

PENGUJIAN FUNGSIONAL DAN STRUKTURAL APLIKASI PENGAJUAN CUTI DENGAN METODE *BLACK BOX* DAN *WHITE BOX*

Jatnika Fahmi Idris¹, Yahya Dian Prastyo²,
Reza Hardiansyah³, Albertus Magnus Foresta Noventona⁴,
Fikri Haikal⁵, Harya Gusdevi⁶, Muchammad Naseer⁷
Departemen Teknik Informatika^{1,2,3,4,5,6,7}
Universitas Teknologi Bandung^{1,2,3,4,5,6,7}
haryagusdevi@utb-univ.ac.id

Abstrak

Efisiensi manajemen sumber daya manusia (SDM) di era digital sangat bergantung pada sistem yang andal, termasuk dalam pengelolaan cuti karyawan. Proses pengajuan cuti secara manual sering kali menghadapi berbagai kendala seperti keterlambatan, kesalahan administratif, dan ketidakefisienan operasional. Penelitian ini bertujuan untuk mengevaluasi keandalan dan fungsionalitas aplikasi pengajuan cuti berbasis web sebagai solusi modern untuk mengatasi permasalahan tersebut. Metode penelitian yang digunakan adalah pengujian perangkat lunak berlapis dengan pendekatan *Black Box* dan *White Box* dalam kerangka *Rapid Application Development* (RAD). Metodologi RAD dipilih karena kemampuannya dalam mempercepat proses pengembangan dan adaptif terhadap perubahan kebutuhan pengguna. Pengujian *Black Box* dilakukan untuk menilai kinerja aplikasi dari sisi fungsional tanpa melihat struktur internal, sementara pengujian *White Box* difokuskan pada analisis logika program, alur kontrol, dan kompleksitas siklomatik untuk memastikan integritas kode secara menyeluruh. Hasil pengujian menunjukkan bahwa aplikasi mampu berfungsi sesuai dengan spesifikasi dan efektif dalam mendukung proses pengajuan cuti karyawan. Namun, ditemukan beberapa kekurangan dalam validasi masukan, seperti tidak tertangkapnya kolom kosong atau tanggal yang tidak logis, yang menghasilkan umpan balik sistem yang kurang informatif. Hal ini menunjukkan perlunya perbaikan pada mekanisme validasi dan antarmuka pengguna untuk meningkatkan pengalaman pengguna secara keseluruhan. Studi ini menegaskan pentingnya penerapan strategi pengujian berlapis untuk menjamin kualitas perangkat lunak, terutama pada sistem SDM yang menangani data dan proses krusial. Kombinasi metode pengujian *Black Box* dan *White Box*, yang dilaksanakan dalam kerangka RAD, terbukti mampu menghasilkan aplikasi yang tangguh, efisien, dan responsif terhadap kebutuhan pengguna. Temuan ini relevan sebagai referensi dalam pengembangan sistem serupa di bidang manajemen SDM berbasis digital.

Kata kunci : Pengujian perangkat lunak, Black Box, White Box, aplikasi pengajuan cuti, keandalan perangkat lunak.

Abstract

Efficient human resource management (HRM) in the digital era increasingly relies on reliable systems, particularly in managing employee leave requests. Manual leave request processes are often hindered by delays, administrative errors, and operational inefficiencies. This study aims to evaluate the reliability and functionality of a web-based leave application system as a modern solution to these challenges. The research employs a layered software testing approach using Black Box and White Box methods within the Rapid Application Development (RAD) framework. RAD was chosen for its ability to accelerate development processes and adapt to evolving user needs. Black Box testing evaluates the application's functionality from the user's perspective without examining internal code structures, while White Box testing focuses on analyzing program logic, control flow, and cyclomatic complexity to ensure code integrity. The testing results indicate that the application functions in accordance with predefined specifications and effectively supports employee leave processes. However, some weaknesses were identified in input validation, such as undetected empty fields or illogical date entries, resulting in uninformative system feedback. These findings suggest the need for improvement in validation mechanisms and user interface design to enhance the overall user experience. This study highlights the importance of implementing layered validation strategies to ensure software quality, particularly for HRM systems that handle sensitive data and critical processes. The combination of Black Box and White Box testing methods, applied within the RAD framework, has proven effective in producing a robust, efficient, and user-responsive application. These findings offer valuable insights for the development of similar systems in the field of digital HR management.

Keywords : Software testing, Black Box, White Box, leave application, software reliability.

I. PENDAHULUAN

Dalam era digital yang berkembang pesat saat ini, efisiensi serta otomatisasi dalam proses bisnis menjadi elemen yang sangat penting dan tidak bisa diabaikan dalam upaya meningkatkan produktivitas kerja serta kesejahteraan karyawan. Salah satu komponen krusial dalam praktik manajemen sumber daya manusia (SDM) yang seringkali menjadi perhatian adalah pengelolaan cuti karyawan. Proses pengajuan dan persetujuan cuti yang dilakukan secara efektif, sistematis, dan transparan tidak hanya membantu menjaga kelancaran aktivitas operasional perusahaan, tetapi juga memberikan kenyamanan bagi karyawan dalam merencanakan dan menikmati waktu istirahat mereka secara optimal tanpa hambatan administratif yang rumit.

Prosedur pengajuan cuti secara manual seringkali dihadapkan pada berbagai tantangan, termasuk keterlambatan dalam pemrosesan, potensi kesalahan administratif, dan ketidakefisienan operasional yang signifikan. Mengatasi permasalahan ini, aplikasi pengajuan cuti berbasis web atau *mobile* muncul sebagai solusi unggulan. Aplikasi semacam

ini memungkinkan proses pengajuan cuti menjadi lebih cepat, akurat, dan mudah diakses oleh karyawan kapan saja dan di mana saja, secara drastis mengurangi hambatan yang melekat pada sistem manual.

Pengembangan aplikasi pengajuan cuti ini, pengujian perangkat lunak memegang peranan yang sangat vital. Pengujian ini memastikan bahwa aplikasi tidak hanya berfungsi sesuai dengan spesifikasi yang ditetapkan, tetapi juga memberikan pengalaman pengguna yang positif dan bebas dari masalah. Terdapat berbagai pendekatan dalam pengujian perangkat lunak, di antaranya adalah pengujian *Black Box* dan *White Box*. Pengujian *Black Box* berfokus pada validasi fungsionalitas aplikasi berdasarkan spesifikasi tanpa melihat struktur internal kode, memastikan bahwa aplikasi memenuhi kebutuhan pengguna dan bekerja sesuai yang diharapkan dalam berbagai skenario penggunaan. Di sisi lain, pengujian *White Box* melibatkan pemeriksaan langsung terhadap kode sumber untuk memastikan bahwa semua jalur dan logika dalam program berfungsi dengan benar, memungkinkan pengembang untuk mengidentifikasi dan memperbaiki kesalahan pada level kode, sehingga meningkatkan keandalan dan performa aplikasi secara keseluruhan.

