

White-Box Testing Using the Path Basis Technique on Digital Learning System in SKB Subang Regency.

Pengujian White-Box Menggunakan Teknik Basis Path pada Sistem Pembelajaran Digital di SKB Kabupaten Subang.

Syifa Nazwa Aulia^{1a,*}

¹Politeknik Negeri Subang

syifanazwaa470@gmail.com

*Correspondence: syifanazwaa470@gmail.com

Abstract

The development of information technology has driven transformation in the education sector, including in non-formal institutions such as the Subang Regency SKB which developed a digital learning system. However, in its implementation, this system has not been fully tested in terms of the application's internal logic. This study aims to test the reliability of the program logic in three main features: editing student data, submitting assignments, and saving grades using a path-based white-box testing technique. The method uses a quantitative-descriptive approach. Data were collected through direct observation, source code analysis, and interviews with system administrators. Testing was carried out by mapping flowgraphs, calculating Cyclomatic complexity (CC) values, and compiling and executing test cases for each logic path. The test results show that the student data editing feature has three independent paths with CC = 3, the submitting assignment feature has CC = 3, and saving grades has two independent paths with CC = 2. All paths have been tested and the results are in accordance with the expected output. In conclusion, the path-based white-box testing technique has proven effective in verifying the reliability of the digital learning system logic, and contributes to improving the quality of technology-based non-formal education services.

Keywords: White box Testing 1, Basis Path 2, Digital Learning System 3.

Abstrak

Perkembangan teknologi informasi telah mendorong terjadinya transformasi dalam sektor pendidikan, termasuk di lembaga nonformal seperti SKB Kabupaten Subang yang mengembangkan sistem pembelajaran digital. Namun, dalam pelaksanaannya, sistem ini belum sepenuhnya diuji dari segi logika internal aplikasi. Penelitian ini bertujuan untuk menguji keandalan logika program pada tiga fitur utama edit data peserta didik, kirim tugas, dan simpan nilai dengan menggunakan teknik white-box testing yang berbasis pada basis path. Metodenya menggunakan pendekatan kuantitatif-deskriptif. Data dikumpulkan melalui observasi langsung, analisis kode sumber, dan wawancara dengan pengelola sistem. Pengujian dilakukan dengan memetakan flowgraph, menghitung nilai *Cyclomatic complexity* (CC), serta menyusun dan mengeksekusi test case untuk setiap jalur logika. Hasil pengujian menunjukkan bahwa fitur edit data peserta didik memiliki tiga jalur independen dengan CC = 3, fitur kirim tugas memiliki CC = 3, dan simpan nilai memiliki dua jalur independen dengan CC = 2. Semua jalur telah diuji dan hasilnya sesuai dengan output yang diharapkan. Kesimpulannya, teknik white-box testing berbasis basis path terbukti efektif dalam memverifikasi keandalan logika sistem pembelajaran digital, serta berkontribusi pada peningkatan mutu layanan pendidikan nonformal yang berbasis teknologi.

Kata Kunci: White box Testing 1, Basis Path 2, Sistem Pembelajaran Digital 3

This is an open access article under the [CC - BY](#) license.



PENDAHULUAN

Pendidikan merupakan pilar utama dalam menciptakan sumber daya manusia yang berkualitas dan kompetitif (Irfan, 2019). Di era transformasi digital saat ini, penguatan sistem pendidikan tidak hanya memerlukan peningkatan kualitas pembelajaran, tetapi juga pemanfaatan teknologi sebagai alat untuk memperluas akses dan efektivitas layanan pendidikan. Teknologi informasi dan komunikasi telah memberikan peluang besar dalam pengembangan sistem pembelajaran digital yang lebih fleksibel, adaptif, dan interaktif (Erwin et al., 2023). Sistem pembelajaran digital memungkinkan peserta didik untuk belajar

