

# PENGUJIAN KUALITAS WEBSITE PT MEDIA CITRA DIGITALINDO BLITAR MENGGUNAKAN WHITE BOX TESTING DENGAN TEKNIK BASIS PATH

Bagus Prasetya\*<sup>1)</sup>, Ilyas Nuryasin<sup>2)</sup>

1. Informatika, Universitas Muhammadiyah Malang, Indonesia
2. Informatika, Universitas Muhammadiyah Malang, Indonesia

## Article Info

**Kata Kunci:** White Box Testing; Basis Path Analysis; Cyclomatic Complexity; Independent Path; Test Case

**Keywords:** White Box Testing; Basis Path Analysis; Cyclomatic Complexity; Independent Path; Test Case

## Article history:

Received 21 April 2025  
Revised 30 April 2025  
Accepted 2 Mei 2025  
Available online 6 Mei 2025

## DOI :

<https://doi.org/10.29100/jupi.v10i2.7693>

\* Corresponding author.

Bagus Prasetya

E-mail address:

[bagusprasetya@webmail.umm.ac.id](mailto:bagusprasetya@webmail.umm.ac.id)

## ABSTRAK

Dalam siklus pengembangan perangkat lunak, pengujian adalah fase penting untuk memastikan kualitas dan reliabilitas sistem. Salah satu tahap pengujian awal adalah pengujian unit, yang dapat diimplementasikan menggunakan White Box Testing dengan teknik Basis Path Analysis. Metode ini menguji struktur internal suatu program melalui bagan alur sistem (system flowchart), yang kemudian dikonversi menjadi flowgraph untuk mengidentifikasi jalur eksekusi independen. Tantangan muncul ketika sistem perangkat lunak terdiri dari banyak file yang saling berhubungan, sulit untuk membuat flowgraph yang komprehensif dari kode sumber. Penelitian ini menerapkan Basis Path Analysis dalam White Box Testing pada situs web PT Media Citra Digitalindo dengan memanfaatkan system-level flowchart sebagai solusi praktis untuk menangani kompleksitas kode. Pendekatan ini terdiri dari empat tahap utama: membangun flowgraph, menghitung Cyclomatic Complexity (CC), mengidentifikasi jalur independen, dan mengeksekusi test case. Hasil pengujian menunjukkan bahwa prosedur sistem yang dianalisis menunjukkan kompleksitas sedang hingga tinggi, dengan tingkat risiko yang sesuai. Selain itu, semua jalur independen yang teridentifikasi berhasil divalidasi melalui test case. Temuan ini mendukung efektivitas teknik Basis Path dalam meningkatkan kemampuan uji (testability) sistem berbasis web yang kompleks, dan berkontribusi pada peningkatan praktik validasi praktik validasi untuk logika internal program.

## ABSTRACT

In the software development lifecycle, testing is a critical phase to ensure the quality and reliability of the system. One of the early testing stages is unit testing, which can be implemented using White Box Testing with the Basis Path Analysis technique. This method examines the internal structure of a program through system flowcharts, which are then converted into flowgraphs to identify independent execution paths. A challenge arises when the software system comprises numerous interconnected files, making it difficult to construct a comprehensive flowgraph from the source code. This study applies Basis Path Analysis in White Box Testing on the PT Media Citra Digitalindo website by utilizing system-level flowcharts as a practical solution for dealing with code complexity. The approach consists of four main stages: constructing flowgraphs, calculating Cyclomatic Complexity (CC), identifying independent paths, and executing test cases. The test results show that the analyzed system procedures exhibit moderate-to-high complexity, with a corresponding medium level of risk. Furthermore, all identified independent paths were successfully validated through test cases. The findings support the effectiveness of the Basis Path technique in enhancing the testability of complex web-based systems and contribute to improved validation practices for internal program logic.

## I. PENDAHULUAN

**P**ERKEMBANGAN teknologi informasi yang pesat telah mendorong penggunaan aplikasi berbasis web di berbagai sektor, termasuk dalam bidang perdagangan digital [1][2]. Salah satu inovasi penting dalam ekosistem digital adalah platform Social Media Marketing (SMM) panel, yaitu layanan berbasis web yang berfungsi untuk mengoptimalkan interaksi dan efektivitas promosi melalui media sosial. Menurut laporan pasar global, sektor SMM panel diproyeksikan mengalami pertumbuhan tahunan sebesar 13,1% (CAGR) selama periode 2022–2027, dengan nilai transaksi diperkirakan mencapai USD 1,2 miliar pada tahun 2027 [3]. Di Indonesia, penggunaan layanan ini juga menunjukkan tren peningkatan seiring dengan kebutuhan promosi digital yang semakin tinggi [4].