Penelitian ini bertujuan untuk mengevaluasi secara komprehensif keandalan dan fungsionalitas aplikasi pengajuan cuti berbasis web. Pertanyaan utama yang ingin dijawab adalah bagaimana aplikasi ini dapat diuji secara menyeluruh menggunakan kombinasi metode pengujian *Black Box* dan *White Box* dalam kerangka *Rapid Application Development* (RAD), serta apa saja kekurangan yang teridentifikasi dari pengujian tersebut. Dengan demikian, tujuan penelitian ini adalah untuk secara spesifik mengevaluasi keandalan dan fungsionalitas aplikasi pengajuan cuti berbasis web sebagai solusi modern untuk mengatasi permasalahan manajemen cuti manual.

Penelitian ini diharapkan dapat memberikan berbagai manfaat yang signifikan. Dari sisi praktis, hasil studi ini dapat menjadi acuan yang bernilai dalam perancangan maupun pengembangan sistem manajemen sumber daya manusia (SDM) berbasis digital yang serupa di masa depan. Penelitian ini menekankan pentingnya penerapan strategi pengujian berlapis sebagai langkah krusial untuk memastikan kualitas, stabilitas, dan keandalan perangkat lunak yang digunakan. Lebih dari itu, secara umum penelitian ini juga memberikan kontribusi nyata dalam upaya meningkatkan efisiensi dan efektivitas pengelolaan SDM melalui pemanfaatan teknologi yang tidak hanya canggih, tetapi juga telah melewati tahapan uji yang menyeluruh dan sistematis.

II. TINJAUAN PUSTAKA

1. Penelitian Terdahulu

Sistem Informasi Pariwisata Berbasis *Website* Pada Taman Nasional Gunung Ciremai Jawa Barat. Penelitian oleh Utarki, Pratama, dan Hellyana pada tahun 2020 [1] ini bertujuan untuk merancang dan membangun sistem informasi pariwisata berbasis *website* guna meningkatkan pendapatan daerah serta jumlah kunjungan wisata di Taman Nasional Gunung Ciremai Jawa Barat. Dalam pengembangannya, mereka menggunakan model Waterfall, memanfaatkan teknologi seperti PHP, HTML, CSS, *framework* Laravel, dan basis data MySQL. Sistem ini dirancang untuk mencakup berbagai fitur, termasuk pemesanan paket tur, informasi objek wisata, laporan, serta fitur *login* dan *dashboard* admin. Hasil dari penelitian ini adalah sebuah sistem informasi pariwisata berbasis *website* yang fungsional, mampu menyediakan informasi dan memfasilitasi manajemen pariwisata secara efektif.

Studi ini relevan karena menunjukkan implementasi sistem informasi berbasis *website* menggunakan tumpukan teknologi yang serupa dengan aplikasi cuti dalam penelitian ini. Keberadaan fitur *login* dan *dashboard* admin dalam sistem pariwisata juga memberikan paralel fungsional yang menarik. Perbedaan signifikan terletak pada metodologi pengembangan; penggunaan model Waterfall dalam penelitian ini berbanding terbalik dengan kerangka RAD pada penelitian utama. Perbandingan antara model Waterfall yang digunakan dalam studi ini dan kerangka RAD dalam penelitian utama menyoroti evolusi dan diversifikasi metodologi pengembangan perangkat lunak. Hal ini menunjukkan bahwa pilihan metodologi tidak bersifat universal, melainkan disesuaikan dengan karakteristik proyek, seperti kebutuhan akan kecepatan, adaptasi terhadap perubahan, atau kompleksitas persyaratan awal. Model Waterfall adalah model sekuensial yang cocok untuk proyek dengan persyaratan stabil, sementara RAD, sebaliknya, lebih adaptif dan cepat. Aplikasi pariwisata mungkin memiliki persyaratan yang lebih statis, sementara aplikasi SDM seperti cuti mungkin memerlukan iterasi cepat karena perubahan kebijakan atau umpan balik pengguna. Perbedaan ini mengindikasikan bahwa pemilihan metodologi adalah keputusan strategis yang memengaruhi efisiensi dan hasil proyek.

Perancangan Sistem Informasi Penginputan *database* Mahasiswa Berbasis Web, Sari, Jannah, Meuraxa, Syahfitri, dan Omar pada tahun 2022 [2] melakukan penelitian ini dengan tujuan merancang sistem informasi berbasis web yang akan memudahkan perguruan tinggi dalam menginput data mahasiswa. Penelitian ini mengadopsi metode kualitatif, di mana *website* dikembangkan menggunakan bahasa pemrograman HTML, CSS, PHP, dan JavaScript. Data yang diinputkan kemudian disimpan dalam basis data XAMPP. Sistem ini juga mencakup fungsi *login* dan penginputan data, serta pengujian dilakukan menggunakan metode *Black Box testing*. Dengan adanya *website* yang dirancang, mahasiswa dapat melakukan *login* dan penginputan data secara mandiri, yang secara signifikan memudahkan proses penginputan data bagi perguruan tinggi.

Studi ini sangat relevan karena fokusnya pada perancangan dan implementasi sistem informasi berbasis web dengan fungsionalitas inti (*login* dan *input* data) dan tumpukan teknologi yang hampir identik (HTML, CSS, PHP, JavaScript, XAMPP, MySQL) dengan aplikasi cuti. Lebih penting lagi, penelitian ini secara eksplisit

menggunakan *Black Box testing* untuk evaluasi, memberikan dasar perbandingan metodologi pengujian yang kuat dan menegaskan praktik umum pengujian fungsional pada aplikasi web. Penggunaan *Black Box testing* dalam penelitian ini untuk memvalidasi *input* data, serupa dengan pengujian input pada aplikasi cuti, menunjukkan adanya tantangan umum dalam memastikan validasi *input* yang *robust* pada aplikasi web. Hal ini mengindikasikan bahwa meskipun fungsionalitas inti tercapai, penanganan kasus-kasus *edge* atau *input* yang tidak valid seringkali menjadi area yang memerlukan perhatian lebih dalam pengembangan dan pengujian. Baik aplikasi cuti maupun sistem input data mahasiswa melibatkan input pengguna. Jika kedua studi menghadapi tantangan dalam validasi input, ini menunjukkan pola di mana pengembang mungkin fokus pada alur "*happy path*" dan kurang memperhatikan skenario *input* yang tidak valid atau kosong, yang merupakan celah umum dalam keamanan dan keandalan aplikasi web.

2. Website

Website adalah seluruh informasi yang disediakan melalui jalur internet sehingga bisa diakses di seluruh dunia, selama terkoneksi dengan jaringan internet. *Website* juga merupakan komponen yang dinamis terdiri dari teks, gambar, animasi, video sehingga membuat tampilan menarik untuk dikunjungi. Secara garis besar, *website* dibagi menjadi tiga jenis:

- a. *Website* Statis, adalah web yang tampilan halamannya tetap, jika akan melakukan perubahan pada halaman yang akan diubah maka halaman tersebut diubah secara manual dengan cara mengedit langsung dari kode programnya.
- b. *Website* Dinamis, adalah *website* yang dapat diubah tampilannya melalui webnya langsung tanpa harus mengubah dan mengedit kode programnya.
- c. *Website* Interaktif, adalah web yang digunakan untuk interaksi sekumpulan komunitas atau forum [1].