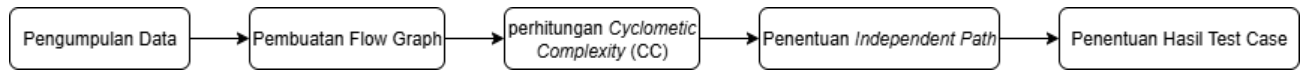
tanpa batasan ruang dan waktu serta meningkatkan keterlibatan aktif mereka dalam proses pembelajaran (Miftah, 2013). Sanggar Kegiatan Belajar (SKB) Kabupaten Subang sebagai lembaga pendidikan nonformal memiliki peran strategis dalam menyediakan layanan pendidikan kesetaraan bagi masyarakat yang tidak mengakses jalur pendidikan formal. Program-program seperti Kejar Paket A, B, dan C menjadi alternatif penting untuk mendukung pemerataan Pendidikan (Irfan, 2019). Namun, dalam praktiknya, SKB Kabupaten Subang masih menghadapi tantangan dalam pengelolaan sistem akademik. Proses pendaftaran masih dilakukan secara manual melalui formulir fisik, penyampaian informasi KBM hanya mengandalkan grup WhatsApp, dan publikasi program kurang maksimal, sehingga menyebabkan rendahnya partisipasi masyarakat serta efektivitas pengelolaan pembelajaran. Untuk permasalahan tersebut, maka dikembangkan sistem pembelajaran digital terintegrasi berbasis web dengan tujuan utama untuk meningkatkan akses, efektivitas, dan kualitas layanan pendidikan nonformal di SKB Kabupaten Subang (Miftah, 2013). Sistem ini mencakup berbagai fitur penting seperti pengelolaan akun peserta didik, pengumpulan tugas, dan penilaian hasil belajar. Namun, meskipun sistem telah tersedia, pengujian terhadap reliabilitas logika program secara menyeluruh belum dilaksanakan. Proses validasi selama ini lebih banyak berfokus pada tampilan antarmuka dan fungsi dasar, tanpa menguji jalur-jalur logika internal aplikasi yang menentukan kebenaran dan kelancaran alur eksekusi sistem (Nurwicaksono et al., 2024). Oleh karena itu, diperlukan pengujian perangkat lunak secara menyeluruh untuk memastikan bahwa seluruh alur logika program berjalan sesuai rancangan (Kurniawan et al., 2022). Salah satu pendekatan yang digunakan dalam pengujian perangkat lunak adalah *white-box testing*, yang menilai struktur internal program, termasuk kondisi logika dan alur eksekusi perintah. Dalam konteks pengujian perangkat lunak, pendekatan *white-box testing* menjadi metode penting untuk mengevaluasi struktur logika program dari sisi internal (Nugraha, 2022). Salah satu teknik utama dalam pendekatan ini adalah *basis path testing*, yang dikembangkan oleh Thomas McCabe melalui konsep *Cyclomatic complexity* untuk menghitung jalur-jalur independen dalam struktur kontrol program (Pranata et al., 2016). Teknik ini tidak hanya memungkinkan cakupan pengujian yang lebih luas, tetapi juga mampu mendeteksi kesalahan pada pengambilan keputusan dan logika percabangan yang kompleks. Oleh karena itu, metode ini sangat tepat diterapkan pada sistem pembelajaran digital di SKB yang memiliki banyak alur pengolahan data dan interaksi antarmuka (Wahyuni et al., n.d.).

Berdasarkan penelitian ini, terdapat perbedaan antara *das sollen*, yaitu kondisi ideal di mana seluruh alur logika program diuji secara struktural dengan pendekatan *white-box*, dan *das sein*, yaitu kondisi nyata yang hanya menguji fungsi permukaan tanpa analisis jalur logika internal. Kesenjangan ini berisiko menyebabkan kesalahan logika sistem yang tidak terdeteksi pada tahap awal. Beberapa penelitian sebelumnya telah menunjukkan efektivitas teknik *basis path* dalam mengidentifikasi kelemahan logika program pada sistem informasi akademik berbasis web (F. A. Putri et al., 2023). Oleh karena itu, penelitian ini bertujuan untuk menerapkan dan menganalisis metode *white-box testing* dengan menggunakan teknik *basis path* pada beberapa modul utama dalam sistem pembelajaran digital di SKB Kabupaten Subang (D. A. Putri et al., 2025). Penelitian ini diharapkan dapat memberikan gambaran menyeluruh mengenai reliabilitas logika program sistem serta berkontribusi dalam meningkatkan kualitas layanan pendidikan berbasis teknologi di lingkungan pendidikan nonformal.