Pertumbuhan pesat industri SMM panel menuntut tersedianya sistem layanan berbasis web yang tidak hanya fungsional, tetapi juga andal, aman, dan mampu menangani volume transaksi dalam skala besar. Kualitas perangkat lunak menjadi faktor krusial dalam mempertahankan kepercayaan pengguna dan mendukung kelangsungan operasional layanan [5].

PT Media Citra Digitalindo, sebuah perusahaan berbasis di Blitar, merupakan salah satu penyedia layanan SMM panel yang menawarkan berbagai fitur untuk mendukung strategi pemasaran media sosial. Website yang dimiliki perusahaan ini berfungsi sebagai platform utama dalam mengelola transaksi layanan, integrasi API dengan berbagai platform media sosial, serta pengelolaan data transaksi secara real-time [6]. Kompleksitas arsitektur sistem yang tinggi, dikombinasikan dengan kebutuhan akan performa yang konsisten dan tingkat keamanan yang tinggi, menjadikan platform ini sebagai objek penelitian yang relevan untuk evaluasi kualitas perangkat lunak.

Dalam menghadapi tantangan tersebut, pengujian kualitas perangkat lunak menjadi aspek yang esensial untuk memastikan reliabilitas, keamanan, dan stabilitas sistem. Penelitian ini mengacu pada kerangka standar kualitas perangkat lunak ISO/IEC 25010 [7][8], dengan fokus khusus pada karakteristik reliability dan security sebagai tolok ukur evaluasi kualitas [9]. Standar ini digunakan secara luas dalam pengembangan sistem perangkat lunak kompleks, termasuk sistem berbasis web dengan layanan terintegrasi lintas platform.

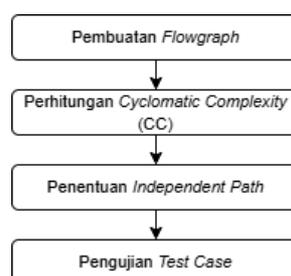
Metode pengujian white box testing dipilih dalam penelitian ini karena memungkinkan analisis mendalam terhadap struktur internal perangkat lunak. Teknik yang digunakan adalah basis path testing, yang bertujuan mengevaluasi kompleksitas jalur logika program serta memastikan seluruh jalur eksekusi yang mungkin dapat diuji secara sistematis. Beberapa studi sebelumnya [10][11] telah mengimplementasikan teknik ini, namun umumnya diterapkan pada sistem berskala kecil hingga menengah tanpa mempertimbangkan karakteristik sistem modern yang terhubung secara real-time dan terdistribusi.

Oleh karena itu, penelitian ini memperkenalkan pendekatan segmentasi kode sebagai inovasi dalam pengujian basis path pada sistem kompleks. Tidak seperti pendekatan konvensional yang menguji seluruh sistem secara monolitik, segmentasi memungkinkan pengujian terfokus pada modul-modul kritis berdasarkan fungsionalitas utama. Pendekatan ini tidak hanya meningkatkan akurasi dalam identifikasi jalur logika, tetapi juga mempercepat deteksi kesalahan dan mendukung efisiensi pengujian secara keseluruhan [12].

Dengan mengimplementasikan white box testing berbasis path dan pendekatan segmentasi kode, penelitian ini bertujuan mengevaluasi kualitas struktur internal website PT Media Citra Digitalindo serta mengidentifikasi potensi kesalahan atau ketidaksesuaian dalam sistem. Melalui upaya ini, diharapkan platform dapat meningkatkan keandalannya dalam menyediakan layanan pemasaran digital yang efektif dan terpercaya.

## II. METODE PENELITIAN

Penelitian ini menggunakan metode white box testing dengan pendekatan basis path testing untuk mengevaluasi kualitas struktur internal website PT Media Citra Digitalindo. Tujuan utama pengujian adalah untuk menilai kestabilan (stability) dan keandalan (reliability) sistem melalui analisis alur logika program. Pengujian dilakukan melalui empat tahapan utama, sebagaimana dapat dilihat pada gambar 1 [13].