3. Basis Data

Basis Data (*database*) adalah sekumpulan informasi yang disimpan secara sistematis pada komputer sehingga dapat diperiksa oleh program komputer untuk mengambil informasi dari *database*. *Database* merupakan salah satu komponen yang penting dalam sistem informasi. Selain itu, *database* merupakan basis dalam menyediakan basis dalam menyediakan informasi informasi bagi para pemakai. Penerapan bagi para pemakai. Penerapan *database* dalam sistem informasi disebut dalam sistem informasi disebut dengan *database system* [2].

4. Visual Studio Code

Visual Studio Code (VS Code) adalah editor teks gratis dari Microsoft, tersedia untuk berbagai sistem operasi termasuk Windows, Linux, dan MacOS. Kelebihan utama dari Visual Studio Code terletak pada ketersediaan ekstensi yang beragam. Dengan menambahkan ekstensi, pengguna dapat menambah dukungan untuk bahasa pemrograman baru, mengubah tema atau warna, serta terhubung dengan berbagai layanan pendukung lainnya [3].

5. HTML

HTML (*HyperText Markup Language*) merupakan bahasa markup standar yang digunakan secara luas dalam pengembangan dan penyusunan halaman web. Bahasa ini berperan penting sebagai fondasi utama dari setiap situs web, karena memungkinkan pengembang untuk menentukan struktur, tata letak, dan berbagai elemen konten yang ditampilkan kepada pengguna. HTML bekerja dengan menggunakan elemen-elemen dan tag-tag khusus yang berfungsi untuk menandai bagian-bagian tertentu dari halaman, seperti paragraf teks, gambar, tautan (*hyperlink*), tabel, daftar, serta elemen interaktif seperti formulir isian. Setiap tag memiliki fungsi spesifik yang membantu browser memahami bagaimana informasi tersebut harus disusun dan ditampilkan kepada pengguna akhir. Seiring perkembangan teknologi web, HTML juga terus mengalami pembaruan, dengan versi terbaru seperti HTML5 yang mendukung fitur-fitur modern seperti audio, video, grafik vektor, dan integrasi aplikasi web yang lebih dinamis. [4].

6. CSS

CSS (*Cascading Style Sheets*), adalah bahasa pemrograman yang digunakan untuk mengatur dan mendeskripsikan tampilan visual dari dokumen yang ditulis dalam bahasa markup, seperti HTML, XML, SVG, maupun XUL. CSS memungkinkan pengembang web untuk memisahkan konten dari desain, sehingga struktur halaman tetap rapi dan mudah dipelihara dalam jangka panjang. Dengan CSS, pengembang memiliki kendali penuh atas berbagai aspek presentasi halaman web. Mereka dapat menyesuaikan warna latar belakang dan teks, memilih jenis dan ukuran *font*, mengatur tata letak elemen secara presisi, serta menerapkan berbagai efek visual seperti garis tepi, bayangan, spasi antar elemen, dan animasi. Kemampuan ini tidak hanya membuat tampilan halaman menjadi lebih menarik dan profesional, tetapi juga berkontribusi pada peningkatan aksesibilitas dan pengalaman pengguna [5] [6].

7. Javascript

JavaScript, adalah bahasa skrip sisi klien yang sangat kuat. JavaScript digunakan terutama untuk meningkatkan interaksi pengguna dengan halaman web. Dengan kata lain, dengan menggunakan JavaScript dapat membuat halaman web lebih hidup dan interaktif, dengan bantuan JavaScript. JavaScript bisa membuat berbagai fitur mulai dari yang sederhana sampai dengan yang paling kompleks, misalnya *layout*, *gallery*, tombol,

causels, dan lain sebagainya. Jika dikembangkan lebih lanjut, JavaScript dapat digunakan untuk membuat baik animasi dua dimensi maupun tiga dimensi, bahkan aplikasi yang berhubungan dengan basis data JavaScript juga digunakan secara luas dalam pengembangan game dan pengembangan aplikasi *Mobile* [5].

8. PHP

PHP dikenal sebagai sebuah bahasa scripting yang menyatu dengan tag-tag HTML yang di eksekusi di server dan digunakan untuk membuat halaman web yang dinamis seperti *guestbook*, statistik pengunjung, *polling*, *email*, dan masih banyak lagi. Konsep kerja PHP diawali dengan satu permintaan suatu halaman web oleh *browser*. Berdasarkan URL (*Uniform Resource Locator*) atau dikenal dengan alamat internet, *browser* mendapat alamat dari *web server*, mengidentifikasi alamat yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *web server* [7].

9. Pengujian *Black Box*

Pengujian *Black Box* adalah pengujian yang berfokus pada pengujian persyaratan fungsional perangkat lunak untuk meyakinkan bahwa apakah fungsi-fungsi yang ada sudah berfungsi dengan baik dan tidak memperhatikan struktur logika internal perangkat lunak. Juga memastikan bahwa program tidak *error*, bisa dijalankan dengan mencoba beberapa transaksi (*input*, *edit*, *delete*, *update data*) dapat berjalan, dan menghasilkan *output* sesuai kebutuhan. Pengujian *Black Box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Ini semua berarti, ketika kita menjalankan program, maka hasil dari running program tersebut sesuai dengan yang diharapkan. Pengujian *Black Box* menemukan kesalahan fungsi-fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses *database* eksternal, kesalahan kinerja, dan inisiasi dan kesalahan terminasi [8].

10. Pengujian *White Box*

White Box Testing, merupakan salah satu metode pengujian perangkat lunak yang difokuskan pada pemeriksaan mendalam terhadap struktur internal dari aplikasi. Pengujian ini dilakukan dengan tujuan untuk memahami secara menyeluruh bagaimana perangkat lunak bekerja dari dalam, termasuk logika, jalur alur program, dan struktur kode yang digunakan. Berbeda dari pengujian *Black Box* yang hanya menilai keluaran berdasarkan masukan tanpa mengetahui bagaimana proses di dalamnya, *White Box Testing* menuntut pengujian untuk memiliki pemahaman tentang kode sumber dan arsitektur sistem yang diuji. Pengujian ini biasanya diterapkan pada tiga lapisan utama aplikasi, yaitu *presentation logic* (tampilan antarmuka pengguna), *business logic* (logika proses utama), dan *persistence layer* (interaksi dengan basis data atau penyimpanan data lainnya). Melalui pendekatan ini, pengujian dapat mendeteksi kesalahan logika, kondisi yang tidak ditangani, serta jalur program yang tidak efisien atau tidak digunakan. Selain itu, teknik ini sangat berguna dalam mengevaluasi kualitas kode, tingkat kompleksitas program, dan cakupan pengujian (*code coverage*), seperti jalur perulangan, percabangan, serta pernyataan kondisional [8].