METODE

Penelitian ini merupakan jenis penelitian terapan yang menggunakan pendekatan kuantitatif-deskriptif, bertujuan untuk menguji keandalan logika program dalam sistem pembelajaran digital dengan menerapkan teknik *white-box testing* berbasis *basis path* (Pratala et al., 2020). Jenis penelitian yang diterapkan adalah studi kasus, dengan objek sistem pembelajaran digital yang dikembangkan untuk SKB Kabupaten Subang. Fokus pengujian ditujukan pada tiga fitur utama, yaitu fitur pengeditan data peserta didik, pengiriman tugas, dan penyimpanan nilai (Londjo, 2021). Teknik pengumpulan data dilakukan melalui observasi langsung terhadap implementasi sistem, analisis dokumentasi kode sumber program, serta wawancara terstruktur dengan pengelola dan admin sistem (Solissa et al., 2023). Analisis data dilakukan dengan pendekatan struktural, dimulai dengan pembuatan *flowgraph* untuk memetakan alur logika dari masing-masing fitur yang diuji. Selanjutnya, perhitungan *Cyclomatic complexity* (CC) dilakukan untuk menentukan jumlah jalur independen yang perlu diuji, kemudian disusun *test case* berdasarkan jalur tersebut.

Setiap test case dijalankan dan hasil aktual dibandingkan dengan output yang diharapkan untuk menilai kesesuaian dan keandalan fungsional sistem (Putra & Nuryasin, 2024). Pendekatan ini mengacu pada praktik pengujian perangkat lunak secara menyeluruh untuk memastikan bahwa seluruh struktur kontrol program telah diuji secara komprehensif (Shiddiq, 2022). Melalui metode ini, penelitian diharapkan dapat memberikan gambaran menyeluruh mengenai keakuratan dan keandalan sistem pembelajaran digital dalam mendukung peningkatan mutu layanan pendidikan nonformal.



Gambar 2. Tahap Penelitian

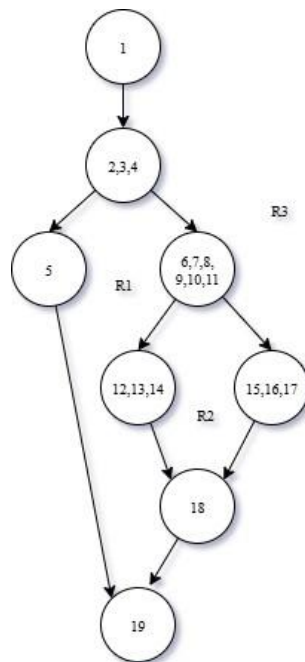
HASIL DAN PEMBAHASAN

Pengujian Logika Edit Data Peserta Didik

Tabel 1. Alur Logika Edit Data Peserta Didik

Alur Logika	Code
1: Mulai (user pilih menu Edit Pesertadidik)	public function edit(\$nisd)
2: Sistem terima request edit dengan \$nisd	{if (!\$this->M_Pesertadidik->detailData(\$nisd)) {
3: Cek data peserta didik (detailData(\$nisd))	abort(404);}
4: if (data tidak ditemukan)	\$data=['pesertadidik'=>\$this->M_Pesertadidik->detailData(\$nisd)];
5: Aksi → abort(404) (tampilkan error) → Selesai	return view('v_editpesertadidik', \$data);}
6: else (data ada)	public function update(\$nisd){
7: Tampilkan form v_editpesertadidik dengan data peserta didik	Request()->validate([
8: User isi form lalu submit (Update)	'nisd' => 'required',
9: Sistem terima request update data \$nisd	'nama_lengkap' => 'required',
10: Validasi input	'tempat_lahir' => 'required',
11: if (ada file fotopesertadidik yang diupload?)	'tanggal_lahir' => 'required',
12: Upload file → simpan ke folder foto_pesertadidik	'jenis_kelamin' => 'required',
13: Buat array data dengan fotopesertadidik	'alamat' => 'required',
14: Update tabel peserta_didik dengan data baru (pakai foto baru)	'agama' => 'required',
15: else (tidak ada file foto)	'program' => 'required',
16: Buat array data tanpa fotopesertadidik	'nomor_handphone' => 'required',
17: Update tabel peserta_didik dengan data baru (tanpa ganti foto)	'fotopesertadidik'=>
	'image mimes:jpg,jpeg,png,bmp max:1024',]);
	if (Request()->fotopesertadidik) {
	\$file = Request()->fotopesertadidik;
	\$fileName = Request()->nisd . '.' . \$file->extension();
	\$file->move(public_path('foto_pesertadidik'), \$fileName);
	\$data = [...];

