OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

Otomatisasi Pengujian Aplikasi POS (*Point Of Sale*) Menggunakan Metode *White Box*

Amin Andriyanto[1*], Apriansyah[1], Auransyah Sekar Pratiwi[1], Iqbal Nurdiansyah[1], Aries Saifudin[1]

[1]Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia
Email: [1*]aminndrynt@gmail.com, [2]aprinsyh23@gmail.com, [3]aurasekar2@gmail.com, [4]iqbalnurdian18@gmail.com, [5]aries.saifudin@unpam.ac.id
(* : coressponding author)

Abstrak− *Point of sale* (POS), atau yang sering disebut POS, adalah sistem dalam dunia bisnis yang menggunakan perangkat keras dan perangkat lunak. Sistem POS ini memiliki fitur-fitur utama seperti menambahkan transaksi, melihat riwayat transaksi penjualan dan grafik kemajuan penjualan, memperkirakan pendapatan, serta mengidentifikasi penjualan yang mencakup daftar produk dan kombinasi produk terlaris. Pengujian aplikasi ini dilakukan dengan metode pengujian kotak putih selama dua minggu setelah program selesai. Hasil dari penerapan aplikasi POS adalah aplikasi web untuk pesanan penjualan yang memudahkan pengelolaan barang dan rekap transaksi bulanan atau tahunan. Dalam evaluasi kecocokan aplikasi dan kesesuaian dengan basis data, aplikasi ini diimplementasikan selama satu minggu untuk memeriksa apakah masih ada masalah atau tidak. Selama program disetujui, akan dilakukan penyempurnaan dan penyesuaian sesuai dengan kebutuhan di toko yang akan menggunakan aplikasi POS ini. Berdasarkan penelitian yang dilakukan, dapat disimpulkan bahwa sistem informasi Point of Sale (POS) yang dibuat dapat digunakan oleh pengguna.

Kata Kunci: *White Box, POS (Point Of Sale)*

Abstract− *Point of sale (POS), or what is often called POS, is a system in the business world that uses hardware and software. This POS system has key features such as adding transactions, viewing sales transaction history and sales progress graphs, estimating revenue, and identifying sales which includes a list of best selling products and product combinations. Testing of this application was carried out using the white box method for two weeks after the program was finished. The result of implementing the POS application is a web application for sales orders that makes it easy to manage goods and recap monthly or annual transactions. In evaluating application compatibility and compatibility with the database, this application is implemented for one week to check whether there are still problems or not. As long as the program is approved, improvements and adjustments will be made according to the needs of stores that will use this POS application. Based on the research conducted, it can be concluded that the Point of Sale (POS) information system created can be used by users.*

Keywords*: White Box, POS (Point Of Sale)*

1. PENDAHULUAN

Pengujian terhadap sistem, merupakan elemen yang digunakan untuk merepresentasikan hasil analisis, perancangan dan implementasi guna menjamin kualitas dari sistem yang dikembangkan. Pengujian sistem dilakukan untuk menemukan kesalahan yang terjadi dalam sistem sebagai dasar dalam menemukan solusi, dan meningkatkan kualitas dari sistem (Pratama et al., 2020). Pengujian sistem dilakukan guna mendapatkan sebuah penilaian, agar nantinya tidak perlu dilakukan pengujian berulang ketika tujuan pengujian telah tercapai. Sistem dikatakan gagal apabila saat proses pengoperasian data tidak sesuai dengan harapan. Kelemahan sistem dan hasil yang dianggap valid akan diketahui setelah proses pengujian dilakukan. Dalam pengujian akan diketahui secara pasti error dan fungsi dari masing-masing proses di dalam sistem apakah telah berjalan dengan baik seluruhnya (Wulandari et al., 2022).

Untuk mengurangi kesalahan yang terjadi, maka perlu dilakukan sebuah pengujian agar menghindari kerugian (Ningrum et al., 2019). Untuk meminimalisir kesalahan berupa error dari sistem tentu diperlukan tindakan preventif berupa pengujian untuk mendeteksi sedini mungkin kesalahan dan

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

kekurangan, sehingga sesegera mungkin sistem diperbaiki sebelum digunakan oleh user (Dewi et al., 2022).

Proses pengujian perangkat lunak melibatkan eksekusi program atau aplikasi guna menemukan bug, serta memvalidasi dan memverifikasi kepatuhan terhadap persyaratan bisnis, teknis, desain, pengembangan, dan kemampuan implementasi yang diharapkan.