11. XAMPP

XAMPP adalah perangkat lunak sumber terbuka yang dikembangkan oleh teman-teman *Apache*. Paket perangkat lunak XAMPP berisi distribusi *Apache* untuk server *Apache*, *MariaDB*, *PHP*, dan *Perl* dan itu pada dasarnya adalah tan rumah lokal atau server lokal. Server lokal ini berfungsi di komputer desktop atau laptop sendiri. Penggunaan XAMPP adalah untuk menguji klien atau *Website* Anda sebelum mengunggahnya ke server web jarak jauh. Perangkat lunak server XAMPP ini memberi Anda lingkungan yang cocok untuk menguji proyek *MySQL*, *PHP*, *Apache* dan *Perl* di komputer lokal. Bentuk lengkap XAMPP adalah X singkatan dari *Cross-platform*, (A) server *Apache*, (M) *MariaDB*, (P) *PHP* dan (P) *Perl*. *Cross-platform* biasanya berarti dapat berjalan di komputer mana saja dengan sistem operasi apa pun [9].

12. SQL

Bahasa Query atau SQL (*Structured Query Language*) merupakan bahasa komputer yang digunakan untuk mengambil, mengelola, dan memanipulasi data dalam basis data relasional. SQL memungkinkan pengguna menjalankan perintah seperti menampilkan, menambahkan, mengubah, dan menghapus data dalam tabel. Beberapa sistem manajemen basis data (DBMS) memiliki varian SQL dengan penyesuaian tertentu, seperti T-SQL untuk *SQL Server*, PL/SQL untuk *Oracle*, dan JET SQL untuk *Microsoft Access*. Meskipun berbeda nama, varian-varian ini tetap berbasis pada struktur dan prinsip dasar SQL [10].

13. MySQL

MySQL adalah salah satu server basis data yang sangat populer dan banyak digunakan, terutama karena mendukung penggunaan SQL sebagai bahasa utama untuk berinteraksi dengan sistem basis datanya. Sebagai sistem manajemen basis data relasional, MySQL menerapkan konsep-konsep dasar seperti tabel, *record* (baris data), dan *field* (kolom) dalam pengelolaan datanya, sehingga memudahkan pengguna dalam menyusun dan mengakses informasi secara terstruktur [11].

III. ANALISIS DAN PERANCANGAN

1. Metode Pengembangan

Rapid Application Development (RAD) adalah salah satu metode pengembangan sistem yang proses pengerjaannya yang cepat. Biasanya metode yang normal membutuhkan minimal 180 hari tapi dengan RAD

hanya butuh 30-90 hari [12], yang dimana tujuannya untuk menghasilkan sebuah produk atau perangkat lunak melalui kolaborasi yang erat anatar semua pihak yang terlibat [13].

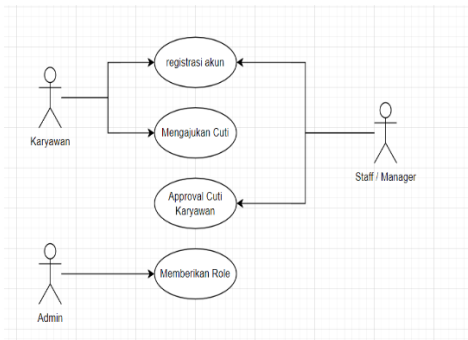
Berikut adalah diagram yang menggambarkan proses pengembangan perangkat lunak menggunakan metodologi *Rapid Application Development (RAD)*. Diagram ini menunjukkan empat tahap utama RAD yang saling berurutan, yaitu Perencanaan Kebutuhan, Desain Pengguna, Konstruksi, dan Cutover. Alur proses digambarkan dengan panah yang menghubungkan setiap tahap, mengilustrasikan bagaimana pengembangan sistem bergerak maju dari awal hingga akhir dalam pendekatan RAD.



Gambar 1. Diagram Alir Proses RAD

2. Perancangan Desain

Use Case Diagram yaitu sebuah interaksi yang saling berhubungan antara Sistem dan Aktor *Use Case Diagram* menggambarkan jenis interaksi antara aktor dengan sistem itu sendiri. Berikut gambar *Use Case Diagram* :



Gambar 2. Use Case Diagram

Pada gambar diatas menunjukkan interaksi antara *User* dan Sistem mulai dari daftar akun, *login*, mengajukan cuti dan lain-lain. Penjelasan aktor:

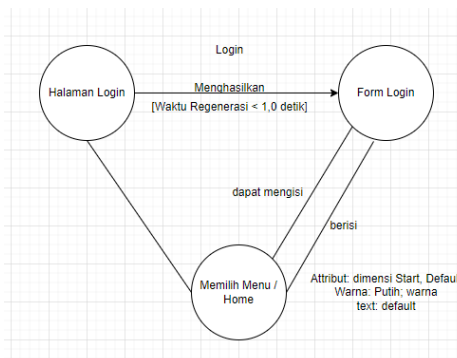
- a. Karyawan : berfungsi sebagai aktor *user* yang dapat melakukan *SignUp*, *Login* dan mengajukan cuti.
- b. Staff / Manager : berfungsi sebagai aktor *user* yang dapat melakukan *SignUp*, *Login* dan *approval* cuti.
- c. Admin : berfungsi untuk memberikan *role*.

3. Pengujian Aplikasi

- a. Pengujian *Black Box*
 - a) *Login*

1) Pengujian Berbasis Grafik (*Graph Based Testing*)

Pada gambar dibawah ini adalah pengujian berbasis grafik.



Gambar 3. Pengujian *Login* Berbasis Grafik

2) Tabel Data Uji

Pada Tabel dibawah ini menjelaskan data uji partisi.

TABEL I
TABEL PARTISI

No Uji	Field	Data Uji	Kode Partisi	Hasil
L11	<i>username</i>	diisi dengan <i>username</i> terdaftar	T101	V
L21	<i>username</i>	diisi dengan <i>username</i> belum terdaftar	T102	IV
L31	<i>password</i>	diisi dengan <i>password</i> terdaftar	T201	V
L41	<i>password</i>	diisi dengan <i>password</i> belum terdaftar	T202	IV

Pada Tabel dibawah ini menjelaskan data uji tipe uji.

