



Pengujian sistem informasi monitoring dava kebab menggunakan white box testing dengan teknik basis path

Testing of dava kebab monitoring information system using white box testing with basis path technique

Alfian Permana Putra*, Ilyas Nuryasin

* Universitas Muhammadiyah Malang, Jl. Raya Tlogomas No.246, Babatan, Tegalondo, Kec. Lowokwaru, Kota Malang, Jawa Timur, Indonesia

INFORMASI ARTIKEL

Article History:

Submission: 29-04-2024

Revised: 20-05-2024

Accepted: 24-05-2024

Kata Kunci:

White box; pengujian perangkat lunak; sistem informasi; *basis path*; *cyclomatic complexity*

Keywords:

White Box; software testing; information system; *basis path*; *cyclomatic complexity*

* Korespondensi:

Alfian Permana Putra
palfain510@gmail.com

ABSTRAK

Di tengah perkembangan teknologi yang semakin maju, para wirausaha kini dapat lebih meningkatkan efektivitas pekerjaan mereka. Serta penyesuaian kinerja internal dalam usaha mereka dengan memakai sistem informasi monitoring penjualan untuk melakukan pengawasan terhadap penjualan dari tiap outletnya. Salah Satu usaha di bidang Food and Beverages atau FnB Dava Kebab yang mengembangkan sistem informasi baru yakni usaha FnB Dava Kebab. Sistem tersebut berjalan pada sebuah website dan memiliki kegunaan untuk membantu memonitoring penjualan dan barang masuk pada tiap outletnya. Penelitian ini bertujuan untuk menguji sistem informasi monitoring dava kebab menggunakan white box testing dengan teknik basis path. Menggunakan metode white box agar menentukan apakah input dan output memenuhi persyaratan yang diperlukan; sementara basis path testing, tidak mustahil jika perancangan test case untuk mengevaluasi kompleksitas logika dari prosedur rancangan serta memakai evaluasi ini sebagai acuan untuk menentukan kumpulan dasar jalur skenario eksekusi. Dari analisis yang dilakukan, diketahui bahwa fitur login memiliki empat jalur independen, sedangkan fitur pelaporan penjualan memiliki tiga jalur independen. Setiap jalur independen yang diuji menghasilkan skenario yang sesuai dengan harapan, di mana tidak didapatkan adanya kesalahan atau bug pada sistem. Oleh sebab itulah, mampu disimpulkan jika pengujian sistem informasi monitoring dava kebab menggunakan white box testing dengan teknik basis path telah memberikan hasil yang positif dan menunjukkan andalan sistem dalam memenuhi kebutuhan fungsionalnya. Hasil pengujian memiliki kelebihan dikarenakan pada pengujian ini tidak hanya berfokus pada input dan output dari sistem saja, melainkan juga pengujian bagaimana alur program yang ada pada sistem tetapi dalam pengujian juga memiliki kekurangan yakni tidak diujinya lebih jauh terkait penyimpanan sementara atau session yang ada pada sistem tersebut. Hasil pengujian ini dapat digunakan sebagai acuan untuk memastikan bahwa aplikasi yang dikembangkan berfungsi sebagaimana mestinya kemudian hasil dari pengujian ini juga mampu dipergunakan sebagai acuan saat melakukan perbaikan.

ABSTRACT

In the midst of advancing technology, entrepreneurs can now enhance the effectiveness of their work and align internal performance within their businesses by using sales monitoring information systems to oversee sales from each outlet. One of the businesses in the Food and Beverages (FnB)



sector that has developed a new information system is Dava Kebab. This system operates on a website and serves to help monitor sales and incoming goods at each outlet. This study aims to test the Dava Kebab Monitoring Information System using White Box Testing with Basis Path techniques. The White Box method is used to determine whether the input and output meet the necessary requirements, while Basis Path Testing allows for the design of Test Cases to evaluate the logical complexity of the design procedures and use this evaluation as a reference for determining the basic set of execution scenario paths. From the analysis conducted, it was found that the login feature has four independent paths, while the sales reporting feature has three independent paths. Each tested independent path produced scenarios that met expectations, with no errors or bugs found in the system. Therefore, it can be concluded that the testing of the Dava Kebab Monitoring Information System using White Box Testing with Basis Path techniques has yielded positive results and demonstrated the system's reliability in meeting its functional needs. The testing results have advantages as this testing not only focuses on the system's input and output but also on how the program flow within the system operates. However, the testing also has a drawback in that it did not further test temporary storage or sessions within the system. These testing results can be used as a reference to ensure that the developed application functions as intended, and the results can also be used as a reference for making improvements.

1. PENDAHULUAN

Serangkaian tindakan pengawasan pada komponen sistem yang dikenal sebagai proses monitoring dilakukan secara bersamaan untuk memastikan bahwa semua proses dijalankan sesuai dengan protokol yang ditentukan. Hal ini dilakukan sebagai bagian dari implementasi sistem monitoring [1]. Salah satu usaha yang perlu adanya sistem monitoring yakni sebuah usaha Food and Beverages (FnB) yang merupakan bisnis yang berfokus pada penyediaan makanan dan minuman dengan tujuan memenuhi kebutuhan konsumsi masyarakat. Pada sebuah usaha *Food and Beverages* atau FnB yang penjualannya tersebar di berbagai outlet menggunakan sistem informasi monitoring penjualan untuk melakukan pengawasan terhadap penjualan dari tiap outletnya. Di tengah perkembangan teknologi yang semakin maju saat ini para wirausaha kini dapat lebih meningkatkan efektivitas pekerjaan mereka dan penyelarasan kinerja internal dalam usaha mereka [2]. Dalam konteks ini, penelitian telah menyoroti kebutuhan akan sistem informasi yang dapat memfasilitasi pengawasan terhadap berbagai aspek operasional, termasuk penjualan dan stok barang, di berbagai lokasi outlet. Salah Satu usaha di bidang *Food and Beverages* atau FnB yang mengembangkan sistem informasi baru yakni usaha FnB Dava Kebab yang mengembangkan sebuah sistem informasi monitoring baru, sistem ini berjalan pada sebuah website yang dapat diakses pada <http://simodav.site>, dan memiliki kegunaan untuk membantu memonitoring penjualan dan barang masuk pada tiap outletnya. Pengujian terhadap sistem informasi monitoring baru yang dimiliki Dava Kebab merupakan tahapan penting dalam memastikan kualitas sistem ini. Pada saat proses pengembangan suatu sistem informasi baru perlu adanya sebuah pengujian, hal ini dikarenakan dalam pengembangan perangkat lunak sering kali menghadapi sejumlah masalah yang terjadi pada sistem, termasuk *bug* pada sistem [3], sehingga sistem harus diuji untuk mengidentifikasi kesalahan ini.

