit	Sistem akan melakukan validasi dan menampilkan pesan bahwa form harus diisi serta tetap berada di halaman edit matriks	Sistem akan melakukan validasi dan menampilkan pesan bahwa form harus diisi serta tetap berada di halaman edit matriks	Valid
3	Ubah ID edit berkas menjadi dengan tidak valid pada URL dan buat form menjadi kosong lalu klik submit	Sistem menampilkan pesan “Data Tidak Ditemukan”	Sistem tidak memproses	Valid
4	Mengubah nama berkas/ <i>tittle</i> dan kosongkan file	Sistem akan menampilkan pesan “Data berhasil Diedit” serta mengarahkan pengguna pada halaman berkas	Sistem menampilkan pesan “Data berhasil Diedit” serta mengarahkan pengguna pada halaman berkas	Valid
5	Mengubah nama berkas/ <i>tittle</i> dan masukkan file yang tidak valid (contoh .txt)	Sistem tidak dapat memproses, serta menampilkan pesan "Data Tidak Berhasil DiUpload" dan diarahkan kembali ke halaman berkas	Sistem tidak dapat memproses.	Valid

4) Fungsi Hapus Berkas

Fungsi hapus berkas digunakan untuk menghapus data berkas di dalam database dan tabel berkas pada tampilan pengguna sesuai dengan permintaan pengguna. Dari kode program hapus berkas yang telah didapatkan, selanjutnya dibuat kedalam bentuk *flowchart* dan kemudian dikonversikan kedalam bentuk *flowgraph*, seperti yang terlihat pada gambar 5 *flowchart* dan *flowgraph* fungsi hapus berkas.



Gambar. 5. Flowchart dan flowgraph fungsi hapus berkas

Pada gambar 5. Flowchart dan flowgraph merupakan fungsi hapus berkas gambar 5 flowgraph hapus berkas adalah hasil konversi gambar 5. Flowchart hapus berkas. Berdasarkan gambar 5. Flowchart dan flowgraph fungsi hapus berkas, didapatkan node berjumlah 9 dan edge berjumlah 9. Setelah diketahui jumlah edge dan node dari flowgraph tersebut, maka dilakukan perhitungan cyclomatic complexity (CC) sebagai berikut:

$$\begin{aligned}
 V(G) &= (E - N) + 2 \\
 &= (9 - 9) + 2 \\
 &= 2
 \end{aligned}$$

Dari hasil perhitungan cyclomatic complexity (CC) di atas, didapatkan 2 independent path atau jalur independen, yakni sebagai berikut:

1. Jalur 1: 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9
2. Jalur 2: 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9

Setelah diketahui independent path, selanjutnya melakukan perbandingan dengan tabel 1 Hubungan antara CC dengan tipe prosedur dan tingkat resiko berdasarkan hasil perhitungan cyclomatic complexity (CC). Hasil yang didapat berdasarkan tabel 1 Hubungan antara CC dengan tipe prosedur dan tingkat resiko menunjukkan fungsi hapus berkas memiliki tingkat resiko rendah dan tipe prosedurnya sederhana. Langkah selanjutnya adalah menjalankan test case untuk mengetahui apakah jalur dari program dari fungsi hapus berkas dapat berjalan dengan sesuai atau valid.

TABEL VI
 PENGUJIAN FUNGSI HAPUS BERKAS

Jalur	Kegiatan	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1	Memilih icon hapus berkas yang terdapat file	Sistem menghapus berkas dan menampilkan pesan serta mengarahkan pengguna halaman berkas	Sistem menampilkan pesan serta mengarahkan pengguna halaman berkas	Valid
2	Memilih icon hapus berkas yang tidak terdapat file	Sistem menghapus berkas dan menampilkan pesan serta mengarahkan pengguna halaman berkas	Sistem menampilkan pesan serta mengarahkan pengguna halaman berkas	Valid