TABEL II
TABEL TIPE UJI

No Uji	Data Uji	Tipe Uji	Output	Hasil
TL1	text field <i>username</i>	V	“Please enter a <i>password</i> .”	berhasil
TL2	text field <i>username</i>	IV	“Please enter <i>username</i> .”	berhasil
TL3	text field <i>password</i>	V	“Please enter a <i>username</i> .”	berhasil
TL4	text field <i>password</i>	IV	“Please enter a <i>password</i> .”	berhasil

3) Metode Partisi Ekuivalensi (*Equivalence Partitioning*)

Pada Tabel dibawah ini menjelaskan pengujian partisi ekuivalensi

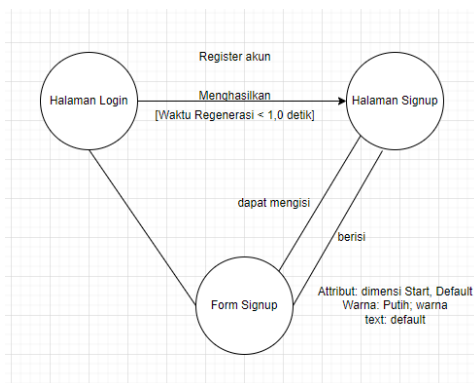
TABEL III
TABEL PENGUJIAN PARTISI EKUIVALENSI

No.	Kondisi Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	mengosongkan semua input lalu submit	<i>Username:</i> <i>Password:</i>	harus semua diisi	sesuai harapan	valid
2	<i>username</i> atau <i>password</i> tidak sesuai (terdaftar)	<i>Username</i> : tes <i>Password</i> : tes salah	Invalid <i>username</i> or <i>password</i>	sesuai harapan	valid

b) Register

1) Pengujian Berbasis Grafik (*Graph Based Testing*)

Pada Gambar dibawah ini adalah pengujian register berbasis grafik.



Gambar 4. Pengujian Register Berbasis Grafik

2) Tabel Data Uji

Pada Tabel dibawah ini menjelaskan data uji partisi untuk menu register.

TABEL IV
 TABEL PARTISI

No Uji	Field	Data Uji	Kode Partisi	Hasil
R11	<i>username</i>	diisi dengan <i>username</i> terdaftar	T101	V
R21	<i>username</i>	diisi dengan <i>username</i> belum terdaftar	T102	IV
R31	<i>password</i>	diisi dengan <i>password</i> terdaftar	T201	V
R41	<i>password</i>	diisi dengan <i>password</i> belum terdaftar	T202	IV

Pada Tabel dibawah ini menjelaskan data uji tipe uji untuk menu register.

TABEL V
 TABEL TIPE UJI

No Uji	Data Uji	Tipe Uji	Output	Hasil
TR1	text field <i>username</i>	V	“Please enter a <i>password</i> .”	berhasil
TR2	text field <i>username</i>	IV	“Please enter <i>username</i> .”	berhasil
TR3	text field <i>password</i>	V	“Please enter a <i>username</i> .”	berhasil
TR4	text field <i>password</i>	IV	“Please enter a <i>password</i> .”	berhasil

3) Tabel Data Uji

Pada Tabel dibawah ini menjelaskan pengujian partisi ekuivalensi untuk menu register

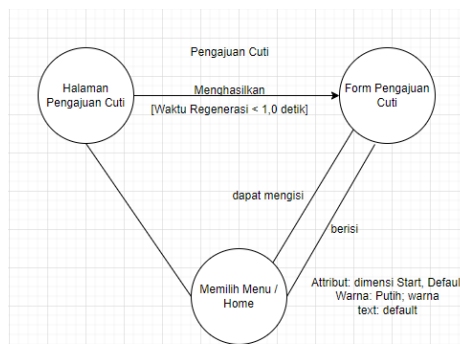
TABEL VI
 TABEL PENGUJIAN PARTISI EKUIVALENSI

No.	Kondisi Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	mengosongkan semua input lalu submit	<i>Username</i> : <i>Fullname</i> : <i>Password</i> : confirm <i>password</i> :	harus semua diisi	Sesuai harapan	valid
2	<i>password</i> dan confirm <i>password</i> tidak sesuai	<i>username</i> : tes <i>fullname</i> : tes <i>Password</i> : tes123 confirm <i>password</i> : tes111	confirm <i>password</i> harus sesuai dengan <i>password</i>	Sesuai harapan	valid

c) Pengajuan Cuti

1) Pengujian Berbasis Grafik (*Graph Based Testing*)

Pada Gambar dibawah ini menjelaskan pengujian berbasis grafik untuk pengajuan cuti.



Gambar 5. Pengujian Pengajuan Cuti Berbasis Grafik

2) Tabel Data Uji

Pada Tabel dibawah ini menjelaskan pengujian data uji partisi untuk pengajuan cuti.

TABEL VII
TABEL PARTISI

No Uji	Field	Data Uji	Kode Partisi	Hasil
PC11	jenis cuti	tidak dipilih jenis cuti	T101	V
PC21	tanggal mulai	tidak memilih tanggal	T102	IV
PC31	tanggal akhir	tidak memilih tanggal	T201	V
PC41	pesan	tidak mengisi pesan	T202	IV

TABEL VIII
TABEL TIPE UJI

No Uji	Field	Data Uji	Kode Partisi	Hasil
PC11	jenis cuti	tidak dipilih jenis cuti	T101	V
PC21	tanggal mulai	tidak memilih tanggal	T102	IV
PC31	tanggal akhir	tidak memilih tanggal	T201	V
PC41	pesan	tidak mengisi pesan	T202	IV

3) Metode Partisi Ekuivalensi (*Equivalence Partitioning*)

Pada Tabel dibawah ini menjelaskan metode partisi ekuivalensi untuk pengajuan cuti

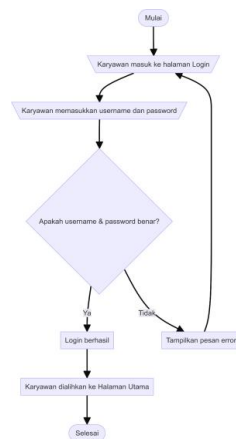
TABEL IX
TABEL PENGUJIAN PARTISI EKUIVALENSI

No.	Kondisi Pengujian	Test Case	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	mengosongkan semua input lalu submit	jenis cuti : - tanggal mulai : - tanggal akhir : - Pesan : -	tidak bisa diinput karena form tidak diisi dengan benar	tidak sesuai harapan	invalid
2	pengajuan cuti tanggal-nya mundur	jenis cuti : Cuti Tahunan tanggal mulai : 20-05-2024 tanggal akhir : 15-05-2024 Pesan : healing	tidak bisa diinput karena tanggalnya mundur	tidak sesuai harapan	invalid