18: Redirect ke route pesertadidik dengan pesan sukses	DB::table('peserta_didik')->where('nispn',\$nispn)->update(\$data);}
19: Selesai	else {\$data = [...]; DB::table('peserta_didik')->where('nispn',\$nispn)->update(\$data);} return redirect()->route('pesertadidik')->with('pesan', 'Data berhasil diupdate !');}



Gambar 2. Flow Graph Edit Data Peserta Didik

Tabel 2. Hitung CC Edit Data Peserta Didik

Keterangan	Jumlah
Node	8
Edge	9
Region	3
Predikat	2
$V(G) = E - N + 2$	$9 - 8 + 2 = 3$
$V(G) = P + 1$	$2 + 1 = 3$

Tabel 3. Penentuan Independent Path Edit Peserta Didik

Path	Flow
Path 1	1 - 2 - 3 - 4 - 5 - 19

Path 2	1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 18 – 19
Path 3	1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 11 – 15 – 16 – 17 – 18 – 19

Tabel 4. Penentuan *Test Case* Edit Data Peserta Didik

Path	Input	Output
Path 1	\$nisd tidak ditemukan	Sistem abort(404) → tampil halaman error
Path 2	\$nisd ditemukan + submit form valid +upload foto baru (jpg/jpeg/png/bmp < 1MB)	File foto disimpan, data peserta didik diupdate dengan foto baru, redirect ke peserta
Path 3	Nisd ditemukan + submit form valid tanpa upload foto	Data peserta didik diupdate tanpa mengubah foto, redirect ke pesertadidik dengan pesan sukses

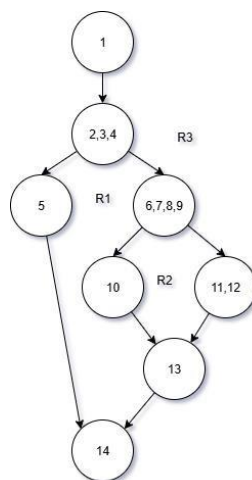
Hasil pengujian white-box pada fitur Edit Data Peserta Didik menggunakan teknik basis path menunjukkan bahwa seluruh jalur logika program telah diuji dan berjalan sesuai yang diharapkan. Berdasarkan perhitungan kompleksitas siklomatik, diperoleh nilai $V(G) = 3$, yang berarti terdapat tiga jalur independen yang harus diuji untuk mencapai cakupan pengujian maksimal. Pada Path 1, sistem berhasil menampilkan halaman 404 ketika data peserta didik tidak ditemukan. Pada Path 2, sistem berhasil memproses pembaruan data dengan upload foto baru, menyimpan file ke direktori yang ditentukan, dan memperbarui data di database. Sementara itu, pada Path 3, sistem juga mampu memperbarui data peserta didik tanpa mengubah foto jika tidak ada file yang diunggah. Dengan demikian, semua jalur pengujian berhasil dilalui, menunjukkan bahwa fitur edit data peserta didik berfungsi dengan baik dan memenuhi alur logika sistem yang telah dirancang.