Dalam System Development Life Cycle (SDLC) dimana istilah tersebut biasa dikenal sebagai siklus pengembangan sebuah sistem [4], tahapan pengujian terhadap sebuah sistem baru menjadi penting karena tahapan pengujian menjadi salah satu komponen penting *Software Development Life Cycle* (SDLC) pengujian perangkat lunak mampu menilai kualitas perangkat lunak serta mengidentifikasi masalah pada sistem baru sehingga kelemahan dapat ditemukan dan diperbaiki [4]. Oleh karena itu, pengujian sistem menjadi tahap kritis dalam siklus pengembangan

perangkat lunak. Berbagai teknik pengujian, termasuk pengujian *White Box*, black box, serta gray box digunakan saat pengujian perangkat lunak [5]. Penelitian ini berkonsentrasi pada pengujian *White Box* Teknik ini memungkinkan pengujian struktur internal dari perangkat lunak, termasuk alur program dan logika kode [6] berbeda dengan pengujian black box yang berfokus pada pengujian fungsional dari hasil input dan output saja [7]. Atau dengan kata lain pengujian *white box* tidak hanya melakukan pengujian terhadap input dan output, melainkan juga mencakup hal hal terkait alur program, kode program, dan juga logic dari program. Dalam pengembangan sebuah sistem pengujian *white box* masih penting karena mencakup pengujian struktural aliran program pengujian *white box* memberi penguji kemampuan untuk memeriksa setiap rute eksekusi program dan memastikan bahwa setiap skenario potensial telah dievaluasi dengan memahami logika internal dan struktur kode [8]. Hal ini meningkatkan pengujian secara keseluruhan dan membantu dalam menemukan dan menyelesaikan kesalahan yang mungkin terkait dengan struktur kontrol dan aliran program. Pada pengujian ini menggunakan salah satu teknik dari *white box* yakni teknik *basis path*, Teknik tersebut dipergunakan agar diketahui kompleksitas alur logika dari program yang diuji [9].

Pada penelitian sebelumnya [8], [10], [11], [12] dilakukan pengujian *white box* dengan menggunakan Teknik *basis path*, yang dimana tahap awal dari pengujian ini membuat sebuah *flowchart* yang dilanjutkan membuat sebuah *flowgraph* beserta menghitung *cyclomatic* yang Dimana hasil dari perhitungan tersebut digunakan untuk menjadi *independent path*, yang digunakan sebagai *test case* untuk menguji program tersebut untuk hasil keseluruhan pengujian dari sistem tersebut dimana hasil pengujian dari tiap penelitian menunjukkan hasil positif dari tiap sistem yang diujikan. Pengujian *white box* dengan menggunakan *basis path* juga digunakan pada penelitian sebelumnya [13] yang digunakan untuk melakukan pada aplikasi cash flow untuk tiap tahapan yang dilakukan pada pengujian ini hampir sama dengan pengujian sebelum sebelumnya, tetapi yang membedakan pengujian ini dengan yang lain yakni pada tahapan *TEST CASE* penelitian ini menggunakan unit test untuk menguji setiap *independent path* dan hasil dari pengujian ini juga tidak ada kesalahan atau bug apapun pada aplikasi yang diuji. Pada penelitian sebelumnya menunjukkan bahwa teknik *basis path* menunjukkan hasil yang positif dalam pengujian sebuah sistem, pada penelitian tersebut juga dapat diketahui bahwa penelitian yang dilakukan sebelumnya pengujian menekankan pengujian berkaitan dengan bagaimana alur program berjalan. Dengan demikian, penelitian ini mampu memberikan kontribusi pada pemahaman kinerja dari sistem informasi monitoring Dava Kebab.

2. METODE

Penelitian pada sistem ini yang menggunakan bahasa pemrograman laravel akan diujikan menggunakan metode *white box testing*, yaitu metode pengujian perangkat lunak pada tingkat aliran kode program, atau pengujian *White Box*, untuk menentukan apakah input dan output memenuhi persyaratan yang diperlukan [6]. Dalam metode *white box* sendiri terdapat beberapa teknik pengujian pengujian, antara lain *loop testing*, *basis path/path testing*, *control flow testing*, serta *data flow testing* [14]. Pada riset ini menggunakan salah satu Teknik terdapat pada metode *White Box* yakni *Basis Path Testing*, teknik ini memungkinkan perancangan *Test Case* untuk mengevaluasi kompleksitas logika dari prosedur rancangan serta memakai evaluasi ini sebagai acuan dengan menentukan kumpulan dasar jalur skenario eksekusi. Penerapan metode *basis path testing* ini dilakukan dengan membangun *flow graph* dari program Laravel yang hendak diuji. Selain itu, dalam mengerjakan *cyclomatic complexity* (CC) serta menyusun Test Scenario, aspek-aspek khusus dari struktur dan fitur-fitur juga dipertimbangkan. Untuk mengidentifikasi jumlah jalur eksekusi, pendekatan pengujian *basis path* memerlukan penghitungan kompleksitas logistik aliran program dan penggunaan ukuran sebagai referensi [15]. Pada *white box testing* memakai *basis path* adanya sejumlah rangkaian yakni saat membuat *flow graph* berasal dari program yang hendak diuji, menghitung *cyclomatic complexity* (CC) serta mengerjakan *test skenario* [8]. Pada penelitian sebelumnya [16] tahapan dalam pengujian telah dilakukan adapun Langkah-langkah proses pengujian *Basis Path* meliputi pembuatan *flowgraph*, *flowchart*, dan perhitungan