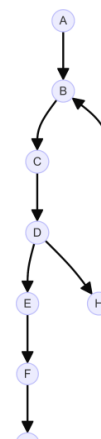
b. Pengujian *White Box*

a) *Login*

1) Diagram Alir



Gambar 6. *Flowchart Login*



Gambar 7. *Flowgraph Login*

2) Pemetaan *Source Code*

Berikut adalah pemetaan source code pada halaman login.

```
<?php
// Initialize the session session_start();

// Check if the user is already logged in, if yes then redirect him to welcome page
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
header("location: ../index.php"); exit;
}

// Include config file
// include '../function/model/Cuti.php';
// require_once "conn.php";
// require_once "koneksi.php";

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "ourdash";

$conn = new mysqli($servername, $username, $password, $dbname); if
($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// var_dump($conn);die();

// Define variables and initialize with empty values
$username = $password = "";
$username_err = $password_err = $login_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

// Check if username is empty if(empty(trim($_POST["username"]))) {
$username_err = "Please enter username.";
} else {
$username = trim($_POST["username"]);
}

// Check if password is empty if(empty(trim($_POST["password"]))) {
$password_err = "Please enter your password.";
} else {
$password = trim($_POST["password"]);
}

// Validate credentials
if(empty($username_err) && empty($password_err)){

// Prepare a select statement
$sql = "SELECT ID, FULLNAME, USERNAME, PASSWORD, ROLE FROM
users WHERE username = ?";
if($stmt = mysqli_prepare($conn, $sql)){

// Bind variables to the prepared statement as parameters
mysqli_stmt_bind_param($stmt, "s", $param_username);

// Set parameters
$param_username = $username;

// Attempt to execute the prepared statement if(mysqli_stmt_execute($stmt)){
```

```
// Store result mysqli_stmt_store_result($stmt);
// Check if username exists, if yes then verify password
if(mysqli_stmt_num_rows($stmt) == 1){
// Bind result variables
mysqli_stmt_bind_result($stmt, $id, $fullname, $username,
$hashed_password, $role);
if(mysqli_stmt_fetch($stmt)){ if(password_verify($password, $hashed_password)){

// Password is correct, so start a new session session_start();
// Store data in session variables
$_SESSION["loggedin"] = true;
$_SESSION["id"] = $id;
$_SESSION["username"] = $username;
$_SESSION["fullname"] = $fullname;
$_SESSION["role"] = $role;
// Redirect user to welcome page header("location: ../index.php");
} else{

// Password is not valid, display a generic error message
$login_err = "Invalid username or password.";
}
}

} else{
// Username doesn't exist, display a generic error message
$login_err = "Invalid username or password.";
}
} else{
echo "Oops! Something went wrong. Please try again later.";
}
// Close statement mysqli_stmt_close($stmt);
}
}
// Close connection mysqli_close($conn);
}
?>
<html lang="en">
<head>
<title>Ourdash</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<style> body{
background-color: lightgrey; font: 14px sans-serif;
}
.wrapper{ width: 360px; padding: 20px;
}
</style>
<body>
<div class="wrapper">
<h2>Login</h2>
<p>Please fill in your credentials to login.</p>
<?php if(!empty($login_err)){
echo '<div class="alert alert-danger">' . $login_err . '</div>';
}
?>
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post">
<div class="form-group">
<label>Username</label>
```

```

<input type="text" name="username" class="form-control <?php echo
(!empty($username_err)) ? 'is-invalid' : ";" value="<?php echo $username; ?>">
<span class="invalid-feedback"><?php echo $username_err; ?></span>
</div>
<div class="form-group">
<label>Password</label>
<input type="password" name="password" class="form-control <?php echo
(!empty($password_err)) ? 'is-invalid' : ";" value="<?php echo $password; ?>">
<span class="invalid-feedback"><?php echo $password_err; ?></span>
</div>
<div class="form-group">
<input type="submit" class="btn btn-primary" value="Login">
</div>

<p>Don't have an account? <a href="register.php">Sign up now</a>.</p>
</form>
</div>
</body>
</html>
    
```

- 3) Penghitungan Cyclomatic Complexity
 Menentukan kompleksitas cyclomatic dari grafik aliran.
 $V(G) = E - N + 2$
 $= 7 - 7 + 2$
 $= 2$

Keterangan:

E : Jumlah Busur atau Link N : Jumlah Simpul

- 4) *Independent Path*
 Menentukan dasar jalur independen
 Path 1 : 1-2-3-4-5-6
 Path 2 : 1-2-3-4-7-2

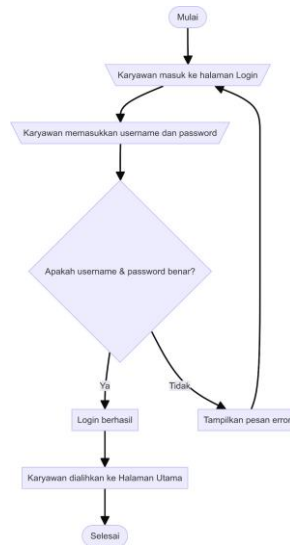
TABEL X
 TABEL INDEPENDENT PATH

Path	Jalur	Skenario
1	1-2-3-4-5-6	1. Mulai 2. Masuk ke halaman <i>Login</i> 3. Memasukan <i>username & password</i> 4. Memeriksa <i>username & password</i> 5. Masuk ke halaman utama 6. Selesai
2	1-2-3-4-7-2	1. Mulai 2. Masuk ke halaman <i>Login</i> 3. Memasukan <i>username & password</i> 4. Memeriksa <i>username & password</i> 5. Tampilkan pesan <i>error</i> 6. Masuk Ke halaman <i>Login</i>

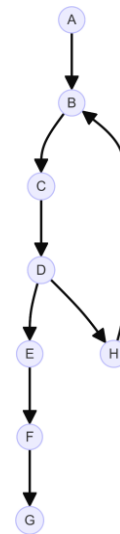
b) Register

1) Diagram Alir

Pada Gambar dibawah ini adalah diagram alir pada halaman *login*.



Gambar 6. Flowchart Login



Gambar 7. Flowgraph Login

2) Pemetaan *Source Code*

Berikut adalah pemetaan source code pada halaman login.

```

<?php
// Initialize the session session_start();

// Check if the user is already logged in, if yes then redirect him to welcome page
if(isset($_SESSION["loggedin"]) && $_SESSION["loggedin"] === true){
header("location: ../index.php"); exit;
}

// Include config file
// include '../function/model/Cuti.php';
// require_once "conn.php";
// require_once "koneksi.php";

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "ourdash";

$conn = new mysqli($servername, $username, $password, $dbname); if
($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}
// var_dump($conn);die();

// Define variables and initialize with empty values
$username = $password = "";
$username_err = $password_err = $login_err = "";

// Processing form data when form is submitted
if($_SERVER["REQUEST_METHOD"] == "POST"){

// Check if username is empty if(empty(trim($_POST["username"]))) {
$username_err = "Please enter username.";
} else {