Gambar. 1. Tahapan Basis Path Analysis

### A. Pembuatan Flowgraph

Definisikan Flowgraph digunakan untuk merepresentasikan alur logika program, yang disusun berdasarkan kode program atau flowchart sistem. Dalam flowgraph, elemen logika diwakili oleh node (simpul) yang merepresentasikan pernyataan prosedural, sedangkan edge (panah) menunjukkan hubungan alur eksekusi antar pernyataan.

Pada penelitian ini, flowgraph dibangun berdasarkan flowchart sistem alih-alih langsung dari kode sumber. Hal ini dikarenakan struktur program website PT Media Citra Digitalindo terdiri atas banyak berkas (file) yang saling berinteraksi, sehingga penyusunan flowgraph langsung dari kode dinilai kurang efektif. Representasi melalui flowchart memberikan gambaran yang lebih terstruktur dan memudahkan proses analisis jalur eksekusi.

### B. Perhitungan CC (Cyclomatic Complexity)

Cyclomatic Complexity (CC) adalah ukuran kuantitatif yang digunakan untuk menilai kompleksitas logika dalam program. Dalam metode basis path, nilai CC menunjukkan jumlah independent path dalam basis set program, dan menentukan jumlah minimum test case yang harus disiapkan agar setiap jalur eksekusi diuji setidaknya satu kali.

Nilai CC dihitung menggunakan persamaan (1):

$$V(G) = E - N + 2 \text{ atau } V(G) = P + 1 \quad (1)$$

dengan:

- V(G) = Cyclomatic Complexity
- E = Jumlah edge pada flowgraph
- N = Jumlah node pada flowgraph
- P = Jumlah predicate node pada flowgraph

Apabila program memiliki ukuran kecil hingga sedang (kurang dari 500 baris kode atau  $CC < 10$ ), perhitungan dapat dilakukan secara manual. Namun, untuk sistem dengan kompleksitas tinggi, penggunaan alat bantu analisis (static code analysis tools) disarankan. Kompleksitas program yang tinggi juga berpotensi menyulitkan penyusunan flowgraph secara akurat akibat keterlibatan banyak Control Flow Graph (CFG) [14].

Selain itu, penelitian ini juga melakukan analisis tingkat risiko berdasarkan nilai CC, walaupun tidak termasuk ke dalam teknik basis path murni. Analisis ini merujuk pada penelitian sebelumnya yang menunjukkan korelasi antara nilai CC dengan tingkat kerentanan sistem terhadap cacat (defect density), sebagaimana dapat dilihat pada tabel 1 [13].

TABEL I  
 KORELASI ANTARA CYCLOMATIC COMPLEXITY DENGAN TINGKAT RESIKO

CC	Procedure	Level of Complexity	Testability	Cost and Effort (Risk Level)
1 – 10	A well structured and stable procedure	Less	High	Less
11 – 20	A more complex procedure	High	Medium	Medium
21 – 40	A complex procedure, alarming	Very High	Low	High
> 40	An error-prone, extremely troublesome, un-testable procedure	-	Not at all Testable	Very High

### C. Penentuan Independent Path

Independent path adalah jalur unik dari awal hingga akhir flowgraph yang mencakup setidaknya satu edge baru yang belum dilewati jalur lainnya. Penentuan jalur ini bertujuan untuk memastikan seluruh logika internal program diuji secara komprehensif, sehingga meningkatkan kestabilan dan kualitas sistem.

### D. Pengujian Test Case

Jumlah test case minimum yang disusun setara dengan jumlah independent path yang diperoleh dari perhitungan CC. Setiap test case dirancang untuk menelusuri jalur eksekusi tertentu dengan tujuan mengidentifikasi potensi kesalahan dalam kode program.

