

Pengujian *White-Box* Menggunakan Teknik *Basis Path* Pada Sistem Informasi Pengajuan dan Monitoring Kerja Praktik

Marwan Aldi Pratama¹, Siti Nur'aini²

^{1,2} Universitas Islam Negeri Walisongo Semarang

Corresponding Author: 23080960135@student.walisongo.ac.id

Abstract: *The Information System for Information Technology Internships at UIN Walisongo Semarang is being developed to support the structured and integrated process of submitting and monitoring student internship activities. Along with the system design, software testing is needed to ensure that the program's logic flow runs according to requirements and is free from errors. This research aims to test the internal structure of the program using the White-Box Testing method with the Basis Path Technique. The testing focusses on two core modules: the internship application process module and the application verification process module by the study program head. The testing stages include identifying the modules to be tested, creating flowcharts, forming flowgraphs, calculating Cyclomatic Complexity (CC), determining independent paths, and testing test cases. The test results show that the internship application process model has a Cyclomatic Complexity value of 7, resulting in seven independent paths, while the application verification process module has a Cyclomatic Complexity of 2 with two independent paths. All independent paths were successfully executed and produced the expected output. Based on these results, it can be concluded that the internal logic flow in each tested module has functioned very well and according to system requirements. Therefore, the White-Box Testing method with the Basis Path Technique is proven effective in testing the quality of the logical structure in the internship information system.*

Keyword: *Basis Path; Cyclomatic Complexity (CC); Independent Path; Software Testing; Test Case; White-Box Testing*

Copyright ©2024 Walisongo Journal of Information Technology

Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



1. Pendahuluan

Saat ini perkembangan teknologi telah memberikan dampak yang signifikan terhadap berbagai aspek kehidupan manusia (Zen & Nuryasin, 2024). Pada era internet dan kemajuan teknologi saat ini, perangkat lunak (*software*) menjadi komponen yang sangat penting dalam meningkatkan kinerja dan produktivitas kerja di berbagai sektor, termasuk pendidikan, bisnis, kesehatan, dan hiburan (Pratiwi & Widiyanti, 2025; Rahman Abdillah et al., 2024; Shabika Aqmarina et al., 2024). Salah satu bidang teknologi yang mengalami perkembangan pesat dan banyak dimanfaatkan oleh masyarakat adalah sistem informasi. Sistem informasi dipilih karena kemampuannya dalam mendukung proses pengelolaan, pengaksesan, serta pendistribusian

informasi secara efektif dan efisien (Putri et al., 2025). Dimana informasi merupakan kebutuhan mendasar yang digunakan oleh seluruh lapisan masyarakat, pemanfaatannya berkaitan erat dengan tingkat intensitas dan efektivitas dalam berbagai aktivitas (Dwi Hafizah Akbar et al., 2025). Selain itu, sistem informasi juga berperan dalam pengambilan keputusan pengendalian serta pelaksanaan kegiatan operasional dalam suatu organisasi (Putri et al., 2025).

Saat ini sistem informasi tidak hanya diterapkan pada perusahaan besar saja, namun juga diterapkan dalam bidang pendidikan. Dimana Sistem informasi ini mendorong arah perubahan dalam tata kelola kehidupan akademik di lingkungan perguruan tinggi. Salah satu contoh implementasinya dapat dilihat pada sistem informasi kerja praktik yang ada di Program Studi Teknologi Informasi UIN Walisongo Semarang.

Sistem informasi kerja praktik merupakan sebuah platform yang dirancang untuk memudahkan dosen dan mahasiswa dalam proses pengajuan serta pemantauan kegiatan kerja praktik (magang). Pengembangan sistem ini didasarkan pada kebutuhan yang ada di Program Studi Teknologi Informasi UIN Walisongo Semarang. Selama ini, proses pengajuan dan pemantauan magang masih dilakukan secara manual melalui Google Form, sehingga pengelolaannya kurang terstruktur dan tidak terdokumentasi dengan baik. Dengan hadirnya sistem informasi magang ini, diharapkan proses pengajuan dan pemantauan kegiatan magang dapat berlangsung lebih tertata, sistematis, dan efisien.

Setelah melihat seberapa pentingnya sistem informasi kerja praktik ini, maka diperlukan adanya pengujian perangkat lunak, untuk memastikan sistem ini berjalan dengan baik. Pengujian dilakukan untuk menemukan dan mengidentifikasi kegagalan yang terdapat dalam sistem sehingga dapat dilakukan perbaikan. Tahap ini merupakan bagian penting untuk memastikan bahwa sistem yang dibangun bebas dari kesalahan, [9]. Secara umum, proses pengujian perangkat lunak dilakukan melalui tiga pendekatan utama, yaitu metode *Black box*, *White box* dan *Greybox*. *Black box* testing merupakan metode pengujian yang berfokus pada pemeriksaan hasil keluaran aplikasi berdasarkan data masukan yang diberikan tanpa melihat kode sumber aplikasi. Tujuannya adalah untuk memastikan bahwa setiap fungsi aplikasi berjalan sesuai dengan persyaratan yang telah ditetapkan (Jailani & Yaqin, 2024). Pengujian *White box* merupakan metode pengujian yang dilakukan dengan cara memeriksa dan menganalisis kode program secara langsung untuk mengetahui apakah terdapat kesalahan atau tidak. Melalui pengujian ini, setiap bagian kode dapat dilihat secara detail, sehingga kesalahan atau bug yang muncul dapat diidentifikasi dengan lebih jelas (Pratala et al., 2020a). Sedangkan pengujian *Gray box* testing merupakan metode pengujian yang menggabungkan pendekatan *White box* testing dan *Black box* testing (Arifandi et al., 2022).