```

```
$username = trim($_POST["username"]);
}

// Check if password is empty if(empty(trim($_POST["password"]))) {
$password_err = "Please enter your password.";
} else {
$password = trim($_POST["password"]);
}

// Validate credentials
if(empty($username_err) && empty($password_err)) {

// Prepare a select statement
$sql = "SELECT ID, FULLNAME, USERNAME, PASSWORD, ROLE FROM
users WHERE username = ?";
if($stmt = mysqli_prepare($conn, $sql)) {

// Bind variables to the prepared statement as parameters
mysqli_stmt_bind_param($stmt, "s", $param_username);

// Set parameters
$param_username = $username;

// Attempt to execute the prepared statement if(mysqli_stmt_execute($stmt)) {
// Store result mysqli_stmt_store_result($stmt);
// Check if username exists, if yes then verify password
if(mysqli_stmt_num_rows($stmt) == 1) {
// Bind result variables
mysqli_stmt_bind_result($stmt, $id, $fullname, $username,
$hashed_password, $role);
if(mysqli_stmt_fetch($stmt)) { if(password_verify($password, $hashed_password)) {

// Password is correct, so start a new session session_start();
// Store data in session variables
$_SESSION["loggedin"] = true;
$_SESSION["id"] = $id;
$_SESSION["username"] = $username;
$_SESSION["fullname"] = $fullname;
$_SESSION["role"] = $role;
// Redirect user to welcome page header("location: ../index.php");
} else {

// Password is not valid, display a generic error message
$login_err = "Invalid username or password.";
}
}

} else {
// Username doesn't exist, display a generic error message
$login_err = "Invalid username or password.";
}
} else {
echo "Oops! Something went wrong. Please try again later.";
}
// Close statement mysqli_stmt_close($stmt);
}
}
// Close connection mysqli_close($conn);
}
?>
```

```

<html lang="en">
<head>
<title>Ourdash</title>
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<style> body{
background-color: lightgrey; font: 14px sans-serif;
}
.wrapper{ width: 360px; padding: 20px;
}
</style>
<body>
<div class="wrapper">
<h2>Login</h2>
<p>Please fill in your credentials to login.</p>
<?php if(!empty($login_err)){
echo '<div class="alert alert-danger">' . $login_err . '</div>';
}
?>
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post">
<div class="form-group">
<label>Username</label>
<input type="text" name="username" class="form-control <?php echo
(!empty($username_err)) ? 'is-invalid' : ''; ?>" value="<?php echo $username; ?>">
<span class="invalid-feedback"><?php echo $username_err; ?></span>
</div>
<div class="form-group">
<label>Password</label>
<input type="password" name="password" class="form-control <?php echo
(!empty($password_err)) ? 'is-invalid' : ''; ?>">
<span class="invalid-feedback"><?php echo $password_err; ?></span>
</div>
<div class="form-group">
<input type="submit" class="btn btn-primary" value="Login">
</div>

<p>Don't have an account? <a href="register.php">Sign up now</a>.</p>
</form>
</div>
</body>
</html>

```

3) Penghitungan *Cyclomatic Complexity*

Menentukan kompleksitas cyclomatic dari grafik aliran.

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

Keterangan:

E : Jumlah Busur atau Link N : Jumlah Simpul

4) *Independent Path*

Menentukan dasar jalur independen

Path 1 : 1-2-3-4-5-6

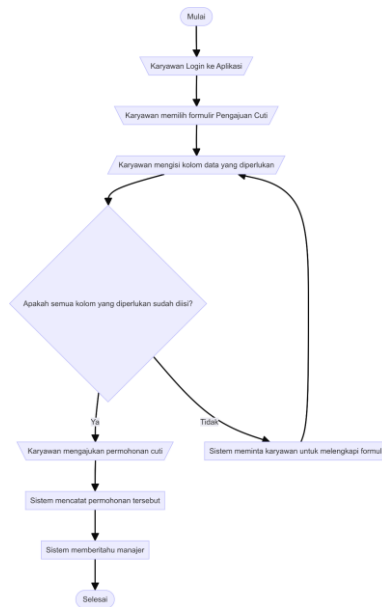
Path 2 : 1-2-3-4-7-2

TABEL X
 TABEL INDEPENDENT PATH

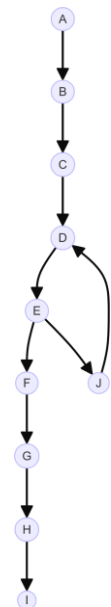
Path	Jalur	Skenario
1	1-2-3-4-5-6	1. Mulai 2. Masuk ke halaman <i>Login</i> 3. Memasukan <i>username & password</i> 4. Memeriksa <i>username & password</i> 5. Masuk ke halaman utama 6. Selesai
2	1-2-3-4-7-2	1. Mulai 2. Masuk ke halaman <i>Login</i> 3. Memasukan <i>username & password</i> 4. Memeriksa <i>username & password</i> 5. Tampilkan pesan <i>error</i> 6. Masuk Ke halaman <i>Login</i>

c) Pengajuan Cuti

1) Diagram Alir



Gambar 10. Flowchart Pengajuan Cuti



Gambar 11. Flowgraph Pengajuan Cuti

2) Pemetaan *Source Code*

```
<?php include '../layout/header.php' ?>
```

```
<div class="card">
<div class="p-4">
<div class="row">
<div class="col-md-6">
<div>
<h5>Pengajuan Cuti</h5>
</div>
<form id="addCuti">
<div class="mb-2">
<label for="">Jenis Cuti</label>
<select name="jenis_cuti" id="jenisCuti" class="form-control">
<option value="-" selected disabled>-</option>
<option value="Cuti Tahunan">Cuti Tahunan</option>
<option value="Cuti Besar">Cuti Besar</option>
<option value="Cuti Melahirkan">Cuti Melahirkan</option>
<option value="Cuti Sakit">Cuti Sakit</option>
<option value="Cuti Pernikahan">Cuti Pernikahan</option>
<option value="Cuti Menikah">Cuti Menikah</option>
```



```

    }
    html += `
    <tr>
    <td>${v.CREATED_DATE}</td>
    <td>${v.TANGGAL_MULAI} - ${v.TANGGAL_AKHIR}</td>
    <td>
    <button type="button" class="btn ${bgColor} text-white">${v.STATUS}</button>
    </td>
    <td>${v.KETERANGAN}</td>
    </tr>
    `;
    });
    $('#dataCuti').html(html);
    },
    error: function (err) { console.error(err);
    }
    })
    }
    $('#saveCuti').on('click', function () {
    // alert('aa');
    var data = $('#addCuti').serialize();
    // console.log(data); Swal.fire({
    title: "Do you want to save the changes?", showDenyButton: true,
    showCancelButton: true, confirmButtonText: "Save", denyButtonText: `Don't save`
    }).then((result) => {
    /* Read more about isConfirmed, isDenied below */ if (result.isConfirmed) {
    $.ajax({
    type: 'post',
    url: './function/controller/cutiController.php?func=addcuti', data: data,
    success: function (response) { Swal.fire("Saved!", "", "success");
    },
    error: function (err) { console.error();
    },
    complete: function () {
    setInterval(function() { location.reload();
    }, 3000);
    },
    })
    } else if (result.isDenied) {
    Swal.fire("Changes are not saved", "", "info");
    }
    });
    });
    </script>
    <?php include './layout/footer.php' ?>

```

3) Penghitungan *Cyclomatic Complexity*
 Menentukan kompleksitas cyclomatic dari grafik aliran.