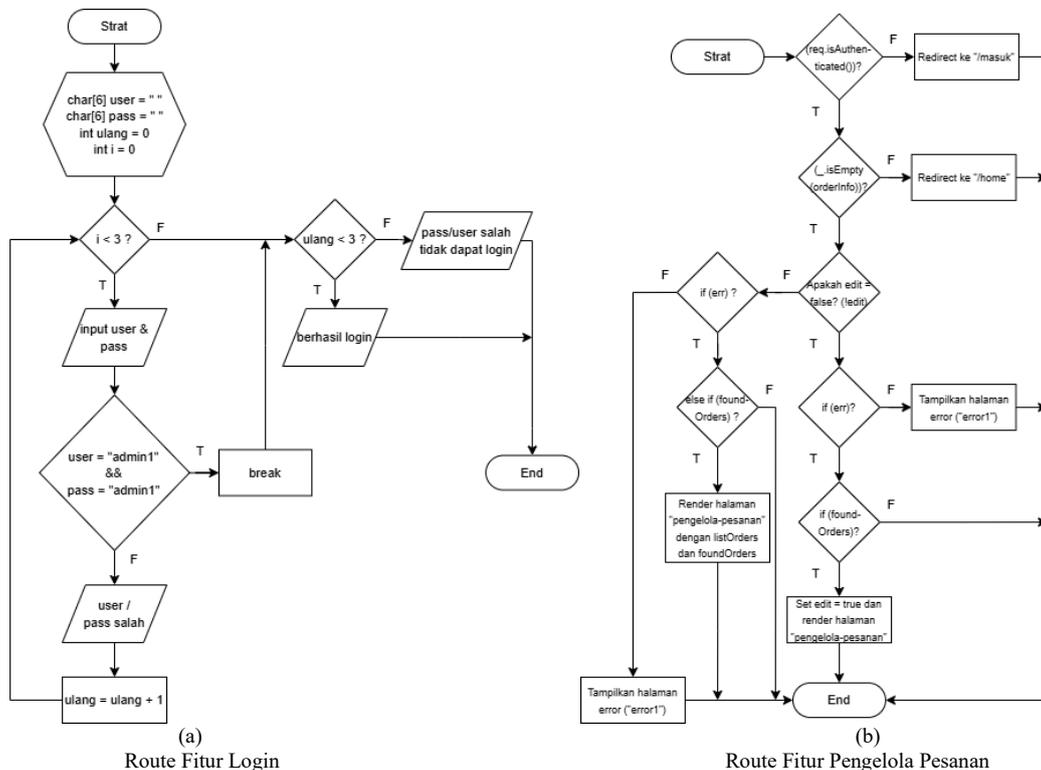
## III. HASIL DAN PEMBAHASAN

Pengujian dilakukan terhadap dua fitur utama pada website PT Media Citra Digitalindo, yaitu fitur login dan fitur pengelola pesanan, menggunakan metode white box testing dengan pendekatan basis path. Proses analisis terdiri atas empat tahapan: pembuatan flowgraph, perhitungan Cyclomatic Complexity (CC), penentuan independent path, dan penyusunan test case. Berikut ini adalah uraian hasil analisis untuk masing-masing fitur.

### A. Pembuatan Flowgraph

Pembuatan flowgraph dilakukan berdasarkan system-level flowchart dari fitur Login dan Pengelola Pesanan. Pendekatan ini dipilih karena struktur sistem menggunakan arsitektur modular yang terdiri dari banyak berkas kode terpisah, sehingga pembuatan flowgraph langsung dari kode sumber menjadi kurang efektif. Oleh karena itu, flowgraph disusun berdasarkan alur logika yang direpresentasikan dalam flowchart.

Diagram flowgraph menggunakan notasi node (lingkaran) untuk mewakili pernyataan prosedural, dan edge (panah) untuk menunjukkan aliran logika antar node. Untuk fitur Login, diperoleh 12 node dan 14 edge. Sementara itu, pada fitur Pengelola Pesanan, jumlah node mencapai 14 dan edge sebanyak 20.