Definisi pengujian perangkat lunak melibatkan beberapa aspek. Pertama, pengujian adalah suatu proses yang terjadi dalam aktivitas tunggal. Kedua, pengujian dilakukan sepanjang Siklus Hidup Pengembangan Perangkat Lunak (SDLC). Ada dua jenis pengujian yang dilakukan, yaitu pengujian statis dan pengujian dinamis. Pengujian statis melibatkan pemeriksaan dokumen dan analisis statis untuk menemukan cacat tanpa menjalankan kode, sementara pengujian dinamis melibatkan menjalankan kode perangkat lunak untuk menguji fungsionalitasnya. Selain itu, pengujian juga melibatkan perencanaan, persiapan, dan evaluasi. Dalam perencanaan, kita merencanakan kegiatan pengujian dan melaporkan kemajuan dan status perangkat lunak yang diuji. Persiapan melibatkan pemilihan jenis pengujian, kondisi pengujian, dan perancangan kasus uji. Sedangkan dalam evaluasi, kita memeriksa hasil pengujian dan mengevaluasi apakah perangkat lunak telah memenuhi kriteria penyelesaian. Akhirnya, dalam pengujian perangkat lunak, tidak hanya kode perangkat lunak yang diuji, tetapi juga persyaratan pengujian, desain spesifikasi, serta dokumen-dokumen terkait lainnya seperti operasi, pengguna, dan materi pelatihan memiliki tingkat penting yang sama.

Pengujian perangkat lunak memiliki berbagai tujuan dan objectives. Tujuan utamanya adalah untuk menemukan kesalahan atau kekurangan yang mungkin terjadi saat pengembangan perangkat lunak oleh para programmer. Selain itu, pengujian juga bertujuan untuk memperoleh kepercayaan dan memberikan informasi tentang tingkat kualitas perangkat lunak tersebut. Pengujian juga dilakukan untuk mencegah terjadinya cacat, memastikan bahwa hasil akhir memenuhi kebutuhan bisnis dan kebutuhan pengguna, serta memastikan bahwa perangkat lunak sesuai dengan Spesifikasi Kebutuhan Bisnis (BRS) dan Spesifikasi Kebutuhan Sistem (SRS). Selain itu, pengujian juga dilakukan untuk mendapatkan kepercayaan dari pelanggan dengan menyediakan produk yang berkualitas.

Perangkat lunak pengujian membantu dalam menyelesaikan aplikasi perangkat lunak atau produk untuk kebutuhan bisnis dan pengguna. Keberadaannya sangat penting untuk memastikan bahwa aplikasi perangkat lunak tersebut diuji secara menyeluruh dan berfungsi dengan baik sesuai spesifikasi. Saat menentukan cakupan tes, penting untuk merancangnya dengan baik agar memiliki kemungkinan maksimum dalam menemukan kesalahan atau bug. Uji kasus juga harus efektif, dengan tujuan yang dapat diukur melalui jumlah cacat yang dilaporkan dalam setiap kasus uji.

Aplikasi yang akan di uji adalah sebuah aplikasi *Point Of Sales (POS)* hasil buatan siswa Rekayasa Perangkat Lunak SMK Citra Negara atas nama Gibran Fajar Satritama, aplikasi POS ini diperlukan sebagai persyaratan kelulusan untuk jurusan Rekayasa Perangkat Lunak (RPL). Aplikasi POS ini memiliki fitur diantaranya : Menambahkan barang, Menambahkan kategori barang, Menambahkan user yang akan menggunakan aplikasi POS, Mengedit data batang, Mengedit data category barang, menghapus data barang, Menghapus data category barang, serta Melakukan transaksi dengan menggunakan API Mitrans.

Aplikasi POS ini memiliki manfaat yang signifikan. Pertama, aplikasi ini menjadi salah satu syarat kelulusan Jurusan Rekayasa Perangkat Lunak. Selain itu, aplikasi ini dapat membantu usaha-usaha kecil dalam melakukan transaksi dengan mudah. Aplikasi ini juga mempermudah pembuatan laporan stok data barang dan laporan penjualan barang. Selain itu, pengguna aplikasi ini dapat melakukan transaksi dengan mudah baik menggunakan uang tunai maupun non-tunai.