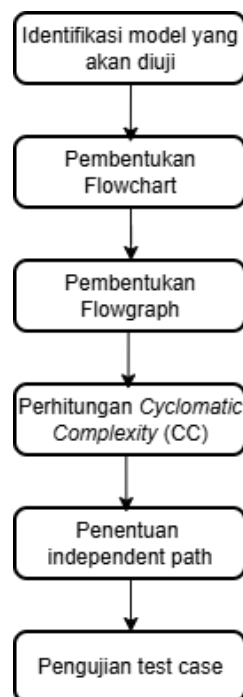
Pengujian *White box* memiliki beberapa teknik, di antaranya pengujian loop, pengujian aliran data, serta pengujian aliran kontrol. *Basis path* merupakan jalur unik yang dilewati oleh alur program, dimana setiap jalur tidak diperbolehkan mengalami pengulangan lintasan yang sama [12]. Pada proses pengujian *white box* dilakukan melalui beberapa tahapan, dimulai dari pembuatan *flowchart*, penyusunan *flowgraph*, hingga perhitungan *cyclomatic complexity* (CC) (Johar Nur Lin et al., 2024; Supriyanto et al., 2023). Pada penelitian ini, penulis akan menggunakan metode *White Box Testing* menggunakan teknik *Basis Path* untuk menguji setiap code. Metode ini dipilih karena memiliki kemampuan untuk mengidentifikasi dan menghilangkan bagian kode tersembunyi yang tidak diperlukan, serta melakukan pengujian secara mendalam dengan mengeksplorasi seluruh struktur program. Dengan demikian, kode program dapat dioptimalkan sehingga sistem mampu berjalan lebih lancar dan terstruktur.

Meskipun sejumlah studi telah banyak menerapkan teknik ini pada beberapa aplikasi, namun sebagian besar penelitian sebelumnya masih berfokus pada pengujian modul tunggal atau aplikasi dengan kompleksitas alur logika yang relatif sederhana, seperti form login atau fitur tertentu [15]. Sementara itu masih sedikit penelitian yang mengkaji penerapan teknik basis

path pada sistem informasi akademik yang memiliki alur bisnis terintegritas, melibatkan banyak pengguna serta mengandung berbagai percabangan dan validasi logika seperti pada sistem informasi kerja praktik (Ferry Kurniawan et al., 2022; Muhammad Frasha Candra Perdana et al., 2024). Selain itu, penggunaan metrik cyclomatic complexity dalam konteks sistem nyata yang memiliki tahapan berlapis seperti pengajuan dan verifikasi kerja praktik, juga masih terbatas. Oleh karena itu, penelitian ini berupaya untuk mengisi celah tersebut dengan menerapkan *Whitebox testing* menggunakan teknik *Basis path* pada modul inti dari sistem informasi kerja praktik Teknologi Informasi UIN Walisongo.

2. Metodologi penelitian

Pada penelitian ini, metode *White box testing* dengan teknik *Basis path* digunakan untuk menguji kualitas struktur kode program pada sistem informasi magang. Pengujian ini bertujuan untuk menilai tingkat kestabilan (*stability*) dan keandalan (*reliability*) sistem melalui analisis alur logika program. Proses pengujian dilakukan melalui enam tahapan utama, sebagaimana ditunjukkan pada Gambar 1. (Zen & Nuryasin, 2024).



Gambar 1. Diagram Alir Pengujian

2.1 Identifikasi Model yang Akan Diuji

Tahap awal dalam pengujian perangkat lunak pada penelitian ini adalah identifikasi modul yang akan diuji. Identifikasi modul bertujuan untuk menentukan bagian sistem yang memiliki peran penting dalam proses bisnis serta memiliki kompleksitas logika yang memerlukan pengujian secara mendalam menggunakan metode *white-box testing* (Solissa et al., 2023). Pada penelitian ini, pengujian dilakukan pada fungsi-fungsi yang berbasis ekstensi PHP.

Fungsi-fungsi yang akan di uji berupa pengajuan magang, tambah anggota, verifikasi status pengajuan, cek duplikasi pengajuan.