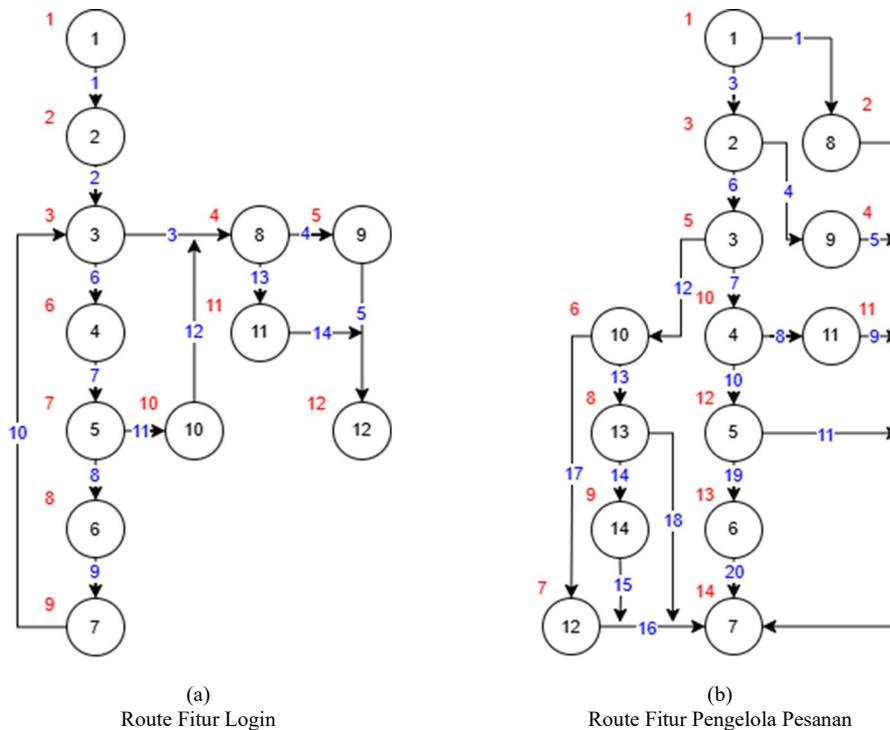
Gambar 3. Flowchart

Informasi visual serta deskripsi rinci terkait setiap node dapat dilihat pada tabel 2.

TABEL II  
 KETERANGAN UNTUK SETIAP ANGKA YANG TERDAPAT DALAM NODE FITUR PENGELOLA PESANAN

No	Keterangan
1	Start – Titik awal proses akses /form-edit-Pesanan..
2	Pengecekan autentikasi (req.isAuthenticated()) – Memeriksa apakah pengguna sudah login.
3	Pengecekan infoOrder kosong (_.isEmpty(infoOrder)) – Memeriksa apakah objek pesanan kosong.
4	Pengecekan apakah mode edit aktif (!edit) – Jika mode edit belum diaktifkan, sistem akan mengambil data pesanan dari database.
5	Pencarian pesanan berdasarkan OrderId (Order.find(...)) – Mencari pesanan di database berdasarkan infoOrder.OrderId.
6	Jika pesanan ditemukan, tambahkan item ke listItems – Jika pesanan ditemukan, sistem akan menambahkan semua item dalam pesanan ke dalam daftar listItems.
7	Set mode edit (edit = true) – Menandai bahwa mode edit telah diaktifkan.
8	Jika tidak autentikasi, redirect ke /masuk – Jika pengguna tidak terautentikasi, mereka akan diarahkan ke halaman login.
9	Jika infoOrder kosong, redirect ke /home – Jika tidak ada pesanan yang diedit, pengguna diarahkan ke halaman utama.
10	Pencarian ulang pesanan untuk edit langsung (Order.find(...)) – Jika mode edit sudah aktif, sistem langsung mencari ulang pesanan tanpa menambahkan item ke listItems.
11	Jika terjadi error dalam pencarian, render halaman error1 – Jika terjadi kesalahan dalam pengambilan data, halaman error akan ditampilkan.
12	Jika terjadi error dalam pencarian ulang, render halaman error1 – Jika ada kesalahan dalam pencarian ulang, tampilkan halaman error.
13	Render halaman edit pesanan pertama kali (res.render("form-edit-Pesanan", ...)) – Menampilkan halaman edit pesanan dengan daftar item.
14	Render halaman edit pesanan langsung (res.render("form-edit-Pesanan", ...)) – Jika tidak ada kesalahan, sistem menampilkan halaman edit pesanan dengan data yang telah ditemukan.

Diagram flowgraph masing-masing fitur dapat dilihat pada gambar 3(a) dan 3(b), yang menggambarkan fitur Login dan Pengelola Pesanan. Setiap node dalam diagram diberi nomor yang diselaraskan dengan flowchart sebelumnya (gambar 2(a) dan 2(b)) untuk menjaga konsistensi dan memudahkan penelusuran logika sistem. Untuk kejelasan visual, elemen node dan edge ditandai dengan warna berbeda dalam diagram.