cyclomatic complexity. *flowgraph* adalah representasi visual dari alur logika program yang digunakan dalam pengujian *Basis Path*. *Flowgraph* membantu dalam memvisualisasikan aliran kontrol program dengan menggunakan simpul (node) untuk mewakili instruksi-instruksi dalam program dan garis-garis penghubung untuk menunjukkan alur eksekusi antar instruksi. *Flowchart*, di sisi lain, adalah representasi grafis yang lebih umum dari alur logika program. *Flowchart* menggunakan simbol-simbol standar yakni persegi panjang untuk merepresentasikan proses, oval untuk merepresentasikan titik awal atau akhir, dan panah untuk menunjukkan alur arah. Sementara itu, *cyclomatic complexity* adalah metrik perangkat lunak yang dipergunakan untuk mengukur kompleksitas struktural dari program. Perhitungan *cyclomatic complexity* memberikan gambaran tentang jumlah jalur independen yang ada dalam program, yang pada gilirannya membantu dalam menentukan kebutuhan pengujian dan mengevaluasi keseluruhan kompleksitas program secara keseluruhan. Perhitungan ini digunakan untuk menentukan jumlah kemungkinan jalur eksekusi. Metode pengujian *Basis Path* memerlukan penentuan kompleksitas logistik aliran program sebelum menghasilkan kasus uji untuk setiap jalur independen.

Untuk membantu penghitungan kompleksitas dari sistem yang diuji penelitian ini menggunakan teknik *cyclomatic complexity* yang akan menyediakan patokan kuantitatif kompleksitas suatu logika pada program [12]. Jumlah kasus uji paling sedikit yang diperlukan untuk menguji aplikasi menggunakan teknik pengujian *Basis Path* dapat ditemukan menggunakan *cyclomatic complexity* [12]. Pada persamaan 1 merupakan rumus untuk menghitung kompleksitas logika dari sistem.

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

Dimana:

E = jumlah *edges* pada *flowgraph*

N = jumlah *Nodes* pada *flowgraph*

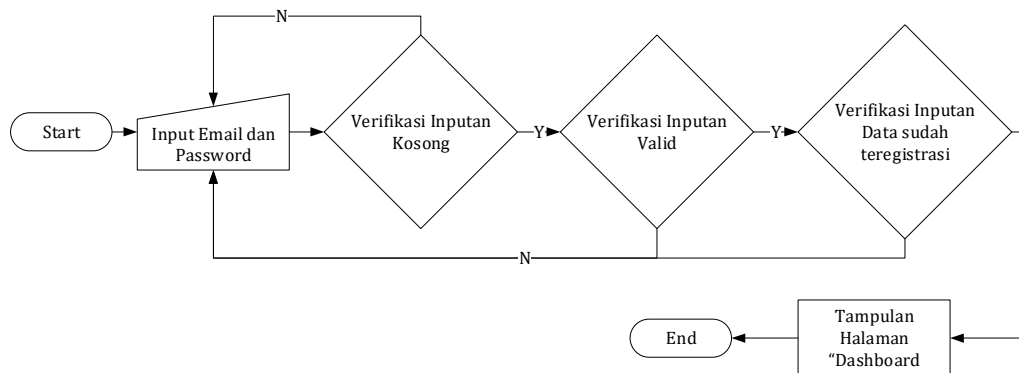
Pada pengujian ini juga menggunakan sebuah *Test Case* untuk menguji output dari tiap alur skenario yang sudah ditetapkan dengan *Cyclomatic Complexity*. Dalam pengujian semua skenario menggunakan *Test Case* tersebut, kode akan diperiksa sekali lagi hingga memenuhi output yang diharapkan apabila output dari *Test Case* yang diperoleh tidak tepat dengan apa yang diharapkan.

3. HASIL DAN PEMBAHASAN

Pada pengujian *White Box* yang menggunakan teknik *Basis Path* dimulai dengan pembuatan *flowchart*, proses pembuatan *flowchart* ini dilakukan dengan cara menganalisis keseluruhan program sehingga dapat menggambarkan bagaimana alur program dari tiap fiturnya. Dalam pengujian ini terdapat 2 fitur yang diujikan yakni Fitur Login dan fitur Pelaporan Penjualan. Berikut merupakan penjelasan dari kedua fitur tersebut:

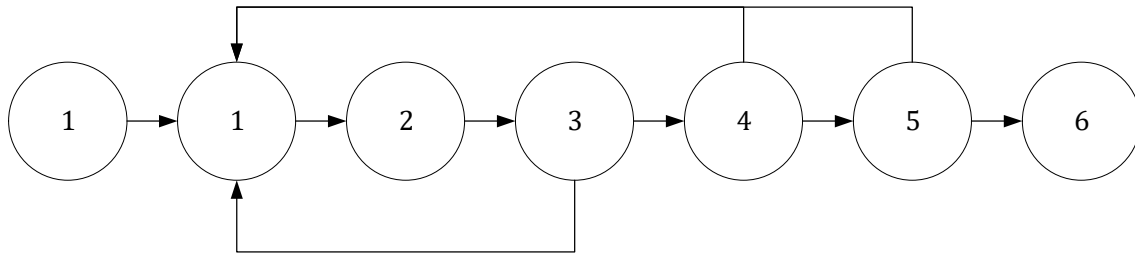
Fitur Login

Fitur Login merupakan sebuah fitur yang digunakan untuk mengontrol akses dari tiap pengguna, dalam prosesnya fitur login melibatkan otentikasi, dimana pengguna yang ingin memiliki akses masuk ke dalam sistem harus memasukkan informasi kredensial yang sesuai.