2.2 Pembentukan Flowchart

melakukan identifikasi model yang akan diuji, selanjutnya adalah pembentukan flowchart untuk memudahkan dalam mengeksekusi pengujian program. Secara definisi, *flowchart* merupakan representasi grafis yang menggambarkan langkah-langkah serta urutan prosedur dalam suatu aplikasi. *Flowchart* membantu analis dan programmer dalam memvisualisasikan alur kerja aplikasi yang dikembangkan, serta memecah sistem ke dalam bagian-bagian yang lebih kecil sehingga lebih mudah untuk dianalisis. Dengan demikian, penggunaan *flowchart* dapat mempermudah proses pemecahan masalah (Londjo, 2021). Selain itu, *flowchart* digunakan dalam tahap perancangan aplikasi untuk menggambarkan logika kerja sistem secara jelas dan terstruktur (Program Studi Informatika, Universitas Nasional et al., 2020).

2.3 Pembentukan Flowgraph

Flowgraph merupakan notasi sederhana yang digunakan untuk merepresentasikan control flow dalam suatu program. Pada tahap ini, kode program yang telah disusun, dikelompokkan ke dalam penomoran tertentu dan kemudian direpresentasikan dalam bentuk grafis yang saling terhubung menggunakan anak panah (Nugraha, 2022). Setiap percabangan dan perulangan dinotasikan secara visual sehingga memudahkan proses penelusuran dan analisis alur kendali (*control flow*) program (Nurwicaksono et al., 2023).

2.4 Perhitungan Cyclomatic Complexity (CC)

Cyclomatic Complexity merupakan satuan perangkat lunak yang digunakan sebagai acuan kuantitatif untuk mengukur tingkat kompleksitas logika dalam suatu program. Nilai *Cyclomatic Complexity* menggambarkan jumlah jalur independen yang terdapat pada kode program. Adapun rumus perhitungan *Cyclomatic Complexity* adalah sebagai berikut:

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

Dimana :

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

Tabel 1. Hubungan Cyclomatic Complexity dengan Resiko (Sie et al., 2022)

Nilai CC	Tipe Prosedur	Tingkat Resiko
1-4	Prosedur sederhana	Rendah
5-10	Prosedur yang terstruktur dengan baik dan stabil	Rendah
11-20	Prosedur yang lebih kompleks	Menengah
21-50	Prosedur yang kompleks dan kritis	Tinggi
>50	Rentan kesalahan, sangat mengganggu, prosedur tidak dapat diuji	Sangat Tinggi

2.5 Penentuan Independent Path

Independent path merupakan jalur unik yang dilalui dari awal hingga akhir *flowgraph* dan mencakup setidaknya satu *edge* baru yang belum di lewati oleh jaliur lainnya. Penentuan jalur independent ini bertujuan untuk memastikan seluruh logika internal program dapat diuji secara menyeluruh, sehingga mampu meningkatkan tingkat kestabilan dan kualitas sistem(Prasetya & Nuryasin, 2025).

2.6 Pengujian Test Case

Pengujian *test case* dilakukan berdasarkan independent path yang diperoleh dari perhitungan *Cyclomatic Complexity (CC)*. Setiap *test case* dieksekusi dengan memberikan data uji yang sesuai dengan jalur independen yang diuji, kemudian hasil keluaran sistem dibandingkan dengan keluaran yang diharapkan untuk menentukan keberhasilan pengujian.

3. Hasil dan pembahasan

Pengujian dilakukan pada dua fitur utama dalam perancangan sistem informasi magang, yaitu fitur pengajuan kerja praktik dan fitur verifikasi pengajuan kerja praktik oleh ketua program studi. Pengujian ini menggunakan metode white box testing dengan teknik basis path. Pengujian dilaksanakan melalui enam tahapan, yaitu : identifikasi modul yang akan diuji, pembentukan *flowchart*, pembentukan *flowgraph*, perhitungan *Cyclomatic Complexity (CC)*, penentuan *independent path*, serta pelaksanaan pengujian *test case*.

3.1 Identifikasi Model yang akan Diuji

Pada tahap ini dilakukan identifikasi terhadap modul-modul yang menjadi objek pengujian white-box pada perancangan Sistem Informasi Kerja Praktik Teknologi Informasi UIN Walisongo. Identifikasi modul bertujuan untuk menentukan bagian sistem yang memiliki peran krusial dalam alur proses magang serta memiliki struktur logika yang kompleks dan berpotensi menimbulkan kesalahan apabila tidak diuji secara menyeluruh.