Gambar. 3. Flowgraph

**B. Perhitungan Cyclomatic Complexity (CC)**

Setelah flowgraph terbentuk, dilakukan perhitungan Cyclomatic Complexity (CC) menggunakan persamaan (2):

$$V(G) = E - N + 2 \tag{2}$$

Hasil perhitungan:

Route fitur login (3):

$$V(G) = (14 - 12) + 2 = 2 + 2 = 4 \tag{3}$$

Route fitur pengelola pesanan (4):

$$V(G) = (20 - 14) + 2 = 6 + 2 = 8 \tag{4}$$

Nilai CC menunjukkan jumlah minimum independent path yang harus diuji. Nilai 4 dan 8 mengindikasikan bahwa logika program memiliki struktur bercabang yang relatif kompleks. Berdasarkan tabel klasifikasi risiko, nilai 4 termasuk dalam kompleksitas rendah hingga sedang, sedangkan nilai 8 tergolong sedang hingga tinggi namun masih dapat dikelola.

**C. Penentuan Independent Path**

Setelah nilai Cyclomatic Complexity (CC) ditentukan, langkah selanjutnya adalah mengidentifikasi independent path untuk masing-masing fitur. Setiap jalur harus mencakup edge yang unik dan menghubungkan node awal hingga node akhir dalam diagram alur kontrol program.

Pada fitur Login, diperoleh 4 jalur independen, sebagaimana dapat dilihat pada tabel 3. Seluruh jalur mencakup koneksi lengkap dari node awal hingga akhir serta mencakup seluruh edge yang tersedia, sehingga memenuhi kriteria sebagai independent path.

TABEL III  
 EMPAT (4) JALUR INDEPENDENT PATH FITUR LOGIN

No	Independent Path
1	1-2-3-8-9-12
2	1-2-3-4-5-6-7-3-8-9-12
3	1-2-3-4-5-10-8-9-12
4	1-2-3-4-5-10-8-11-12

Sementara itu, pada fitur Pengelola Pesanan, diperoleh 8 jalur independen sebagaimana dapat dilihat pada tabel 4. Jalur-jalur ini juga memenuhi syarat karena mencakup seluruh koneksi dari awal hingga akhir dan melintasi semua edge yang ada.

TABEL IV  
 DELAPAN (8) JALUR INDEPENDENT PATH FITUR PENGELOLA PEMESANAN

No	Independent Path
1	1-8-7
2	1-2-9-7
3	1-2-3-4-11-7
4	1-2-3-4-5-7
5	1-2-3-10-13-14-7
6	1-2-3-10-12-7
7	1-2-3-10-13-7
8	1-2-3-4-5-6-7

#### D. Pengujian Test Case

Setiap independent path dikonversi menjadi satu test case untuk memverifikasi jalur eksekusi dalam sistem. Konversi dilakukan dengan mengganti identitas node ke dalam bentuk aksi atau kondisi program sesuai flowchart.

Untuk fitur login, diuji 4 test case, dan fitur pengelola pesanan diuji dengan 8 test case. Hasil pengujian menunjukkan seluruh jalur berjalan sesuai ekspektasi tanpa ada jalur yang gagal dieksekusi. Ini menunjukkan stabilitas dan konsistensi logika program.

Secara keseluruhan, hasil pengujian menunjukkan bahwa prosedur sistem pada kedua fitur memiliki kompleksitas sedang hingga tinggi namun masih dalam kategori risiko yang dapat diterima. Metode Basis Path terbukti efektif dalam mengidentifikasi jalur kritis, membantu proses validasi struktural sistem, serta mendukung perancangan uji perangkat lunak berbasis logika internal yang kuat.

#### E. Diskusi Analitis dan Implikasi

Nilai Cyclomatic Complexity (CC) yang diperoleh, yaitu 4 untuk fitur login dan 8 untuk fitur pengelola pesanan, berada dalam rentang yang secara industri masih dianggap dapat dikelola. Namun, nilai  $CC = 8$  menandakan bahwa fitur terkait memiliki logika percabangan yang padat dan mungkin memerlukan perhatian ekstra dalam pemeliharaan. Ini sejalan dengan klasifikasi McCabe dan berbagai studi modern yang menyarankan pengujian lanjutan untuk modul dengan nilai  $CC > 7$  [15].

Seluruh independent path yang diuji berhasil dijalankan, namun cakupan pengujian secara struktural belum sepenuhnya menjangkau skenario kegagalan yang khas pada sistem SMM panel, seperti timeout API, keterlambatan respons dari layanan media sosial eksternal, atau kondisi akses bersamaan (race condition). Oleh karena itu, pengujian tambahan berbasis skenario dinamis tetap direkomendasikan [16].