Gambar 1. Flowchart fungsi login

Dalam **Gambar 1** menunjukkan *flowchart* dari fitur login, dari *flowchart* tersebut dapat diketahui alur program yang berjalan dari fitur tersebut. Alur program fitur login dimulai dari pengisian inputan untuk email dan password dan dilanjutkan adanya 3 validasi yang terdiri dari validasi inputan kosong dan dari *flowchart* tersebut dapat diketahui apabila inputan tidak kosong atau no dalam *flowchart* maka proses dapat dilanjutkan dan yang kedua apabila verifikasi inputan valid dan apabila inputan valid atau yes maka proses dapat dilanjutkan dan yang terakhir verifikasi data sudah teregistrasi dari verifikasi tersebut apabila data sudah teregistrasi atau yes maka proses dapat dilanjutkan dan *flowchart* tersebut diakhiri dengan menampilkan halaman isi dashboard dari web tersebut.



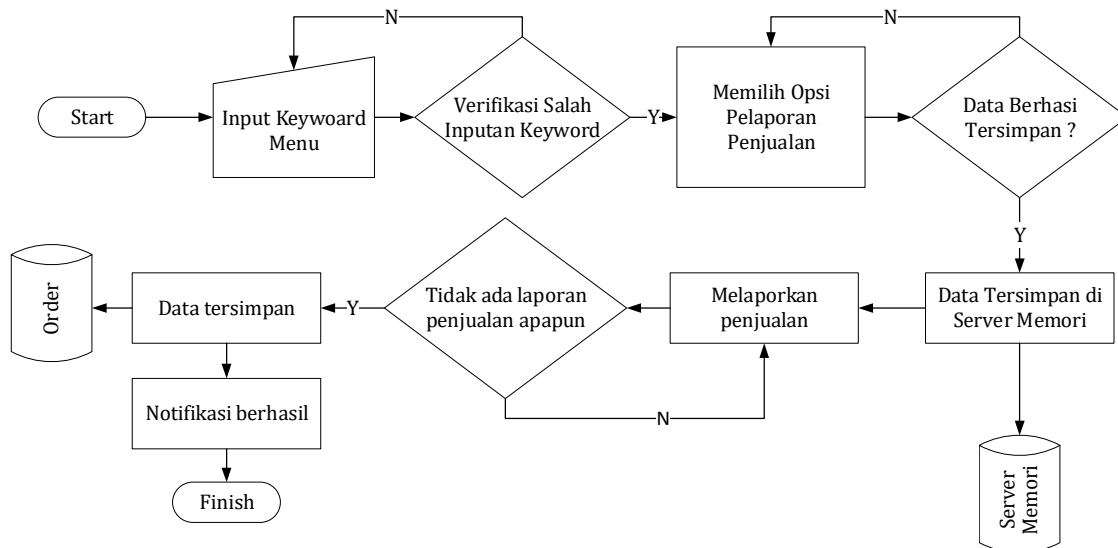
Gambar 2. Flowgraph fitur login

Flowgraph yang menggambarkan prosedur fitur login pada suatu sistem dapat dilihat pada **Gambar 2**. Pada *flowgraph* kita dapat menentukan jumlah *Nodes* (N) dan *Edges* (E) yang terlibat dalam diagram *flowgraph*. *Flowgraph* fitur login pada instance ini terdiri dari total 7 nodes dan 9 edges. Sesuai dengan aturan dari perhitungan kompleksitas menggunakan *cyclomatic complexity*, perhitungan *cyclomatic complexity* memerlukan komputasi berdasarkan jumlah *nodes* dan *edges*. Dengan demikian, kompleksitas dapat dihitung menggunakan aturan yang telah ditetapkan dengan memperoleh nilai E dan N dari *flowgraph* fitur login.

Fitur pelaporan penjualan

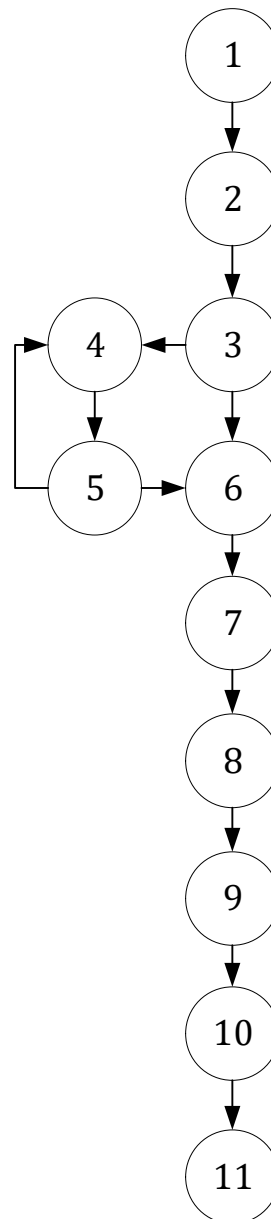
Fitur pelaporan penjualan merupakan sebuah fitur dimana pengguna dapat melaporkan penjualan yang sudah dilakukan di tiap outletnya. Pengguna hanya menekan beberapa button saja dan laporan tersebut akan dapat langsung diakses di website utama

Dari fitur tersebut akan dibuat sebuah *flowchart*, dalam pembuatan sebuah aplikasi *flowchart* memiliki peranan penting karena *flowchart* digunakan untuk menjelaskan logika dari sebuah aplikasi [17]. Gambar 1 dan Gambar 2 menunjukkan *flowchart* dari tiap fungsi yang akan diujikan.