Pengujian Logika Kirim Tugas

Tabel 5. Alur Logika Kirim Tugas

Alur Logika	Code
1: Mulai (terima request kirim tugas)	public function kirimTugas(Request \$request){
2: Validasi input (id_tugas harus ada di tabel tb_tugas, file_pengumpulan wajib, format pdf/doc/docx/zip, max 2MB)	\$request->validate(['id_tugas' => 'required exists:tb_tugas,id_tugas', 'file_pengumpulan' => 'required file mimes:pdf,doc,docx,zip max:2048',]);
3: Ambil id_akun dari session → cari data siswa di tabel tb_siswa	\$id_akun = session('user')->id_akun;
4: if (\$siswa tidak ditemukan)	\$siswa = DB::table('tb_siswa')->where('id_akun', \$id_akun)->first();
5: Aksi → Redirect back dengan pesan error "Akun belum terdaftar sebagai siswa"	if (!\$siswa) {
6: else (data siswa ada)	return back()->with('error', 'Akun belum terdaftar sebagai siswa.');
7: Upload file → simpan di folder jawaban dengan nama unik	\$file = \$request->file('file_pengumpulan');

8: Cek data pengumpulan tugas di tabel tb_pengumpulan_tugas berdasarkan (id_tugas, id_siswa)	<pre> \$namaFile = time() . '_' . \$file- >getClientOriginalName(); \$file->move(public_path('jawaban'), \$namaFile); </pre>
9: if (sudah ada data pengumpulan)	<pre> \$ccek = DB::table('tb_pengumpulan_tugas') </pre>
10: Update record tb_pengumpulan_tugas (ganti file, update tanggal & updated_at)	<pre> ->where('id_tugas', \$request->id_tugas) ->where('id_siswa', \$siswa->id_siswa) ->first(); </pre>
11: else (belum ada data)	<pre> if (\$ccek) { </pre>
12: Insert record baru ke tb_pengumpulan_tugas	<pre> DB::table('tb_pengumpulan_tugas') </pre>
13: Redirect back dengan pesan sukses "Tugas berhasil dikumpulkan"	<pre> ->where('id_pengumpulan', \$cek- >id_pengumpulan) ->update(['file_pengumpulan' => \$namaFile, 'tanggal_pengumpulan' => now(), 'updated_at' => now(),]); } else { DB::table('tb_pengumpulan_tugas')->insert(['id_tugas' => \$request->id_tugas, 'id_siswa' => \$siswa->id_siswa, 'file_pengumpulan' => \$namaFile, 'tanggal_pengumpulan' => now(), 'created_at' => now(), 'updated_at' => now(),]);} return back()->with('success', 'Tugas berhasil dikumpulkan.');</pre>
14: Selesai	



Gambar 3. Flow Graph Kirim Tugas

Tabel 6. Hitung CC Kirim Tugas

Keterangan	Jumlah
Node	8
Edge	9
Region	3
Predikat	2
$V(G) = E - N + 2$	$9 - 8 + 2 = 3$
$V(G) = P + 1$	$2 + 1 = 3$

Tabel 7. Penentuan *Independent Path* Kirim Tugas

Path	Flow
Path 1	1 – 2 – 3 – 4 – 5 – 14
Path 2	1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 10 – 13 – 14
Path 3	1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 – 11 – 12 – 13 – 14

Tabel 8. Penentuan *Test Case* Kirim Tugas

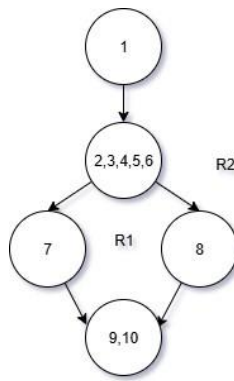
Path	Input	Output
Path 1	Session user ada tapi tidak ditemukan di tb_siswa	Redirect back dengan pesan error “Akun belum terdaftar sebagai siswa”.
Path 2	Session user valid → siswa ditemukan → sudah ada record di tb_pengumpulan_tugas untuk id_tugas	Sile upload → record exiting di update (file, tanggal_pengumpulan, update_at) → redirect back dengan pesan sukses “Tugas berhasil dikumpulkan”.
Path 3	Session user valid → siswa ditemukan → belum ada record di tb_pengumpulan_tugas untuk id_tugas	File diupload → record baru di insert → redirect back dengan pesan sukses “Tugas berhasil dikumpulkan”.

Pengujian white-box pada fitur Kirim Tugas menghasilkan tiga jalur logika independen dengan nilai *cyclomatic complexity* sebesar 3. Seluruh jalur berhasil diuji dengan hasil sesuai harapan, pada path 1 sistem menolak akun yang tidak terdaftar sebagai siswa, pada path 2 berhasil memperbarui file tugas jika sudah ada data sebelumnya, dan pada path 3 dapat menyimpan data baru saat pengumpulan pertama kali. Proses validasi, upload file, serta insert dan update data berjalan dengan baik, menunjukkan bahwa fitur ini berfungsi sesuai dengan logika yang dirancang.