Jalur pada fitur pengelola pesanan menunjukkan struktur yang lebih kompleks, dengan percabangan bersarang dan kondisi bergantung pada status pesanan. Jalur seperti ini memiliki risiko lebih tinggi terhadap kesalahan ketika sistem berjalan dalam kondisi beban tinggi atau dalam situasi tak terduga. Perlu dilakukan pemantauan khusus terhadap jalur ini pada tahap produksi [17].

Dari sisi pengelolaan kode, nilai CC yang tinggi berdampak pada peningkatan effort dalam pemeliharaan. Penambahan logika baru seperti integrasi platform media sosial tambahan akan memperluas jalur eksekusi, yang dapat menyebabkan ledakan kompleksitas jika tidak diantisipasi melalui modularisasi dan refactoring [18].

Dibandingkan black-box testing, pendekatan white-box memberikan keunggulan dalam pengujian logika internal yang tersembunyi, seperti penanganan error atau validasi internal. Pengujian ini sangat berguna untuk menemukan

jalur logika ekstrem yang mungkin tidak terpicu melalui pengujian berbasis input-output biasa [19].

Namun, perlu diakui bahwa pendekatan ini memiliki keterbatasan karena bersifat statis. Jalur interaktif dinamis, seperti permintaan asinkron menggunakan AJAX atau pembaruan status berbasis event real-time, tidak dapat diidentifikasi melalui flowgraph statis. Oleh karena itu, kombinasi pendekatan statis dan dinamis direkomendasikan untuk cakupan pengujian yang lebih holistic [20].

#### IV. KESIMPULAN

Berdasarkan hasil pengujian unit menggunakan metode white box testing dengan pendekatan basis path analysis pada website PT Media Citra Digitalindo, dapat disimpulkan bahwa struktur prosedural sistem, khususnya pada fitur login dan pengelola pesanan, memiliki tingkat kompleksitas logika yang bervariasi. Nilai Cyclomatic Complexity yang diperoleh, yakni 4 dan 8, menunjukkan bahwa sistem memiliki jumlah jalur eksekusi yang signifikan dan perlu mendapat perhatian dalam proses pengujian serta pemeliharaan.

Seluruh independent path yang teridentifikasi melalui pembuatan flowgraph berhasil dieksekusi dalam bentuk test case, dan hasil pengujian menunjukkan bahwa seluruh jalur berjalan sesuai dengan ekspektasi. Hal ini mengindikasikan bahwa struktur logika internal telah dirancang secara konsisten dan dapat diandalkan. Meskipun kompleksitas pada fitur tertentu tergolong tinggi, risiko pengelolaan sistem masih dalam kategori sedang dan dapat ditangani dengan pendekatan pengembangan yang sistematis.

Dengan demikian, pendekatan basis path terbukti efektif dalam mengidentifikasi kompleksitas logika program serta mendeteksi potensi kesalahan tersembunyi pada struktur internal perangkat lunak. Temuan ini memberikan dasar yang kuat untuk peningkatan kualitas sistem di masa mendatang, sekaligus menunjukkan pentingnya pengujian berbasis struktur dalam sistem digital berskala menengah hingga besar.