Gambar 3. Flowchart pelaporan penjualan

Pada **Gambar 3** merupakan *flowchart* untuk fitur pelaporan penjualan sama halnya dengan *flowchart* dari fitur login, *flowchart* dari fitur pelaporan penjualan dapat digunakan untuk melihat alur program dari fitur pelaporan penjualan itu sendiri pada *flowchart* ini dapat diketahui awal dari sistem tersebut dimulai untuk mencari menu atau item yang akan dilaporkan dilanjutkan untuk memilih opsi laporan penjualan apakah akan ditambahkan atau dikurangi. Setelah memilih opsi tersebut proses akan dilanjutkan untuk memverifikasi apakah data tersebut sudah dapat tersimpan di session server laravel atau belum, session server sendiri merupakan fitur yang disediakan laravel untuk penyimpanan sementara penyimpanan ini tidak akan masuk ke database yang ada tetapi hanya akan tersimpan di server dari laravel tersebut. Apabila berhasil tersimpan atau yes dalam *flowchart* maka proses akan dilanjutkan dimana user akan melaporkan penjualan harian keseluruhannya untuk dicatat dalam database tetapi sebelum data tersebut dikirimkan ke database akan ada pengecekan apakah di dalam session server sudah terdapat data yang sudah dicatatkan sebelumnya atau belum, jika sudah ada maka data akan tersimpan di database.



Gambar 4. *Flowgraph* Laporan Penjualan

Dari **Gambar 4** yang merupakan *flowgraph* dari fitur laporan penjualan dapat diketahui bahwa Jumlah Edge yang terdapat di *flowgraph* yakni $E = 12$ dan untuk *Nodes* dari *flowgraph* tersebut yakni $N = 11$. Nantinya *Edges* dan *Nodes* tersebut yang akan dihitung menggunakan *Cyclomatic complexity* dan dari perhitungan tersebut, dapat diketahui jalur independen yang digunakan sebagai *test case* dari pengujian ini.

Setelah mengetahui *edges* dan *nodes* dari tiap fitur login dan pelaporan penjualan, maka dalam pengujian ini dapat menghitung nilai CC dari tiap fiturnya. Untuk Perhitungan dari tiap fitur sebagai berikut:

Perhitungan *Cyclomatic complexity* Fitur Login

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

$$V(G) = 9 - 7 + 2$$

$$V(G) = 4$$

Pada fitur login nilai perhitungan *Cyclomatic Complexity* memiliki nilai 4, nilai ini nantinya digunakan juga dapat diketahui bahwa dalam fitur login memiliki 4 jalur independen

Perhitungan *Cyclomatic complexity* Pelaporan Penjualan

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

$$V(G) = 13 - 11 + 2$$

$$V(G) = 4$$

Pada fitur pelaporan penjualan didapatkan nilai 4 dari perhitungan *Cyclomatic Complexity*, dan dari perhitungan tersebut dapat diketahui terdapat 4 jalur independen dalam fitur tersebut. Dari kedua perhitungan tersebut didapatkan jumlah jalur independen dari fitur login dan pelaporan penjualan. Untuk Jalur Independen dari fitur login dan pelaporan penjualan berada pada **Tabel 1** serta **Tabel 2**:

Tabel 1. Jalur independen fitur login

Jalur Independen Fitur Login	
1	1-2-3-4-5-6-7 (User berhasil melakukan login)
2	1-2-3-2-3-4-5-6-7 (User tidak menginputkan apapun di form login)
3	1-2-3-4-2-3-4-5-6-7 (User menginputkan email dan password yang tidak valid/tidak sesuai penulisan)
4	1-2-3-4-5-2-3-4-5-6-7 (User menginputkan email dan password yang belum teregistrasi)

Tabel 2. Jalur independen laporan penjualan

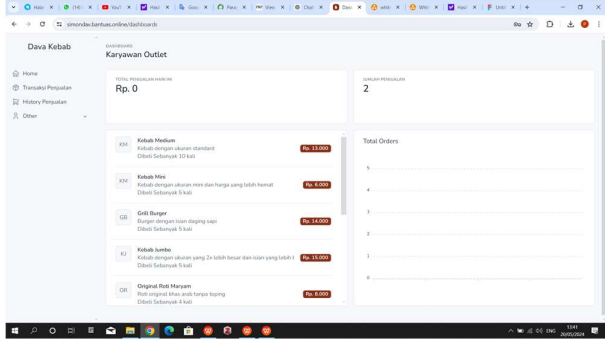
Jalur Independen Fitur Laporan Penjualan	
1	1-2-3-4-5-6-7-8-9-10-11 (User Berhasil Melaporkan Penjualan)
2	1-2-3-2-3-4-5-6-7-8-9-10-11 (User Salah Menginputkan Keyword yang Dimana menu tersebut tidak ada dalam data)
3	1-2-3-4-5-4-5-6-7-8-9-10-11 (Data Yang ada di server session tidak tersimpan)

Pengujian sistem informasi monitoring dava kebab menggunakan *white box testing* dengan teknik *basis path*

Jalur Independen Fitur Laporan Penjualan	
	1-2-3-4-5-6-7-8-7-8-9-10 -11
4	(Tidak ada laporan apapun yang akan dikirimkan ke database di server session)

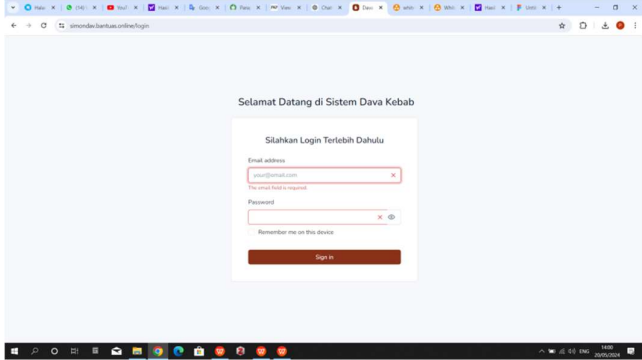
Setelah mendapatkan jalur independen dari kedua fitur tersebut maka selanjutnya dibuat sebuah uji skenario. Pengujian tersebut menggunakan *Test case*, Dimana pengujian ini dilakukan secara manual dengan mengujikan secara langsung ke sistem. *Test case* digunakan untuk menguji apakah skenario dari Jalur Independen yang sudah didapatkan dari tiap fiturnya ini sudah sesuai atau belum dengan sistem, dan apabila hasil pengujian skenario tersebut berbeda dengan output yang sudah ditentukan sebelumnya maka nantinya hal tersebut dapat menjadi acuan perbaikan kode untuk program yang diujikan. *Test case* juga memainkan peran penting dalam memastikan bahwa setiap *independent path* sudah diketahui telah diperiksa secara menyeluruh pada pengujian *white box* memakai teknik *basis path*. *test case* dibuat dengan mencakup setiap Jalur Independen dengan memperhatikan struktur kontrolnya. Untuk *test case* dari kedua fitur dapat dilihat sebagai berikut:

Tabel 3. Test case fitur login path 1

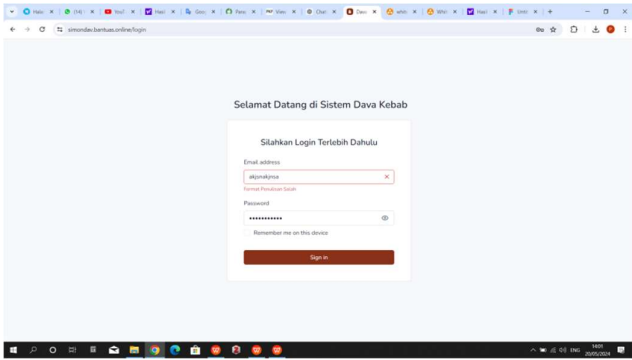
Path	1
Jalur	1-2-3-5-6-7
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input email dan password 3. Verifikasi inputan kosong 4. Verifikasi inputan valid 5. Verifikasi data teregistrasi 6. Tampilkan "Halaman Dashboard" 7. End
Tampilan Pengujian	
Hasil Pengujian	Berhasil

Tabel 4. Test case fitur login path 2

Path	2
Jalur	1-2-3-2-3-4-5-6-7
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input email dan password 3. Vervikasi inputan kosong 2. Input email dan password 3. Verifikasi inputan kosong 4. Verifikasi inputan valid 5. Verifikasi data teregistrasi 6. Tampilkan "Halaman Dashboard" 7. End

Path	2
Tampilan Pengujian	
Hasil Pengujian	Berhasil

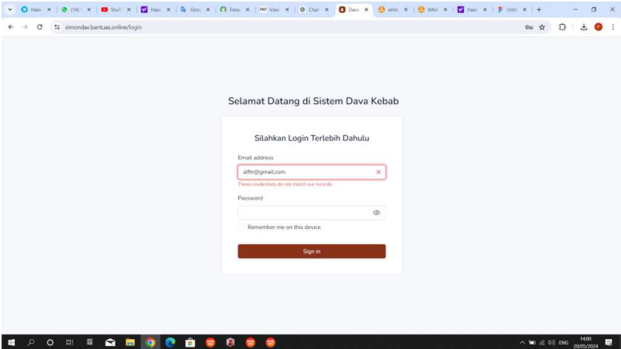
Tabel 5. Test case fitur login path 3

Path	3
Jalur Skenario	<p>1-2-3-4-2-3-4-5-6-7</p> <ol style="list-style-type: none"> 1. Start 2. Input email serta password 3. Verifikasi inputan kosong 4. Verifikasi inputan valid 2. Input email dan password 3. Verifikasi inputan kosong 4. Verifikasi inputan valid 5. Verifikasi data teregistrasi 6. Tampilkan “Halaman Dashboard” 7. End
Tampilan Pengujian	
Hasil Pengujian	Berhasil

Tabel 6. Test case fitur login path 4

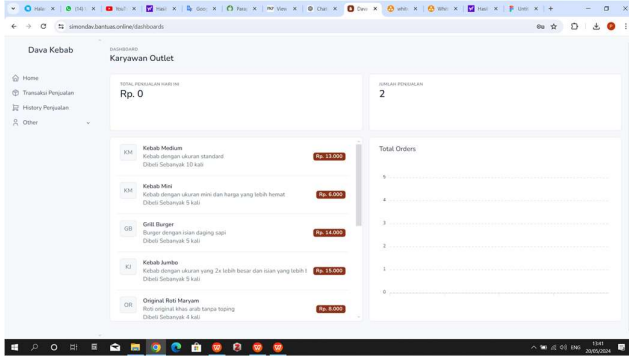
Path	4
Jalur Skenario	<p>1-2-3-4-5-2-3-4-5-6-7</p> <ol style="list-style-type: none"> 1. Start 2. Input email dan password 3. Verifikasi inputan kosong 4. Verifikasi inputan valid 5. Verifikasi data teregistrasi 2. Input email dan password 3. Verifikasi inputan kosong 4. Verifikasi inputan valid 5. Verifikasi data teregistrasi

Pengujian sistem informasi monitoring dava kebab menggunakan *white box testing* dengan teknik *basis path*

Path	4
Tampilan Pengujian	<p>6. Tampilkan "Halaman Dashboard"</p> <p>7. End</p> 
Hasil Pengujian	Berhasil

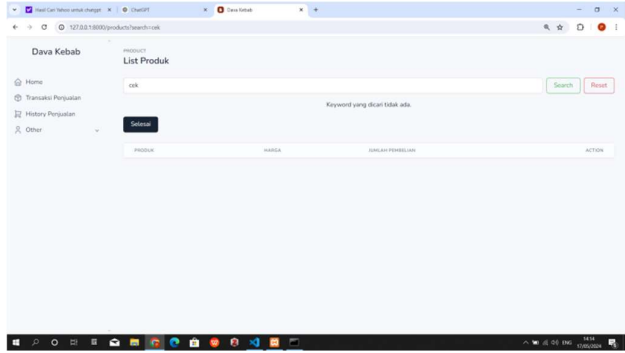
Pada [Tabel 3](#), [Tabel 4](#), [Tabel 5](#), [Tabel 6](#) bisa diketahui bahwa *test case* yang dilakukan pada fitur login sesuai dengan jalur independennya, dan dalam penggunaannya bisa diketahui bahwa dari tiap jalur pengujiannya tidak ditemukan bug atau kesalahan yang terjadi dari program tersebut