Pengujian Logika Simpan Nilai

Tabel 9. Alur Logika Simpan Nilai

Alur Logika	Code
1: Mulai (terima request simpan nilai)	public function simpan(Request \$request){
2: Validasi input (id_tugas, id_siswa, nilai, komentar)	\$request->validate([
3: Ambil tahun ajaran aktif dari tabel tb_tahun_ajaran	'id_tugas' => 'required integer',
4: Simpan id_tahun_ajaran (jika ada, ambil id; jika tidak, null)	'id_siswa' => 'required integer',
5: Cek data nilai di tabel tb_nilai berdasarkan (id_tugas, id_siswa, id_tahun_ajaran)	'nilai' => 'required integer min:0 max:100',
6: if (\$cek ada?)	'komentar' => 'nullable string max:255',]);
7: Aksi Update → update nilai & komentar di tabel tb_nilai	// Ambil tahun ajaran aktif
8: Aksi Insert → simpan nilai baru ke tabel tb_nilai	\$tahunAjaran = DB::table('tb_tahun_ajaran')->where('is_active', 1)->first();
9: Redirect sukses → "Nilai berhasil disimpan"	\$id_tahun_ajaran = \$tahunAjaran ? \$tahunAjaran->id_tahun_ajaran : null;
10: Selesai	\$cek = DB::table('tb_nilai')->where([
	['id_tugas', '=', \$request->id_tugas],
	['id_siswa', '=', \$request->id_siswa],
	['id_tahun_ajaran', '=', \$id_tahun_ajaran],
])>first();
	if (\$cek) {
	// Update nilai jika sudah ada
	DB::table('tb_nilai')->where('id_nilai', \$cek->id_nilai)->update([
	'nilai' => \$request->nilai,
	'komentar' => \$request->komentar,
	'updated_at' => now(),]);
	} else {
	// Simpan nilai baru
	DB::table('tb_nilai')->insert([
	'id_tugas' => \$request->id_tugas,
	'id_siswa' => \$request->id_siswa,
	'id_tahun_ajaran' => \$id_tahun_ajaran,
	'nilai' => \$request->nilai,
	'komentar' => \$request->komentar,
	'created_at' => now(),
	'updated_at' => now(),]); }
	return redirect()->back()->with('success', 'Nilai berhasil disimpan!');}}



Gambar 4. *Flow Graph* Simpan Nilai

Tabel 10. Hitung CC Simpan Nilai

Keterangan	Jumlah
Node	5
Edge	5
Region	2
Predikat	1
$V(G) = E - N + 2$	$5 - 5 + 0 + 2 = 2$
$V(G) = P + 1$	$1 + 1 = 2$

Tabel 11. Penentuan *independent path* Simpan Nilai

Path	Flow
Path 1	1 – 2 – 3 – 4 – 5 – 6 – 7 – 9 – 10
Path 2	1 – 2 – 3 – 4 – 5 – 6 – 8 – 9 – 10

Tabel 12. Penentuan *Test Case* Simpan Nilai

Path	Input	output
Path 1	Input valid (id_tugas, id_siswa, nilai, komentar) dan data nilai sudah ada di tabel tb_nilai untuk kombinasi (id_tugas, id_siswa, id_tahun_ajaran)	Sistem melakukan update pada record tb_nilai (nilai & komentar diperbarui, update_at berubah), lalu redirect back dengan pesan sukses “Nilai berhasil disimpan”
Path 2	Input valid (id_tugas, id_siswa, nilai, komentar) dan data nilai belum ada di tabel tb_nilai	Sistem melakukan insert record bar uke tb_nilai (isi semua field termasuk created_at, updated_at), lalu redirect back dengan pesan sukses “Nilai berhasil disimpan”

Pengujian white-box pada fitur Simpan Nilai menghasilkan dua jalur independen dengan nilai *cyclomatic complexity* sebesar 2. Pada Path 1, sistem berhasil melakukan update nilai dan komentar jika data sudah ada, sedangkan pada Path 2, sistem menyimpan data baru saat belum ada nilai sebelumnya. Kedua jalur berhasil dieksekusi dengan benar, membuktikan bahwa proses validasi, pengecekan data, serta penyimpanan (insert dan update) berjalan sesuai logika yang dirancang.