Tabel 2. Code Program Proses Pengajuan

Code Program	Node
session_start();	1
include '../sim_magang/koneksi.php';	
if (!isset(\$_SESSION['login']) \$_SESSION['role'] != 'mahasiswa') {	2
header("Location: ../sim_magang/login.php");	
exit;	3
}	
\$nim = \$_SESSION['identitas'];	3
\$nama = \$_SESSION['nama_lengkap'];	
\$cek_duplikat = mysqli_query(\$conn, "	4
SELECT p.status	
FROM pengajuan p	4
LEFT JOIN anggota_kelompok a ON p.id_pengajuan = a.id_pengajuan	
WHERE p.nim_ketua = '\$nim' OR a.nim_anggota = '\$nim'	4
ORDER BY p.id_pengajuan DESC LIMIT 1	
");	4
if (\$cek_duplikat && mysqli_num_rows(\$cek_duplikat) > 0) {	
\$row = mysqli_fetch_assoc(\$cek_duplikat);	

```

$status = strtolower($row['status']);
if (in_array($status, ['menunggu','disetujui','validasi'])) {
    echo "<script>
        alert(' ⚠️ Kamu sudah memiliki pengajuan aktif atau sedang diverifikasi. Tidak
dapat mengajukan lagi. ');
        window.location.href='balasan.php';
    </script>";
    exit();
}
}
$instansi = mysqli_real_escape_string($conn, trim($_POST['instansi']));
$alamat = mysqli_real_escape_string($conn, trim($_POST['alamat']));
$mulai = $_POST['mulai'];
$selesai = $_POST['selesai'];
$no_hp = $_POST['no_hp'];
$status = 'Menunggu';
$cek_alamat = mysqli_query($conn, "
    SELECT id_pengajuan
    FROM pengajuan
    WHERE LOWER(TRIM(alamat)) = LOWER('$alamat')
    AND status IN ('Menunggu','Disetujui','Validasi')
    LIMIT 1
");
if ($cek_alamat && mysqli_num_rows($cek_alamat) > 0) {
    echo "<script>
        alert(' ⚠️ Alamat \"$alamat\" sudah digunakan oleh kelompok lain. Silakan pilih
lokasi magang berbeda. ');
        history.back();
    </script>";
    exit();
}
$q = mysqli_query($conn, "INSERT INTO pengajuan
(nama_ketua,nim_ketua,instansi,alamat,tanggal_mulai,tanggal_selesai,no_hp,status)
VALUES
('$nama','$nim','$instansi','$alamat','$mulai','$selesai','$no_hp','$status')");
if (!$q) {
    echo "<script>alert(' ❌ Gagal menyimpan pengajuan. ');history.back();</script>";
    exit();
}
if (!empty($_POST['nama_anggota']))
    foreach ($_POST['nama_anggota'] as $i => $n) {
        $nama_angg = mysqli_real_escape_string($conn, $n);
        $nim_angg = mysqli_real_escape_string($conn, $_POST['nim_anggota'][$i]);

        if (!$nama_angg || !$nim_angg) continue;
        $cek = mysqli_query($conn, "
            SELECT p.id_pengajuan FROM pengajuan p
            LEFT JOIN anggota_kelompok a ON p.id_pengajuan=a.id_pengajuan
            WHERE p.nim_ketua='$nim_angg' OR a.nim_anggota='$nim_angg' LIMIT 1
        ");

```

if (mysqli_num_rows(\$cek) > 0) { mysqli_query(\$conn, "DELETE FROM pengajuan WHERE id_pengajuan='\$id_pengajuan'"); echo "<script> alert('✗ Anggota dengan NIM \$nim_angg sudah terdaftar di pengajuan lain. Pengajuan dibatalkan.'); history.back(); </script>"; exit(); }	
mysqli_query(\$conn, "INSERT INTO anggota_kelompok (id_pengajuan,nama_anggota,nim_anggota) VALUES ('\$id_pengajuan','\$nama_angg','\$nim_angg')");	14
}	
echo "<script> alert('✅ Pengajuan berhasil dikirim dan menunggu verifikasi Kaprodi.');	15
window.location.href='balasan.php'; </script>"; exit();	16
?>	

Tabel 3. Code Program Proses Pengajuan

Code Program	Node
<?php session_start();	1
include '../sim_magang/koneksi.php'; if (!isset(\$_SESSION['login']) \$_SESSION['role'] != 'kaprodi') { header("Location: ../sim_magang/login.php"); exit(); }	2
\$id_pengajuan = \$_POST['id_pengajuan']; \$aksi = \$_POST['aksi'];	3
mysqli_query(\$conn, "UPDATE pengajuan SET status='\$aksi' WHERE id_pengajuan='\$id_pengajuan'");	4
echo "<script>alert('Status pengajuan berhasil diperbarui menjadi: \$aksi'); window.location.href='verifikasi_pengajuan.php';</script>";	5
exit(); }	6

Berdasarkan hasil analisis terhadap arsitektur sistem dan alur bisnis kegiatan magang, diketahui bahwa tidak seluruh modul memerlukan pengujian menggunakan metode *white box*. Oleh karena itu, pengujian difokuskan pada modul-modul inti yang secara langsung berpengaruh terhadap keberhasilan proses magang, mulai dari pengajuan magang oleh mahasiswa, hingga proses verifikasi pengajuan oleh ketua program studi. Hasil identifikasi menunjukkan bahwa terdapat dua modul utama yang memenuhi kriteria untuk dilakukan pengujian tersebut.

Modul pertama adalah *proses_pengajuan.php*, yang berfungsi sebagai pengajuan lokasi magang oleh mahasiswa. Modul ini dipilih karena memiliki kompleksitas yang cukup tinggi, dengan melibatkan berbagai validasi, seperti pengecekan status pengajuan sebelumnya, pemeriksaan duplikasi alamat magang, penyimpanan data pengajuan, serta pengelolaan data anggota kelompok melalui proses perulangan. Kompleksitas alur kontrol dan percabangan menjadikannya sebagai modul dengan risiko kesalahan login yang cukup tinggi apabila tidak diuji secara sistematis.