Berdasarkan hasil pengujian white box dengan teknik basis path yang telah dilakukan, dari 4 fungsi yang diuji, satu fungsi dengan tipe prosedurnya terstruktur dengan baik serta konsisten, sedangkan 3 fungsi lainnya prosedurnya sederhana dan juga rendah risiko kesalahan. Persentase yang didapat berdasarkan kategorinya adalah 25% tipe prosedurnya terstruktur dengan baik serta konsisten dan 75% tingkat prosedur rendah. Secara keseluruhan,

sistem ini dinilai memiliki risiko kesalahan prosedur yang rendah. Dari hasil perhitungan ini, dapat diprediksi bahwa kode program yang ada pada *controller* sistem informasi ini memiliki tingkat risiko terhadap cacat atau error yang rendah karena tipe prosedur kode programnya yang sederhana. Kemudian hasil analisis yang dilakukan pada 12 jalur dalam 4 fungsi menghasilkan hasil yang valid dari sisi logika internal sistem.

B. Evaluasi usability

Untuk memastikan bahwa sistem dapat berfungsi dengan baik serta memenuhi harapan pengguna dan standar kualitas, maka juga dilakukan pengujian dengan melakukan *usability testing* dengan metode *system usability scale* (SUS). Pada tahapan ini pertanyaan dibagikan kepada responden melalui kuisioner dengan 15 responden. Berikut data hasil jawaban responden yang disajikan pada tabel VII data hasil jawaban responden.

TABEL VII
 DATA HASIL JAWABAN RESPONDEN

No	Responden (R)	Pertanyaan (P)									
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	R1	4	4	4	2	3	4	2	3	2	4
2	R2	5	4	4	4	3	4	3	3	3	4
3	R3	4	4	3	4	3	5	2	2	3	3
4	R4	4	3	4	3	2	4	3	4	2	4
5	R5	3	4	3	4	4	4	4	4	3	4
6	R6	3	3	5	4	2	4	3	3	3	3
7	R7	5	5	4	4	2	4	3	3	3	3
8	R8	5	4	3	4	2	4	3	3	3	4
9	R9	4	4	3	3	2	4	4	2	3	3
10	R10	4	4	3	3	2	3	3	3	3	4
11	R11	5	3	3	2	3	4	4	4	4	4
12	R12	4	3	3	3	2	3	4	4	3	4
13	R13	4	3	2	3	2	3	4	4	4	4
14	R14	5	4	3	2	2	4	3	4	2	3
15	R15	4	3	3	3	2	4	3	4	2	4

Kemudian dilakukan perhitungan untuk kuisionernya untuk dicari hasil skor SUS. Pada perhitungan ini, untuk mendapatkan skor SUS, setiap jawaban pengguna pada pertanyaan bernomor ganjil dikurangi 1, sedangkan pada pertanyaan bernomor genap, nilai 5 dikurangi dengan jawaban pengguna. Seluruh hasil pengurangan ini kemudian dijumlahkan dan dikalikan dengan 2,5 untuk memperoleh skor akhir SUS. Berikut merupakan data hasil perhitungan SUS yang disajikan pada tabel VIII data hasil perhitungan SUS

TABEL VIII
 DATA HASIL EVALUASI *USABILITY* MENGGUNAKAN SUS

No	Responden (R)	Pertanyaan (P)										Jumlah	Skor (Jumlah × 2,5)
		P1	P2	P3	P4	P5	P6	P7	P8	P9	P10		
1	R1	3	1	3	3	2	1	1	2	1	1	18	45
2	R2	4	1	3	1	2	1	2	2	2	1	19	47,5
3	R3	3	1	2	1	2	0	1	3	2	2	17	42,5
4	R4	3	2	3	2	1	1	2	1	1	1	17	42,5
5	R5	2	1	2	1	3	1	3	1	2	1	17	42,5
6	R6	2	2	4	1	1	1	2	2	2	2	19	47,5
7	R7	4	0	3	1	1	1	2	2	2	2	18	45
8	R8	4	1	2	1	1	1	2	2	2	1	17	42,5
9	R9	3	1	2	2	1	1	3	3	2	2	20	50
10	R10	3	1	2	2	1	2	2	2	2	1	18	45
11	R11	4	2	2	3	2	1	3	1	3	1	22	55
12	R12	3	2	2	2	1	2	3	1	2	1	19	47,5
13	R13	3	2	1	2	1	2	3	1	3	1	19	47,5
14	R14	4	1	2	3	1	1	2	1	1	2	18	45
15	R15	3	2	2	2	1	1	2	1	1	1	16	40
Total skor												685	
Skor rata - rata												45,67	