Ketidakadanya pengujian aplikasi sebelum digunakan oleh banyak orang dapat menimbulkan masalah dan kerugian. Pertama, masalah keamanan dapat muncul karena aplikasi yang tidak diuji memiliki kerentanan terhadap serangan dari berbagai sisi, terutama aplikasi dengan fitur transaksi yang dapat menyebabkan pencurian data atau penyalahgunaan oleh pihak yang tidak bertanggung jawab. Kedua, masalah pengembangan terjadi ketika aplikasi tidak melalui pengujian, sehingga tidak dapat berkembang sesuai keinginan atau kebutuhan pengguna, yang mengakibatkan pembuatan banyak aplikasi baru untuk memenuhi semua kebutuhan pengguna. Ketiga, masalah ketidaksesuaian dapat

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

terjadi saat aplikasi yang telah dibuat tidak sesuai dengan rancangan awal, yang akan menyebabkan pemborosan waktu karena perlu dilakukan perubahan dan pengujian ulang terhadap aplikasi yang sudah diperbaiki.

Dalam pengujian perangkat lunak *(Software Testing)* ini kami menggunakan metode Kotak Putih *(White Box)*, alasan kami menggunakan metode ini karna kami hanya berfokus pada sisi UI/UX dan alur program berjalan sehingga kedepannya dapat dikembangkan atau disesuaikan dengan keinginan atau kebutuhan user.

Tahap dalam melakukan pengujian Kotak Putih (White Box) meliputi pembuatan flowchart source code yang akan diuji, pembuatan listing program pengujian yang fokus pada program logika seperti If...else..., switch...case, dan lain sebagainya, serta pembuatan metrik trafik untuk menghitung Cyclomatic Complexity.

2. METODOLOGI PENELITIAN

Untuk memastikan bahwa setiap langkah dalam sistem aplikasi berjalan sesuai yang diinginkan, perlu dilakukan pengujian sistem. Pengujian ialah salah satu proses membangun fitur lunak buat menciptakan bug serta memperbaikinya sehingga fitur lunak bisa dikatakan berperan (Nurudin et al., 2019). Jika metode pengujian yang digunakan tidak optimal, akan memiliki dampak negatif pada kualitas perangkat lunak yang dihasilkan. Pengujian perangkat lunak yang tidak efektif dan tidak lengkap dapat menimbulkan berbagai masalah ketika perangkat lunak digunakan oleh pengguna akhir (Hanifa et al., 2016).

2.1 *White Box* (Kotak Putih)

Metode pengujian aplikasi yang dikenal sebagai White Box melibatkan penggunaan penjelasan struktur kontrol sebagai bagian dari desain level komponen untuk menghasilkan kasus-kasus uji. White Box memiliki beberapa teknik yang digunakan dalam proses pengujian, seperti Pengujian Aliran Data, Pengujian Aliran Kontrol, Pengujian Basic Path / Path, dan Pengujian Loop. Dalam pengujian White Box, para penguji harus memiliki pemahaman mendalam tentang kode sumber yang akan diuji. Metode pengujian White Box ini bertujuan untuk mengungkap kesalahan implementasi yang mungkin ada dalam sebuah aplikasi. Pengujian White Box dapat diterapkan pada berbagai tingkatan, termasuk integrasi, unit, dan sistem. Terdapat beberapa keunggulan dan kelemahan dalam pengujian menggunakan metode White Box, antara lain:

a. Keunggulan:
    1. Metode White Box dapat mengungkapkan kesalahan dalam kode dengan mengidentifikasi dan menghapus baris kode yang tidak diperlukan.
    2. Pengujian dengan metode White Box memiliki cakupan yang maksimal dalam menguji aplikasi saat skenario uji coba.
b. Kekurangan:
    1. Biaya pengujian menggunakan metode White Box cenderung tinggi karena memerlukan penguji yang berpengalaman di bidang ini.
    2. Beberapa alur program mungkin tidak diuji karena tidak praktis untuk menguji setiap baris kode guna menemukan kesalahan.

2.2 Teknik Basis Path

Tom McCabe pertama kali memperkenalkan Teknik Basis Path sebagai salah satu metode Pengujian White Box. Teknik ini memungkinkan penguji untuk mengukur tingkat kompleksitas logika dalam desain prosedural. Dalam pengujian menggunakan Teknik Basis Path, skenario uji coba yang dibuat dijamin akan menjalankan setiap pernyataan dalam aplikasi yang diuji minimal satu kali selama tahap pengujian.