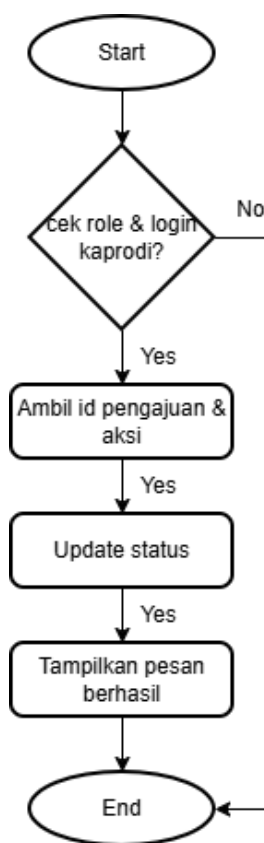
Selanjutnya, modul kedua yang akan diuji adalah *proses_verifikasi_pengajuan.php*, moduli ini digunakan oleh kaprodi untuk melakukan verifikasi pengajuan magang oleh mahasiswa. Modul ini dipilih karena merepresentasikan proses pengambilan keputusan penting dalam sistem, yaitu penentuan status pengajuan apakah disetujui atau ditolak.

3.2 Pembentukan Flowchart

Setelah menentukan modul yang akan diuji, langkah selanjutnya adalah menyusun *flowchart* untuk memudahkan dalam pengujian sistem dengan mengetahui urutan dan langkah program. Berikut adalah *flowchart* fitur pengajuan dan verifikasi pengajuan kerja praktik.



Gambar 2. Flowchart Proses Pengajuan Kerja Praktik

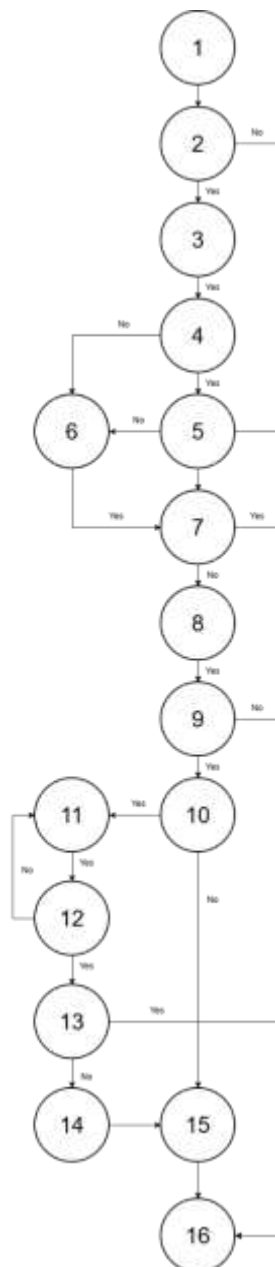


Gambar 3. Flowchart Proses Verifikasi Pengajuan

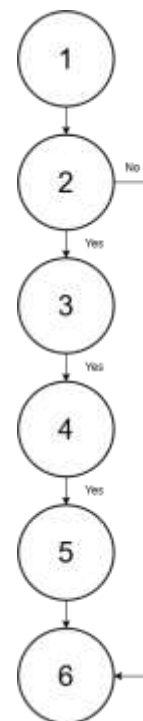
Berdasarkan *flowchart* program yang telah disusun, tahap selanjutnya adalah mengonversi alur *flowchart* tersebut ke dalam notasi program yang direpresentasikan melalui diagram *flowgraph*.

3.3 Pembentukan Flowgraph

Setelah tahap pembuatan *flowchart*, langkah selanjutnya adalah menyusun *flowgraph*. Pembuatan *flowgraph* ini bertujuan untuk memperoleh jumlah *node* dan *edge* yang diperlukan dalam perhitungan *Cyclomatic Complexity (CC)*.



Gambar 4. Flowgraph Proses Pengajuan Kerja Praktik



Gambar 2. Flowgraph Proses Verifikasi Pengajuan

Berdasarkan *Flowgraph* yang telah disusun, dapat diketahui bahwa pada modul pengajuan magang, ada 16 *node* dan 21 *edge*. Sedangkan pada modul proses verifikasi terdapat 6 *node* dan 6 *edge*. *Node* merepresentasikan urutan simbol proses dan simbol keputusan, sedangkan *edge* merupakan anak panah yang menggambarkan aliran kendali (*control flow*) sesuai dengan diagram alir (Andriyadi, 2022).

3.4 Perhitungan Cyclomatic Complexity

Setelah flowgraph terbentuk, akan dilakukan perhitungan *Cyclomatic Complexity (CC)* dengan menggunakan persamaan berikut:

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

Hasil perhitungan :

- a.) Fitur pengajuan magang

$$E = 21$$

$$N = 16$$

$$V(G) = (21 - 16) + 2 = 7$$

- b.) Fitur proses verifikasi

$$E = 6$$

$$N = 6$$

$$V(G) = (6 - 6) + 2 = 2$$

Nilai *Cyclomatic Complexity (CC)* yang telah di hitung, menunjukkan jumlah *independent path* yang perlu diuji. Nilai CC sebesar 7 dan 2 mengindikasikan bahwa logika program memiliki struktur percabangan yang cukup kompleks. Berdasarkan tabel klasifikasi risiko, nilai 2 termasuk dalam kategori rendah hingga sedang, sedangkan nilai 7 berada pada kategori sedang hingga tinggi, namun masih dalam batas yang dapat dikelola.