Tabel 7. Test case fitur pelaporan penjualan path 1

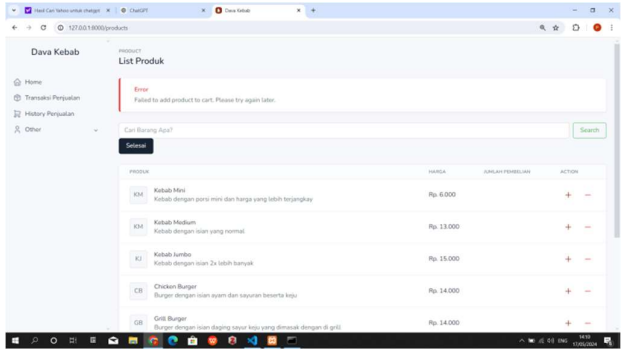
Path	1
Jalur Skenario	<p>1-2-3-4-5-6-7-8-9-10-11</p> <ol style="list-style-type: none"> 1. Start 2. Input Keyword Menu 3. Verifikasi Salah Inputan Keyword 4. Memilih Opsi Pelaporan Penjualan 5. Data Berhasil tersimpan? 6. Data tersimpan di server memori 7. Melaporkan Penjualan 8. Tidak Ada Laporan Penjualan Apapun? 9. Data Tersimpan (Order) 10. Notifikasi Berhasil 11. End
Tampilan Pengujian	
Hasil Pengujian	Berhasil

Tabel 8. Test case fitur pelaporan penjualan path 2

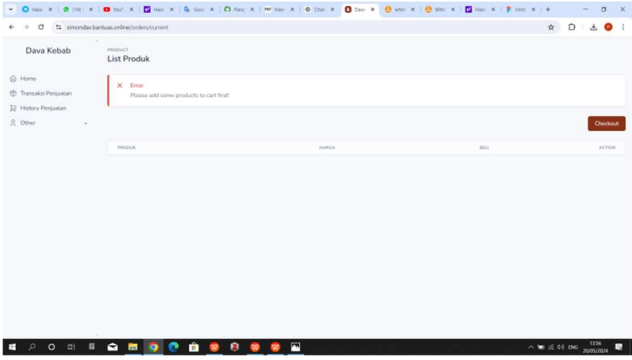
Path	2
Jalur Skenario	<p>1-2-3-2-3-4-5-6-7-8-9-10-11</p> <ol style="list-style-type: none"> 1. Start 2. Input Keyword Menu

Path	2
	3. Verifikasi Salah Inputan Keyword 2. Input Keyword Menu 3. Verifikasi Salah Inputan Keyword 4. Memilih Opsi Pelaporan Penjualan 5. Data Berhasil tersimpan? 6. Data tersimpan di server memori 7. Melaporkan Penjualan 8. Tidak Ada Laporan Penjualan Apapun? 9. Data Tersimpan (Order) 10. Notifikasi Berhasil 11. End
Tampilan Pengujian	
Hasil Pengujian	Berhasil

Tabel 9. Test case fitur pelaporan penjualan path 3

Path	3
Jalur	1-2-3-4-5-4-5-6-7-8-9-10-11
Skenario	1. Start 2. Input Keyword Menu 3. Verifikasi Salah Inputan Keyword 4. Memilih Opsi Pelaporan Penjualan 5. Data Berhasil tersimpan? 4. Memilih Opsi Pelaporan Penjualan 5. Data Berhasil tersimpan? 6. Data tersimpan di server memori 7. Melaporkan Penjualan 8. Tidak Ada Laporan Penjualan Apapun? 9. Data Tersimpan (Order) 10. Notifikasi Berhasil 11. End
Tampilan Pengujian	
Hasil Pengujian	Berhasil

Tabel 10. Test case fitur pelaporan penjualan path 4

Path	4
Jalur	1-2-3-4-5-6-7-8-7-8-9-10 -11
Skenario	<ol style="list-style-type: none"> 1. Start 2. Input Keyword Menu 3. Verifikasi Salah Inputan Keyword 4. Memilih Opsi Pelaporan Penjualan 5. Data Berhasil tersimpan? 6. Data tersimpan di server memori 7. Melaporkan Penjualan 8. Tidak Ada Laporan Penjualan Apapun? 7. Melaporkan Penjualan 8. Tidak Ada Laporan Penjualan Apapun? 9. Data Tersimpan (Order) 10. Notifikasi Berhasil 11. End
Tampilan Pengujian	
Hasil Pengujian	Berhasil

Begitu juga dengan [Tabel 7](#), [Tabel 8](#), [Tabel 9](#), [Tabel 10](#) *test case* yang diujikan memiliki hasil yang positif, pada fitur pelaporan penjualan sesuai dengan jalur independennya. Dalam pengujian sendiri bisa bahwa dari tiap jalur pengujiannya tidak ditemukan bug atau kesalahan yang terjadi dari program tersebut. Dari kedua pengujian tersebut dapat disimpulkan bahwa sistem yang diujikan telah diujikan sesuai dengan jalur independennya diman untuk fitur login memiliki 4 jalur independen dan fitur pelaporan penjualan memiliki 4 jalur independen. Dari setiap jalur independen yang diujikan, pada pengujian ini semua skenario sesuai dengan yang diharapkan atau tidak ada kesalahan ataupun bug dalam sistem informasi monitoring dava kebab.