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

2.3 *Cyclomatic Complexity*

Cyclomatic Complexity merupakan sebuah metrik yang digunakan untuk mengukur kompleksitas logika dalam sebuah program. Metode ini memberikan nilai kuantitatif terhadap tingkat kompleksitas perangkat lunak. Rumus yang digunakan untuk menghitung Cyclomatic Complexity adalah sebagai berikut: V(G) = E - N + 2

Di mana:

E = jumlah sisi (edges) pada flowgraph
N = jumlah simpul (nodes) pada flowgraph
P = jumlah simpul predikat (predicate nodes) pada flowgraph

2.4 *Flowchart*

Flowchart merupakan representasi visual dari langkah-langkah dan urutan prosedur dalam suatu aplikasi. Fungsinya adalah membantu analis dan programmer dalam memahami dan memecah aplikasi menjadi segmen-segmen yang lebih kecil agar lebih mudah dianalisis. Dengan menggunakan flowchart, penyelesaian masalah dalam aplikasi dapat lebih mudah dilakukan. Flowchart sering digunakan dalam perancangan aplikasi untuk menggambarkan logika yang digunakan dalam aplikasi tersebut.

2.4 *Flowgraph*

Flowgraph merupakan representasi grafis dari program yang dihasilkan dari pemetaan flowchart program. Tujuan dari flowgraph adalah untuk merepresentasikan aliran kontrol logika program yang ada. Flowgraph umumnya digunakan dalam tahap pengujian yang berfokus pada visualisasi aliran dari suatu program.

Penelitian ini melibatkan pengujian langsung terhadap aplikasi yang telah dirancang dengan menggunakan pendekatan kuantitatif. Oleh karena itu, hasil penelitian ini telah dilakukan hingga akhir untuk mengumpulkan fakta yang dapat menjadi bukti penelitian. Penelitian ini terdiri dari tiga tahapan yang dibagi secara berurutan:

2.4.1 Pembuatan *Form User*

Tahap awal dalam penelitian ini melibatkan pembuatan aplikasi pengguna di sisi admin menggunakan bahasa pemrograman PHP. Aplikasi ini menggunakan library Bootstrap dan Javascript untuk menciptakan tampilan antarmuka yang memungkinkan program berjalan dengan menggunakan antarmuka grafis (GUI). Aplikasi pengguna ini memiliki empat fungsi yang diperlukan agar aplikasi dapat beroperasi. Beberapa fungsi yang membutuhkan tampilan GUI akan memanggil layar agar dapat ditampilkan pada monitor (misalnya: menu utama dan pesan kesalahan pop-up). Berikut adalah daftar fungsi yang terdapat dalam aplikasi Form tambah pengguna:

a. Fungsi session
b. Fungsi create tambah user
c. Fungsi read tambah user
d. Fungsi update user
e. Fungsi delete user

2.4.2 Pengujian *Form User*

Proses pengujian aplikasi ini dimulai dengan menghitung jumlah skenario uji yang akan dilakukan, dengan menggunakan Cyclomatic Complexity menggunakan rumus V(G) = E - N + 2 berdasarkan flowgraph yang telah dibuat. Setelah mendapatkan nilai Cyclomatic Complexity, langkah selanjutnya adalah membuat skenario uji. Pengujian dilakukan secara manual, di mana setiap skenario uji dijalankan secara terpisah dan hasil keluaran dari setiap jalur independen diperhatikan dan dievaluasi.

2.4.3 Analisis *Form User*

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

Tahap selanjutnya adalah analisis, di mana dibuat sebuah tabel test case dan hasil keluaran setelah percobaan dibandingkan dengan tabel test case tersebut. Tabel test case ini ditampilkan dalam Tabel 1.

Tabel 1. Format *Test Case*

No	Skenario Uji	Hasil Yang Diharapkan	Hasil	Keterangan
1	-	-	-	-
2	-	-	-	-

Setelah semua tahapan selesai, kesimpulan dari penelitian ini dapat ditarik dalam pengujian aplikasi form login menggunakan metode White Box.

3. HASIL DAN PEMBAHASAN

3.1 *WHITE BOX* (Kotak Putih)

3.1.1 *Source Code Form User*

Pada pembuatan aplikasi form pengguna ini, aplikasi melakukan validasi dengan memeriksa file-file yang terdapat dalam folder yang sama dengan tempat aplikasi PHP berjalan. Data yang akan di tambah akan diperiksa apakah sesuai dengan field pada database atau tidak, lalu untuk updae data akan diperiksa dan ditarik data dari database yang disesuaikan apa yang ingin diupdate. Dalam pengujian aplikasi ini telah dibuat sebuah file dengan nama “act_create” dan “act_update”. Secara keseluruhan, kode sumber fungsi pengguna ditampilkan dalam Tabel 2.