3.5 Penentuan Independent Path

Setelah dilakukan perhitungan *Cyclomatic Complexity (CC)* maka diperoleh nilai CC yang dapat dijadikan acuan untuk mengidentifikasi *independent path* untuk masing-masing fitur. Dimana setiap jalur harus memuat *edge* yang bersifat unik serta menghubungkan node awal hingga node akhir dalam alur kendali program (*flowgraph*). Berikut adalah tabel pemetaan *independent path*:

Tabel 4. Independent Path Proses Pengajuan Kerja Praktik

No	Independent path
IP 1	1-2-3-4-6-7-8-9-10-11-12-13-14-15-16
IP 2	1-2-16
IP 3	1-2-3-4-5-16
IP 5	1-2-3-4-6-7-16
IP 6	1-2-3-4-6-7-8-9-16
IP 7	1-2-3-4-6-7-8-9-10-11-12-13-16
	1-2-3-4-6-7-8-9-10-15-16

Dari flowgraph dan perhitungan yang telah dilakukan, maka diperoleh 7 route yang menangani node dari awal hingga akhir program. Sedangkan untuk *independent path* pada fitur verifikasi, dapat dilihat pada tabel berikut:

Tabel 5. Independent Path Proses Verifikasi Pengajuan

No	Independent path
IP 1	1-2-6
IP 2	1-2-3-4-5-6

Pada fitur verifikasi diperoleh 2 independent path. Dimana hasil ini juga memenuhi kriteria independent path, karena seluruh edge dapat menangani konektivitas dari awal hingga akhir program dan melalui semua edge yang ada.

3.6 Pengujian Test Case

Setelah penentuan jalur independent path selesai, tahapan selanjutnya adalah menyusun table test case dan membandingkan hasil keluaran actual dengan hasil yang diharapkan. Tabel test case yang telah dibuat disajikan dalam table 6 berikut :

Tabel 6. Pengujian Test Case Proses Pengajuan Kerja Praktik

No	Jalur IP	Kondisi Yang Diuji	Hasil Yang Di Harapkan	Hasil Pengujian	Keterangan
IP 1	1-2-3-4-6-7-8-9-10-11-12-13-14-15-16	Ketua Dan Anggota Valid	Pengajuan Berhasil Di Kirim	Pengajuan Berhasil	Sesuai/Valid
IP 2	1-2-16	Login Bukan Mahasiswa	Pengajuan Ditolak	Pengajuan Gagal	Sesuai/Valid
IP 3	1-2-3-4-5-16	Pengajuan Sebelumnya Masih Aktif	Tidak Dapat Membuka/Mengakses Form Pengajuan Magang	Pengajuan Gagal	Sesuai/Valid
IP 4	1-2-3-4-6-7-16	Alamat Magang Sudah Digunakan	Pengajuan Tidak Dapat Dilakukan, Karena Alamat Sama Atau Sudah Digunakan Oleh Mahasiswa Lain	Pengajuan Gagal	Sesuai/Valid
IP 5	1-2-3-4-6-7-8-9-16	Gagal Simpan Data Pengajuan	Pengajuan Tidak Berhasil	Pengajuan Gagal	Sesuai/Valid
IP 6	1-2-3-4-6-7-8-9-10-11-12-13-16	Anggota Kelompok Sudah Terdaftar	Anggota Tidak Dapat Ditambahkan Dan Pengajuan Tidak Berhasil	Pengajuan Gagal	Sesuai/Valid
IP 7	1-2-3-4-6-7-8-9-10-15-16	Pengajuan Magang Mandiri (Bukan Kelompok)	Pengajuan Berhasil	Pengajuan Berhasil	Sesuai/Valid

Berdasarkan hasil pengujian pada fitur pengajuan magang, dari total tujuh jalur yang diuji, seluruh jalur dinyatakan valid tanpa ditemukan jalur yang gagal. Hasil ini menunjukkan bahwa tidak terdapat kesalahan pada logika program, sehingga fitur pengajuan magang dapat berfungsi dengan baik dan sebagaimana yang diharapkan.

Tabel 7. Pengujian Test Case Proses Verifikasi Pengajuan

No	Jalur IP	Kondisi yang diuji	Hasil yang diharapkan	Hasil Pengujian	Keterangan
IP 1	1-2-6	User Tidak Login/ Bukan Kaprodi	Akses Ditolak	Akses Ditolak	Sesuai/Valid
IP 2	1-2-3-4-5-6	Kaprodi Memverifikasi Pengajuan Mahasiswa	Status diperbarui (Diterima/ Ditolak)	Status Berubah	Sesuai/Valid

Pada pengujian fitur verifikasi kerja praktik, diperoleh hasil yang serupa, yaitu tidak ditemukannya error ataupun bug yang mengganggu jalannya program. Hasil ini menunjukkan bahwa fungsi tersebut telah berjalan sesuai dengan yang diharapkan oleh pengembang sistem.