4. SIMPULAN

Pendekatan *white box testing* menggunakan teknik *Basis Path* digunakan dalam pengujian sebuah sistem informasi monitoring Dava Kebab, pendekatan ini digunakan untuk menilai aliran program dan menghitung jumlah skenario pengujian yang perlu dijalankan selama *Test Case*. Dalam Pengujian ini hanya beberapa fitur saja yang diujikan di dalamnya yakni fitur login dan pelaporan penjualan. Hasil pengujian melibatkan pembuatan *flowchart*, *flowgraph*, perhitungan *Cyclomatic Complexity*, dan pembuatan *Test Case* untuk setiap jalur independen. Berdasarkan hasil pengujian memakai metode White Box Testing pada teknik Basis Path, dapat disimpulkan bahwa sistem informasi Monitoring Dava Kebab telah diuji secara menyeluruh sesuai dengan jalur independen yang telah ditetapkan. Ditemukan bahwa fitur login memiliki 4 jalur independen, sedangkan fitur pelaporan penjualan memiliki 4 jalur independen. Setiap jalur independen yang diuji menghasilkan skenario yang sesuai dengan harapan, di mana tidak terdapat adanya kesalahan ataupun bug pada sistem. Oleh karena itu, dapat disimpulkan bahwa pengujian sistem informasi Monitoring Dava Kebab menggunakan White Box Testing pada teknik Basis Path telah

memberikan akhir yang memuaskan dan menunjukkan keandalan sistem dalam memenuhi kebutuhan fungsionalnya.

REFERENSI

- [1] A. Bagus Setiawan, W. Rachmawati, A. Taufiq Arrahman, N. Natasyah, and F. N. S. Fadil, "Aplikasi Monitoring Stok Barang Berbasis Web Pada PT. Intermetal Indo Mekanika," *ABDI*, vol. 2, no. 2, pp. 1–6, Sep. 2021, doi: 10.34306/abdi.v2i2.254.
- [2] A. Agustian, Sifa Fauziah, and Wahyu Hadikristanto, "Sistem informasi dan metode pengelolaan manufaktur order berbasis website dengan metode waterfall (Studi kasus PT. Aji)," *infotech*, vol. 4, no. 2, pp. 147–156, Jul. 2023, doi: 10.37373/infotech.v4i2.662.
- [3] G. I. Marthasari, A. T. Wahyuningsih, M. R. Aviansyah, M. A. Ramadhani, and Z. Rahmatullah, "Pengujian Website Infotech Menggunakan Teknik Black-Box Decision Table," vol. 7, no. 1, 2022.
- [4] R. Parlika, T. A. Nisaa', S. M. Ningrum, and B. A. Haque, "Studi Literatur Kekurangan dan Kelebihan Pengujian Black Box," 2020.
- [5] I. R. Dhaifullah, M. Muttanifudin H, A. Ananda Salsabila, and M. Ainul Yaqin, "Survei Teknik Pengujian Software," *jacis*, vol. 2, no. 1, pp. 31–38, Jun. 2022, doi: 10.47134/jacis.v2i1.42.
- [6] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING*, vol. 3, no. 2, p. 206, Dec. 2018, doi: 10.30998/string.v3i2.3048.
- [7] Anas Faridrahman and D. Gunawan, "Sistem informasi monitoring kelayakan dan perawatan kendaraan angkutan barang pada PT Mitra Gemah Sentosa Jakarta berbasis website," *infotech*, vol. 4, no. 2, pp. 179–190, Dec. 2023, doi: 10.37373/infotech.v4i2.748.
- [8] M. F. Londjo, "IMPLEMENTASI WHITE BOX TESTING DENGAN TEKNIK BASIS PATH PADA PENGUJIAN FORM LOGIN," 2021.
- [9] A. D. Herlambang, A. Rachmadi, K. Utami, R. I. Hakim, and N. Rohmah, "Pengembangan Fitur E-Matur dengan V-Model sebagai Alat Pengaduan Publik untuk Website Badan Kepegawaian Negara," *JTIK*, vol. 6, no. 5, pp. 467–474, Oct. 2019, doi: 10.25126/jtiik.2019651319.
- [10] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. Abu Bakar, N. Novianti, and R. Ramadan, "PENGUJIAN WHITE-BOX PADA APLIKASI DEBT MANAGER BERBASIS ANDROID," *Naratif*, vol. 4, no. 1, pp. 11–22, Jun. 2022, doi: 10.53580/naratif.v4i1.147.
- [11] J. B. L. Sie, Izmy Alwiah Musdar, and Syamsul Bahri, "Pengujian White Box Testing Terhadap Website Room Menggunakan Teknik Basis Path," *KHARISMA Tech*, vol. 17, no. 2, pp. 45–57, Sep. 2022, doi: 10.55645/kharismatech.v17i2.235.
- [12] R. I. Ndaumanu, "Pengujian Sistem Informasi Perpustakaan Berbasis Website dengan Basis Path Testing," *JUSTEK*, vol. 6, no. 1, p. 123, Mar. 2023, doi: 10.31764/justek.v6i1.13808.
- [13] C. T. Pratala, E. M. Asyer, I. Prayudi, and A. Saifudin, "Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path," *JIUP*, vol. 5, no. 2, p. 111, Jun. 2020, doi: 10.32493/informatika.v5i2.4713.
- [14] V. Pratap, "WHITE-BOX TESTING TECHNIQUE FOR FINDING DEFECTS," vol. 8, no. 7.
- [15] T. A. Kurniawan, "Pengujian Struktur Program Dengan Pengujian Jalur Dasar (Basis Path Testing): Teori Dan Aplikasi," *jeccis*, vol. 1, no. 1, p. 29, Jun. 2016, doi: 10.21776/jeccis.v1i1.357.
- [16] Y. J. Solissa, F. Putra, A. N. Putri, and S. R. C. Nursari, "Pengujian White Box Berbasis Path pada Form Daftar Jobstreet.co.id," *KONSTELASI*, vol. 3, no. 2, pp. 353–362, Dec. 2023, doi: 10.24002/konstelasi.v3i2.8287.
- [17] Program Studi Informatika, Universitas Nasional *et al.*, "Pengembangan Aplikasi Sistem Informasi Smart Register Online Berbasis Android Menggunakan Algoritma BruteForce," *EDUMATIC*, vol. 4, no. 1, pp. 47–56, Jun. 2020, doi: 10.29408/edumatic.v4i1.2106.
- [18] S. Pare, "DESAIN DAN IMPLEMENTASI E-COMMERCE PADA TOKO AS 88 CELLULER MERAUKE," no. 3, 2013.