Berdasarkan hasil evaluasi menggunakan skala *System Usability Scale* (SUS) yang melibatkan 15 responden, diperoleh skor akhir sebesar 45,67 skor ini menempatkan sistem pada kategori "buruk", menandakan bahwa sistem sistem ini sulit digunakan, tidak intuitif, atau tidak efisien. Pengalaman pengguna yang buruk ini disebabkan oleh

berbagai faktor, seperti antarmuka yang rumit, navigasi yang membingungkan, atau fungsionalitas yang tidak sesuai dengan harapan pengguna.

Dari hasil pengujian yang telah dilakukan pada sistem informasi perencanaan dan penganggaran menunjukkan bahwa Siperansinder dengan menggunakan metode *white box testing* dan dipadukan dengan evaluasi *usability* dengan metode SUS menunjukkan bahwa sistem ini sudah berjalan dengan baik berdasarkan logika internal sistem, namun dari sisi pengalaman pengguna, sistem ini masih perlu perbaikan. Meskipun fungsionalitasnya telah bekerja sesuai dengan yang diharapkan, antarmuka yang rumit, navigasi yang membingungkan, dan kesulitan dalam penggunaan sehari-hari menyebabkan skor SUS yang rendah. Ini menunjukkan bahwa sistem masih belum sepenuhnya ramah pengguna dan memerlukan penyempurnaan untuk meningkatkan kemudahan penggunaan dan kepuasan pengguna. Adapun rekomendasi perbaikan sistem yang berupa kritik dan saran untuk perbaikan sistem ini yang disajikan pada tabel IX rekomendasi perbaikan sistem.

TABEL IX
 REKOMENDASI PERBAIKAN SISTEM

No	Rekomendasi Perbaikan
1	Fitur dalam tambah berkas, edit berkas perlu diperbaiki karena terdapat ketidaksesuaian <i>interface</i>
2	Sistem ini harus ditingkatkan performanya agar lebih mudah dan cepat digunakan.
3	Desain <i>interface</i> perlu disesuaikan seperti penggunaan ikon tombol
4	Fitur – fitur perlu ditambahkan dan diperjelas untuk menunjang kebutuhan informasi

Penelitian yang dilakukan sebelumnya pada aplikasi Debt Manager berbasis Android menggunakan metode *white box testing* untuk memeriksa integritas kode¹⁹. Dalam penelitian tersebut, ditemukan bahwa metode *white box* efektif dalam mengidentifikasi kesalahan logika dan memastikan bahwa jalur eksekusi yang kritis telah diuji. Namun, penelitian ini tidak mencakup evaluasi *usability*, sehingga tidak ada analisis mengenai bagaimana pengguna berinteraksi dengan aplikasi atau kesulitan yang mereka hadapi dalam menggunakannya. Dibandingkan dengan penelitian pada aplikasi Debt Manager, penelitian ini memberikan kontribusi tambahan dengan mengintegrasikan *usability* testing. Pengujian ini penting karena, meskipun perangkat lunak memiliki arsitektur kode yang kuat, pengalaman pengguna yang buruk dapat mengurangi kepuasan dan efektivitas penggunaan perangkat lunak tersebut. Dengan demikian, penelitian ini menunjukkan bahwa pendekatan yang menggabungkan *white box* dan *usability testing* dapat memberikan evaluasi yang lebih komprehensif, memastikan bahwa perangkat lunak tidak hanya bebas dari cacat teknis, tetapi juga ramah pengguna.