Secara keseluruhan, hasil pengujian menunjukkan bahwa prosedur sistem pada kedua fitur memiliki kompleksitas sedang hingga tinggi, namun masih berada dalam basis risiko yang dapat diterima. Penerapan metode *basis path* terbukti efektif dalam mengidentifikasi jalur-jalur kritis, mendukung proses validasi struktur sistem, serta membantu perancangan pengujian perangkat lunak yang berfokus pada kekuatan logika internal program.

4. Kesimpulan dan Saran

Penelitian ini menunjukkan bahwa penggunaan metode *White-Box Testing* dengan *Teknik Basis Path* berhasil diterapkan pada perancangan Sistem Informasi Kerja Praktik pada Program Studi Teknologi Informasi UIN Walisongo Semarang. Dimana pengujian difokuskan pada dua modul inti dari sistem, yaitu modul proses pengajuan magang, modul proses verifikasi pengajuan. Hasil pengujian menunjukkan bahwa modul proses pengajuan magang memiliki tujuh jalur independent yang memiliki kompleksitas logika yang cukup tinggi, karena melibatkan validasi data, percabangan kondisi, serta perulangan dalam pengelolaan anggota kelompok.

Berdasarkan perhitungan *Cyclomatic Complexity (CC)* dan penentuan path pada masing-masing modul, seluruh jalur independent yang dihasilkan diuji melalui penyusunan dan eksekusi test case. Hasil eksekusi menunjukkan bahwa keluaran sistem pada setiap jalur uji sesuai dengan hasil yang diharapkan. Dengan demikian, dapat disimpulkan bahwa alur logika internal pada kedua modul yang diuji telah berjalan dengan baik dan sesuai dengan kebutuhan sistem.

Selain itu, terdapat beberapa saran yang dapat dijadikan sebagai bahan pertimbangan untuk pengembangan sistem maupun penelitian selanjutnya. Pengujian sistem dapat dikembangkan dengan mengombinasikan dengan metode white box testing dan white box testing. Pendekatan ini diharapkan mampu menghasilkan evaluasi yang lebih komprehensif, baik dari aspek struktur internal program maupun sisi fungsionalitas sistem dan pengalaman pengguna.

References

- Andriyadi, A. (2022). Evaluasi Sistem Informasi Perpustakaan Institut Informatika Darmajaya Dengan Whitebox Testing. *Journal Of Innovation Research And Knowledge*, 1(2). <https://doi.org/10.53625/jirk.V1i8.1132>
- Arifandi, A., Simamora, R. N. Z., Janitra, G. A., Yaqin, M. A., & Huda, M. M. (2022). *Survei Teknik-Teknik Pengujian Software Menggunakan Metode Systematic Literature Review*. 4(3). <https://doi.org/10.28926/Ilkomnika.V4i3.436>
- Dwi Hafizah Akbar, Abdullah Umar Muzammil, & Tudi Firmanto. (2025). Perancangan Sistem Informasi Pendaftaran Magang Berbasis Web Pada Pt Perkebunan Nusantara Iv Regional I. *Satesi: Jurnal Sains Teknologi Dan Sistem Informasi*, 5(1), 23–30. <https://doi.org/10.54259/Satesi.V5i1.4031>
- Ferry Kurniawan, Erlin Ayu Khrisnawati, Rizka Hadiwiyaniti, & Anindo Saka Fitri. (2022). Pengujian Sistem Informasi Manajemen Siswa Berbasis Website Menggunakan Metode Black Box Dan White Box. *Prosiding Seminar Nasional Teknologi Dan Sistem Informasi*, 2(1), 249–261. <https://doi.org/10.33005/Sitasi.V2i1.306>
- Jailani, A., & Yaqin, M. A. (2024). Pengujian Aplikasi Sistem Informasi Akademik Menggunakan Metode Blackbox Dengan Teknik Boundary Value Analysis. *Journal Automation Computer Information System*, 4(2), 60–66. <https://doi.org/10.47134/jacis.V4i2.78>
- Johar Nur Iin, Sutoyo, Muh. N., & Astuti, U. L. (2024). Sistem Informasi Supervisi Akademik Untuk Sekolah Dasar Di Kabupaten Kolaka. *Bit-Tech*, 7(1), 153–161. <https://doi.org/10.32877/Bt.V7i1.1595>
- Londjo, M. F. (2021). *Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login*. <https://doi.org/10.37058/Jssainstek.V7i2.4086>
- Muhammad Frasha Candra Perdana, Vihi Atina, & Faulinda Ely Nastiti. (2024). Sistem Informasi Pengelolaan Magang Berbasis Web. *Informasi (Jurnal Informatika Dan Sistem Informasi)*, 16(2), 231–243. <https://doi.org/10.37424/Informasi.V16i2.314>
- Nugraha, W. A. (2022). Pengujian White Box Berbasis Path Pada Form Autentikasi Berbasis Mobile. *Jurnal Siliwangi Seri Sains Dan Teknologi*, 8(2). <https://doi.org/10.37058/Jssainstek.V8i2.4098>
- Nurwicaksono, M. A., Lisa, I. N., Tiara, A. R., & Sidik, R. (2023). Optimasi Sistem Informasi Konsultasi Hukum Melalui Pendekatan Pengujian Kombinasi White-Box Dan Black-Box. *Jurnal Manajemen Informatika (Jamika)*, 14(1), 1–15. <https://doi.org/10.34010/Jamika.V14i1.10110>
- Praniffa, A. C., Syahri, A., Sandes, F., Fariha, U., & Giansyah, Q. A. (2023). *Pengujian Black Box Dan White Box Sistem Informasi Parkir Berbasis Web*. 1.
- Prasetya, B., & Nuryasin, I. (2025). Pengujian Kualitas Website Pt Media Citra Digitalindo Blitar Menggunakan White Box Testing Dengan Teknik Basis Path. *Jipi (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 10(2), 1595–1601. <https://doi.org/10.29100/jipi.V10i2.7693>

- Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. (2020a). Pengujian White Box Pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path. *Jurnal Informatika Universitas Pamulang*, 5(2), 111. <https://doi.org/10.32493/informatika.v5i2.4713>
- Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. (2020b). Pengujian White Box Pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path. *Jurnal Informatika Universitas Pamulang*, 5(2), 111. <https://doi.org/10.32493/informatika.v5i2.4713>
- Pratiwi, Y., & Widiyanti, L. W. (2025). Implementasi Whitebox Testing Dengan Teknik Basis Path Pada Pengujian Halaman Pencarian Program Promo. *Jurnal Kecerdasan Buatan Dan Teknologi Informasi*, 4(2), 173–180. <https://doi.org/10.69916/jkbti.v4i2.280>
- Program Studi Informatika, Universitas Nasional, Suprpto, D. D. A., Fauziah, F., Program Studi Informatika, Universitas Nasional, Fitri, I., Program Studi Informatika, Universitas Nasional, Hayati, N., & Program Studi Informatika, Universitas Nasional. (2020). Pengembangan Aplikasi Sistem Informasi Smart Register Online Berbasis Android Menggunakan Algoritma Bruteforce. *Edumatic : Jurnal Pendidikan Informatika*, 4(1), 47–56. <https://doi.org/10.29408/Edumatic.v4i1.2106>
- Putri, D. A., Wahanani, H. E., & Nurlaili, A. L. (2025). Penerapan Teknik Basis Path Pada Pengujian White Box Sistem Informasi Perencanaan Dan Penganggaran Responsive Gender Di Diskominfo Kabupaten Jombang. *Jipi (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 10(3), 2366–2377. <https://doi.org/10.29100/jipi.v10i3.6338>
- Rahman Abdillah, Rudi Hermawan, Wawan Hermawansyah, Ibnu Adkha, & Heri Arifin. (2024). Pengujian Perangkat Lunak Sistem Informasi Inventori Pada Usaha Jasa Pengiriman Paket. *Polygon : Jurnal Ilmu Komputer Dan Ilmu Pengetahuan Alam*, 2(4), 166–175. <https://doi.org/10.62383/Polygon.v2i4.199>
- Shabika Aqmarina, A., Prima Aditiawan, F., & Endah Wahanani, H. (2024). Pengujian Sistem Informasi Perpustakaan Sma Wijaya Putra Surabaya Menggunakan Metode Black Box Testing Dengan Teknik Equivalence Partitioning Dan Boundary Value Analysis. *Jati (Jurnal Mahasiswa Teknik Informatika)*, 8(1), 855–860. <https://doi.org/10.36040/jati.v8i1.8898>
- Sie, J. B. L., Izmy Alwiah Musdar, & Syamsul Bahri. (2022). Pengujian White Box Testing Terhadap Website Room Menggunakan Teknik Basis Path. *Kharisma Tech*, 17(2), 45–57. <https://doi.org/10.55645/kharismatech.v17i2.235>
- Solissa, Y. J., Putra, F., Putri, A. N., & Nursari, S. R. C. (2023). Pengujian White Box Berbasis Path Pada Form Daftar Jobstreet.Co.Id. *Konstelasi: Konvergensi Teknologi Dan Sistem Informasi*, 3(2), 353–362. <https://doi.org/10.24002/konstelasi.v3i2.8287>
- Supriyanto, K., Aziiz, D. W., & Permatasari, H. (2023). *Pengujian White Box Sistem Informasi Kasir Toko A Berbasis Web*. <https://ojs.ujb.ac.id/senatib/article/view/3234>
- Zen, H. R. R., & Nuryasin, I. (2024). *Penerapan Whitebox Testing Pada Pengujian Sistem Menggunakan Teknik Basis Path*. <https://doi.org/10.35145/joisie.v8i1.4229>
- Zikra, A. A., Kuntara, M. R., & Hutabarat, S. H. (2025). *Perancangan Sistem Informasi Berbasis Web Untuk Pengelolaan Permohonan Data Pada Instansi Pemerintah*. <https://doi.org/10.56338/jks.v8i1.6963>