SIMPULAN

Berdasarkan hasil pengujian white box yang menggunakan teknik basis path pada tiga fitur utama dalam sistem pembelajaran digital di SKB Kabupaten Subang, dapat disimpulkan bahwa seluruh jalur logika program telah diuji secara menyeluruh dan menunjukkan hasil yang sesuai dengan harapan. Pengujian pada fitur edit data peserta didik, kirim tugas, dan simpan nilai masing-masing menghasilkan nilai kompleksitas siklomatik yang valid dan berhasil mengidentifikasi jalur-jalur independen yang mewakili seluruh kemungkinan alur eksekusi program. Seluruh test case yang disusun berdasarkan jalur tersebut dapat dijalankan dengan hasil aktual yang sesuai dengan output yang diharapkan dan menunjukkan bahwa sistem memiliki tingkat keandalan logika yang tinggi. Dengan demikian, penerapan white-box testing berbasis basis path terbukti efektif dalam menjamin kestabilan dan akurasi sistem pembelajaran digital, serta memberikan kontribusi terhadap peningkatan mutu layanan pendidikan nonformal secara digital di lingkungan SKB Kabupaten Subang.

DAFTAR PUSTAKA

- Erwin, E., Suardi, C., & Nasir, A. (2023). *Transformasi Digital*. Sonpedia Publishing Indonesia.
- Irfan, F. (2019). Pengembangan E-Learning untuk Pembelajaran Sekolah Nonformal. *Delta: Jurnal Ilmiah Pendidikan Matematika*, 7(2), 29–36.
- Kurniawan, F., Khrisnawati, E. A., Hadiwiyan, R., & Fitri, A. S. (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.
- Londjo, M. F. (2021). Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login. *Jurnal Siliwangi Seri Sains Dan Teknologi*, 7(2).
- Miftah, M. (2013). Fungsi, dan peran media pembelajaran sebagai upaya peningkatan kemampuan belajar siswa. *Kwangsan: Jurnal Teknologi Pendidikan*, 1(2), 95–105.
- Nugraha, W. A. (2022). Pengujian White Box Berbasis Path Pada Form Autentikasi Berbasis Mobile. *Jurnal Siliwangi Seri Sains Dan Teknologi*, 8(2).
- Nurwicaksono, M. A., Lisa, I. N., Tiara, A. R., & Sidik, R. (2024). Optimasi sistem informasi konsultasi hukum melalui pendekatan pengujian kombinasi white-box dan black-box. *Jurnal Manajemen Informatika (JAMIKA)*, 14(1), 1–15.
- Pranata, F. N., Kom, S., Pradana, F., & Astoto, T. (2016). Pengembangan Sistem Perhitungan Kompleksitas Kode Sumber Berdasarkan Metrik Halstead dan Cyclomatic Complexity. *Universitas Brawijaya*.
- Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. (2020). Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path. *J. Inform. Univ. Pamulang*, 5(2), 111.
- Putra, A. P., & Nuryasin, I. (2024). Pengujian sistem informasi monitoring daya kebawah menggunakan white box testing dengan teknik basis path. *INFOTECH: Jurnal Informatika & Teknologi*, 5(1), 63–75.

- 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. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 10(3), 2366–2377.
- Putri, F. A., Marthasari, G. I., & Nuryasin, I. (2023). Rancang Bangun Perangkat Lunak Perhitungan Metrik Cyclomatic Complexity Berdasarkan Control Flow Graph Berbasis Web. *Jurnal Repositor*, 5(1).
- Shiddiq, M. I. (2022). Implementasi White Box Testing Berbasis Path Pada Form Login Aplikasi Berbasis Web. *Jurnal Siliwangi Seri Sains Dan Teknologi*, 8(1).
- 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.
- Wahyuni, A. E. T., Adawiyah, R., & Makhfudloh, I. I. (n.d.). *Graph Applications in Software Engineering*.