IV. KESIMPULAN DAN SARAN

Dari hasil pengujian yang telah dilakukan pada Siperansinder, dapat disimpulkan bahwa secara logika internal, sistem ini sudah berfungsi dengan baik berdasarkan hasil pengujian *white box testing* dengan teknik *basis path* yang telah dilakukan dengan menganalisis kode program sistem, menggambarkannya dalam bentuk flowchart, kemudian mengonversinya menjadi *flowgraph* untuk perhitungan *cyclomatic complexity* (CC) dan penentuan jalur independen yang dilanjutkan dengan pembuatan *test case*. Pada pengujian berhasil mengidentifikasi 4 skenario pengujian, di mana sebagian besar skenario (75%) memiliki risiko rendah dengan tipe prosedur yang sederhana. Dari total 12 jalur pengujian, didapatkan hasil yang valid semua yang artinya jalur pengujian berhasil. Namun, dari sisi pengalaman pengguna, sistem ini masih memerlukan perbaikan. Skor SUS yang rendah menunjukkan bahwa meskipun fungsionalitasnya berjalan sesuai harapan, kompleksitas antarmuka, navigasi yang membingungkan, dan kesulitan dalam penggunaan sehari-hari menurunkan kenyamanan pengguna. Oleh karena itu, sistem ini belum sepenuhnya ramah pengguna.

Untuk pengembangan sistem di masa yang akan datang, disarankan untuk dilakukan perbaikan pada antarmuka pengguna untuk membuatnya lebih intuitif dan mudah dinavigasi. Penyederhanaan elemen visual dan penataan ulang menu navigasi bisa membantu meningkatkan kenyamanan pengguna. Kemudian dapat mengikuti rekomendasi yang telah disusun dan disajikan pada tabel IX, untuk meningkatkan sistem berdasarkan kritik dan saran yang telah diidentifikasi.