#### DAFTAR PUSTAKA

- [1] Z. Abidin and A. Sofyan, "Strategi Manajemen Pemasaran Dalam Era Digital Pada Masa Sekarang," *Khidmatussifa J. Islam. Stud.*, vol. 2, no. 1, pp. 11–16, 2022, doi: 10.56146/khidmatussifa.v2i1.76.
- [2] I. A. Kusuma, F. Dwi, N. Affah, M. G. Ruba, Y. Yudha Utama, and I. Kediri, "Analisis Manajemen Digital dalam Mengoptimalkan Kinerja Bisnis," *Proc. Islam. Econ. Business, Philanthr.*, vol. 2, no. 1, 2023, [Online]. Available: <https://jurnalfebi.iainkediri.ac.id/index.php/proceedings>
- [3] MarketsandMarkets. (Januari 2024). Social Media Management Market worth \$51.8 billion by 2028 – Exclusive Report by MarketsandMarkets™. [Online]. Tersedia: <https://www.marketsandmarkets.com/PressReleases/social-media-management.asp>.
- [4] E. Saliin, *Social Media Marketing*, no. June. 2023. doi: 10.1201/9781003200154-2.
- [5] M. A. Arsath, "Digitalization in Business' Social Media Marketing: Advantages and Disadvantages," *Shanlax Int. J. Manag.*, vol. 6, no. 1, pp. 152–158, 2018, [Online]. Available: <https://doi.org/10.5281/>
- [6] M. Radisi *et al.*, "Transformasi Digital UMKM Shofi Cookies : Pengembangan Dashboard Penjualan Berbasis API , Database , dan Visualisasi Interaktif menggunakan wpDataTable," vol. 04, pp. 89–97, 2024.
- [7] W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, "Proceedings of the ninth international conference on dependability and complex systems DepCoS-RELCOMEX. June 30 – July 4, 2014, Brunów, Poland," *Adv. Intell. Syst. Comput.*, vol. 286, pp. 223–224, 2014, doi: 10.1007/978-3-319-07013-1.
- [8] M. D. Mulyawan, I. N. S. Kumara, I. B. A. Swamardika, and K. O. Saputra, "Kualitas Sistem Informasi Berdasarkan ISO/IEC 25010: Literature Review," *Maj. Ilm. Teknol. Elektro*, vol. 20, no. 1, p. 15, 2021, doi: 10.24843/mite.2021.v20i01.p02.
- [9] J. P. Miguel, D. Mauricio, and G. Rodríguez, "A Review of Software Quality Models for the Evaluation of Software Products," *Int. J. Softw. Eng. Appl.*, vol. 5, no. 6, pp. 31–53, 2014, doi: 10.5121/ijsea.2014.5603.
- [10] G. W. Sasmito, "White Box Testing with Basis Path Technique in the Demography Administration Website," *ICSECC 2020 - 2nd Int. Conf. Sustain. Eng. Creat. Comput. Proc.*, no. October, pp. 86–92, 2020, doi: 10.1109/ICSECC51444.2020.9557428.
- [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, 2022, doi: 10.55645/kharismatech.v17i2.235.
- [12] S. Sharma\* and D. S. Bhatia, "An Algorithm for Finding the Optimal Path In Basis Path Testing using GABVIE Model," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 3, pp. 587–593, 2020, doi: 10.35940/ijtee.c8436.019320.
- [13] C. P. C. Munaiseche and G. C. Rorimpandey, "Penerapan Metode Basis Path Analysis dalam Pengujian White Box Sistem Pakar," *Pros. Semin. Nas. Sist. Inf. dan Teknol.*, vol. 5, no. 1, pp. 124–128, 2021.
- [14] D. Madhavi, "A White Box Testing Technique in Software Testing: Basis Path Testing," *J. Res.*, vol. 2, no. 4, pp. 12–17, 2016, [Online]. Available: [www.journalforresearch.org](http://www.journalforresearch.org)
- [15] Y. Tashtoush, M. Al-Maolegi, and B. Arkok, "The Correlation among Software Complexity Metrics with Case Study," no. 2, 2014, [Online]. Available: <http://arxiv.org/abs/1408.4523>
- [16] M. E. Khan, "Different approaches to white box testing technique for finding errors," *Int. J. Softw. Eng. its Appl.*, vol. 5, no. 3, pp. 1–14, 2011, doi: 10.5121/ijsea.2011.2404.
- [17] S. Huntsman, "Path homology as a stronger analogue of cyclomatic complexity," 2020, [Online]. Available: <http://arxiv.org/abs/2003.00944>
- [18] J. Caldeira, F. Brito e Abreu, J. Cardoso, and J. P. dos Reis, "Unveiling process insights from refactoring practices," *Comput. Stand. Interfaces*, vol. 81, 2022, doi: 10.1016/j.csi.2021.103587.
- [19] D. Wintana, D. Pribadi, and M. Y. Nurhadi, "Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing," *J. Larik Ldng. Artik. Ilmu Komput.*, vol. 2, no. 1, pp. 8–16, 2022, doi: 10.31294/larik.v2i1.1382.
- [20] E. Bozdog, A. Mesbah, and A. Van Deursen, "A comparison of push and pull techniques for AJAX," *Proc. - 9th IEEE Int. Symp. Web Site Evol. WSE 2007*, pp. 15–22, 2007, doi: 10.1109/WSE.2007.4380239.