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= 10 - 10 + 2 \\
 &= 2
 \end{aligned}$$

Keterangan:
 E : Jumlah Busur atau Link
 N : Jumlah Simpul

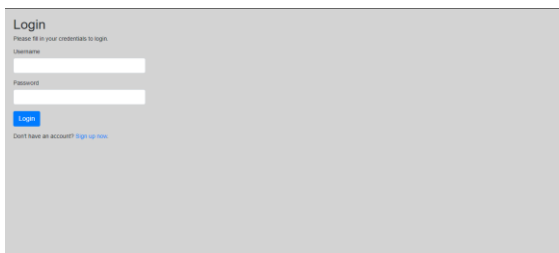
4) *Independent Path*
 Menentukan dasar jalur independen
 Path 1: 1-2-3-4-5-6-7-8-9
 Path 2 : 1-2-3-4-5-2

TABEL XII
 TABEL INDEPENDENT PATH

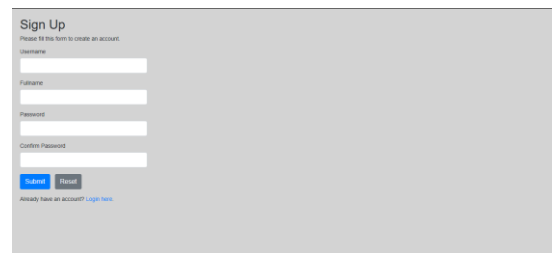
Path	Jalur	Skenario
1	1-2-3-4-5-6-7-8-9	1. Mulai 2. <i>Login</i> ke Aplikasi 3. Masuk ke halaman Pengajuan Cuti 4. Mengisi formulir Pengajuan Cuti 5. Memeriksa isi data 6. Mengirim pengajuan cuti 7. Menyimpan pengajuan ke <i>database</i> 8. Memberikan notifikasi ke manajer 9. Selesai
2	1-2-3-4-5-10-4	1. Mulai 2. <i>Login</i> ke Aplikasi 3. Masuk ke halaman Pengajuan Cuti 4. Mengisi formulir Pengajuan Cuti 5. Memeriksa isi data 6. Mengisi formulir Pengajuan Cuti

4. Hasil Perancangan

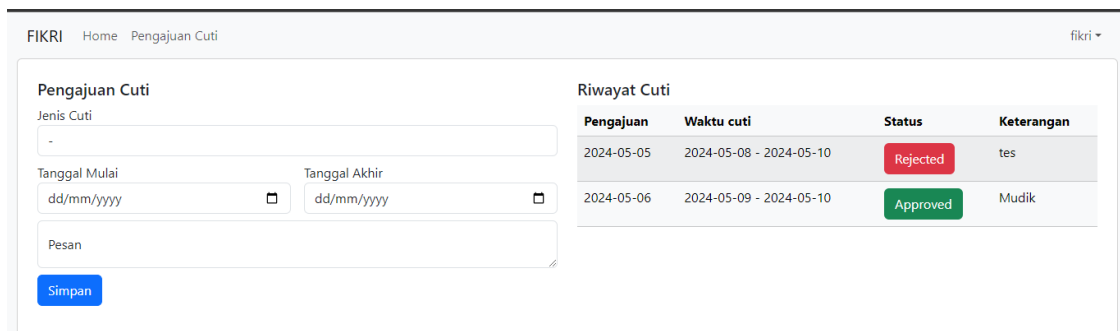
Pada gambar berikut adalah hasil perancangan untuk halaman *login*, halaman register dan halaman pengajuan cuti.



Gambar 12. Halaman *Login*



Gambar 13. Halaman Register



Gambar 14. Halaman Pengajuan Cuti

IV. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian dari penelitian yang berjudul “Pengujian Fungsional dan Struktural Aplikasi Pengajuan Cuti Dengan Metode Blackbox dan Whitebox” maka dapat ditarik kesimpulan bahwa aplikasi berfungsi dengan baik sesuai spesifikasi. Pengujian *Black Box* memastikan semua fungsionalitas memenuhi harapan pengguna, sementara pengujian *White Box*, termasuk kompleksitas siklotatik, menjamin keandalan dan kualitas kode. Hasilnya menunjukkan bahwa aplikasi efektif dalam pengelolaan cuti dan memberikan pengalaman pengguna yang memuaskan, berkat metode pengembangan *Rapid Application Development (RAD)* yang mempercepat proses pengembangan.

REFERENSI

- [1] S. Utarki, E. A. Pratama, and C. M. Hellyana, "Sistem Informasi Pariwisata Berbasis *Website* Pada Taman Nasional Gunung Ciremai Jawa Barat," *IJSE - Indonesian Journal on Software Engineering*, vol. 6, no. 1, pp. 19–32, 2020.
- [2] H. Hasnita et al., "Implementasi *Website* Sistem Informasi Pemasaran Pada Rumah Makan Kampong Bakau," *SIMKOM*, vol. 7, no. 2, pp. 63–73, 2022.
- [3] A. K. Saputro et al., *Pemrograman Dasar Web dengan HTML*. Penerbit KBM INDONESIA, 2024.
- [4] I. P. Sari et al., "Perancangan Sistem Informasi Penginputan *Database* Mahasiswa Berbasis Web," *Hello World Jurnal Ilmu Komputer*, vol. 1, no. 2, pp. 106–110, 2022.
- [5] D. R. Anamisa and F. A. Mufarroha, *Dasar Pemrograman WEB Teori dan Implementasi: HTML, CSS, Javascript, Bootstrap, Codelgniter*. Media Nusa Creative, 2022.
- [6] E. Nur Annisa et al., *SISTEM INFORMASI PARIWISATA BERBASIS WEB PADA KABUPATEN NUNUKAN*, 2022.
- [7] D. Krisbiantoro and P. D. Abda`u, *Dasar Pemrograman Web dengan Bahasa HTML, PHP, dan Database MySQL*. Zahira Media Publisher, 2021.
- [8] A. M. Dawis et al., *Rekayasa Perangkat Lunak: Panduan Praktis untuk Pengembangan Aplikasi Berkualitas*. Penerbit Widina, 2023.
- [9] H. Kurniawan et al., *BELAJAR WEB PROGRAMMING: Referensi Pengenalan Dasar Tahapan Belajar Pemrograman Web Untuk Pemula*. PT. Sonpedia Publishing Indonesia, 2023.
- [10] L. V. Aprilian and M. H. K. Saputra, *Belajar Cepat Metode SAW. Kreatif*, 2020.
- [11] M. M. Amin, A. Sutrisman, and Y. Dwitayanti, *Bahasa Query Menggunakan MySQL*. Penerbit Ahatek, 2022.
- [12] Santoso, L., & Amanullah, J., "Pengembangan Sistem Informasi Akademik Berbasis Website Menggunakan Metode Rapid Application Development (Rad)". *Elkom: Jurnal Elektronika dan Komputer*, 15(2), 250-259, 2022.
- [13] Gusdevi, H., Hutama, B. D., & Rizaldi, A. A., "Pengujian White Box pada Aplikasi Pinjam Dulu Seratus Berbasis Website". *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, 6(2), 97-102, 2024.