DAFTAR PUSTAKA

- [1] A. Wijaya and N. Hendrastuty, "Rancang Bangun Sistem Informasi Manajemen Kepegawaian (SIMPEG) Berbasis Web (Studi Kasus: PT Sembilan Hakim Nusantara)," *Jurnal Teknologi Dan Sistem Informasi (JTSI)*, vol. 3, no. 2, pp. 9–17, 2022. [Online]. Tersedia: <http://jim.teknokrat.ac.id/index.php/JTSI>.
- [2] A. Shabika Aqmarina, F. Prima Aditiawan, and H. Endah Wahanani, "Pengujian Sistem Informasi Perpustakaan SMA Wijaya Putra Surabaya Menggunakan Metode Black Box Testing dengan Teknik Equivalence Partitioning dan Boundary Value Analysis," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 1, 2024.
- [3] A. C. Praniffa, A. Syahri, F. Sandels, Ul. Fariha, Q. A. Giansyah, and M. L. Hamzah, "Pengujian Black Box dan *White box* Sistem Informasi Parkir Berbasis Web," *Jurnal Testing dan Implementasi Sistem Informasi*, vol. 1, no. 1, [n.d.].
- [4] F. Agil Sakinah, F. Prima Aditiawan, and A. Lina Nulrlaili, "Pengujian pada Aplikasi Manajemen Aset Menggunakan Black Box Testing," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 3, 2024.
- [5] G. Fadiyah, R. P. Soelsanto, and A. F. Rizana, "Pemetaan Potensi Wisata Alam di Kabupaten Rembang Berbasis Geographic Information System (GIS) Menggunakan Metode Scrum," *eProceedings of Engineering*, vol. 10, no. 3, 2023.
- [6] A. Arifandi, R. Nafal Zulhdi Simamora, G. Azam Janitra, M. Ainull Yaqin, M. Maarifull Hulda, and I. Artikell Abstrak, "Survei Teknik-Teknik Pengujian Software Menggunakan Metode Systematic Literature Review," *ILKOMNIKA: Journal of Computer Science and Applied Informatics*, vol. 4, no. 3, pp. 297–315, 2022. Tersedia: <https://doi.org/10.28926/ilkomnika.v4i3.436>
- [7] R. I. Ndaulmanul, "Pengujian Sistem Informasi Perpustakaan Berbasis Website dengan Basis Path Testing," *Jurnal Teknik*, vol. 6, no. 1, pp. 123–134, 2023. doi: 10.31764/julstek.vXiY.ZZZ.
- [8] A. Yasir, "Sistem Informasi Perpustakaan Berbasis Web Pada Perpustakaan Universitas Dharmawangsa," *Djitelchno: Journal Teknologi Informasi*, vol. 1, no. 2, pp. 36-40, 2020.
- [9] Y. Irawan, "Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus dengan Metode Whitebox Testing," *Jurnal Speled-Sentra Penelitian Engineering dan Edukasi*, vol. 9, CD-ROM.
- [10] L. Siregar, "Review Pengujian Keamanan Perangkat Lunak dalam Software Development Life Cycle (SDLC)," *Jurnal ASEECT*, vol. 1, no. 3, 2020.
- [11] C. T. Pratala, El. M. Asyeln, I. Prayuldi, and A. Saifuldin, "Pengujian *White box* pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, pp. 111, 2020. Tersedia: <https://doi.org/10.32493/informatika.v5i2.4713>
- [12] Y. J. Solissa, F. Pultra, A. N. Pultri, and S. R. C. Nulrsari, "Pengujian *White box* Berbasis Path pada Form Daftar Jobstreet.co.id," *KONSTELLASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 3, no. 2, pp. 353-362, 2023.
- [13] W. A. Nulgraha, "Pengujian *White box* Berbasis Path pada Form Autentikasi Berbasis Mobile," *Jurnal Teknologi dan Sistem Informasi*, vol. 8, no. 2, 2022.
- [14] J. B. L. Siel, I. A. Muldsar, and S. Bahri, "Pengujian *White box* Testing Terhadap Website Room Menggunakan Teknik Basis Path," *KHARISMA Tech*, vol. 17, no. 2, pp. 45-57, 2022.
- [15] Ghibran, M., & Khamaeni, A. L. (n.d.). IMPLEMENTASI *WHITE BOX* TESTING BERBASIS PATH PADA APLIKASI BERBASIS WEB. *Jurnal Siliwangi*, 9(1), 2023.
- [16] N. Kheisya, "Meingenal Flowchart Dan Pseudocodei Dalam Algoritma Dan Peimrograman," 2021. [Online]. Tersedia: <https://osf.io/dq45ei/download>.
- [17] M. F. Londjo, "Implementasi *White box* Testing Dengan Teknik Basis Path Pada Pengujian Form Login, *Jurnal Siliwangi Seri Sains dan Teknologi*", vol. 7, no. 2, 2021.
- [18] D. Wintana, D. Pribadi, dan M. Y. Nurhadi, "Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing," *Jurnal Larik Ladang Artikel Ilmu Komputer*, vol. 2, no. 1, pp. 8-16, 2022.
- [19] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. A. Bakar, N. Novianti, dan R. Ramadan, "Pengujian White-Box pada Aplikasi Debt Manager Berbasis Android," *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, vol. 4, no. 1, pp. 11-22, 2022.
- [20] A. F. Rosyada, I. S. I. Sukirman, M. A. Nur, K. Kristiawan, and A. Saifudin, "Pengujian Sistem Informasi Aplikasi Perpustakaan Basis Website Menggunakan *White box* Testing," **BULLET: Jurnal Multidisiplin Ilmu**, vol. 1, no. 06, pp. 1034-1039.
- [21] A. Setiawan, M. R. D. Ananda, R. Alvario, T. N. Hidayah, dan J. Riyanto, "Pengujian Sistem Informasi Aplikasi Perpustakaan Berbasis Web Di SMAN 1 Gunung Sindur Dengan *White box* Testing," *Scientia Sacra: Jurnal Sains, Teknologi dan Masyarakat*, vol. 2, no. 1, pp. 180-188, 2022.
- [22] M. A. Kosim, S. R. Aji, and M. Darwis, "Pengujian *Usability* Aplikasi Pedulilindungi Dengan Metode System *Usability* Scale (SUS)," *Jurnal Sistem Informasi dan Sains Teknologi*, vol. 4, no. 2, pp. 1-7, 2022.
- [23] A. A. N. H. Susila and D. M. S. Arsa, "Analisis System *Usability* Scale (SUS) dan Perancangan Sistem Self Service Pemesanan Menu di Restoran Berbasis Web," *Majalah Ilmiah UNIKOM*, vol. 21, no. 1, pp. 3-8, 2023.