Tabel 2. *Function act_create dan act_update*

Souce Code

```
$username = $_POST['username'];
$password = password_hash($_POST['password'], PASSWORD_DEFAULT);
$nama_lengkap = $_POST['nama_lengkap'];
$level = $_POST['level'];

$query = "INSERT INTO tbl_users
VALUES('','$username','$password','$nama_lengkap','$level')";
$hasil = mysqli_query($conn, $query);

if ($hasil) {
    echo "<script>
            Swal.fire({
                title: 'Simpan data berhasil!',
                icon: 'success',
                confirmButtonColor: '#3085d6',
                confirmButtonText: 'Ok'
            }).then(result => {
                if (result.isConfirmed) {
                    window.location='view.php'
                }
```

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

```
            });
        </script>";
} else {
    echo "Error updating record: " . mysqli_error($koneksi);
}
```

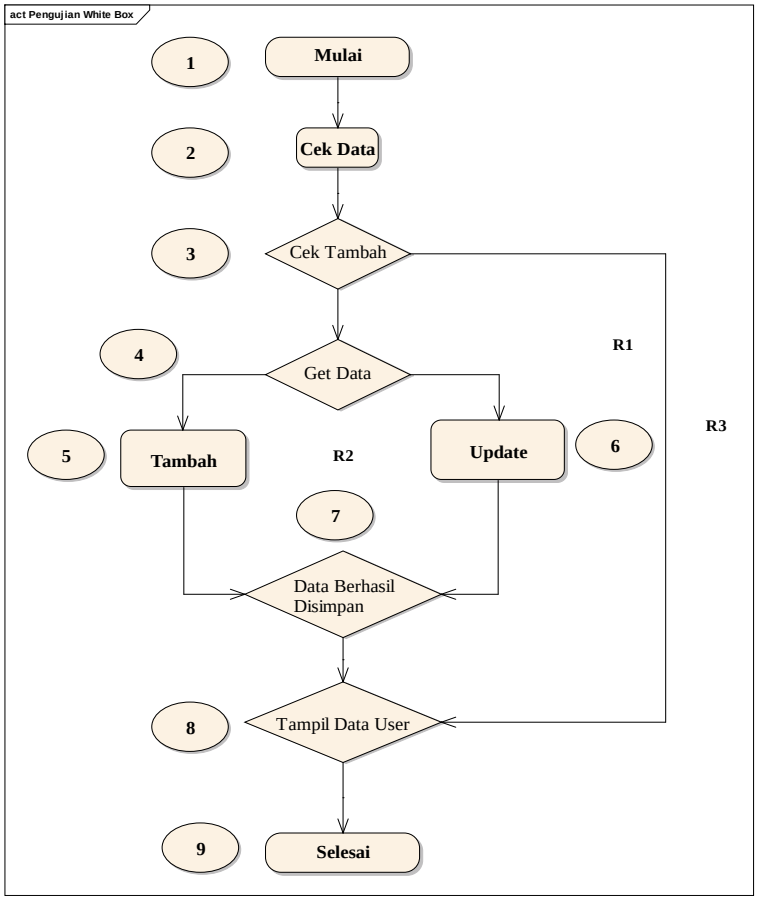
```
$id = $_POST['id'];
$username = $_POST['username'];
$password = password_hash($_POST['password'], PASSWORD_DEFAULT);
$nama_lengkap = $_POST['nama_lengkap'];
$level = $_POST['level'];

if ($password == "") {
    $query = "UPDATE tbl_users SET username='$username',
nama_lengkap='$nama_lengkap', level='$level' WHERE id='$id' ";
    $hasil = mysqli_query($conn, $query);

    if ($hasil) {
        echo "<script>
        Swal.fire({
            title: 'Ubah data berhasil!',
            icon: 'success',
            confirmButtonColor: '#3085d6',
            confirmButtonText: 'Ok'
        }).then(result => {
            if (result.isConfirmed) {
                window.location='view.php'
            }
        });
        </script>";
    } else {
        echo "Error updating record: " . mysqli_error($koneksi);
    }
} else {
    $query = "UPDATE tbl_users SET username='$username', password='$password',
nama_lengkap='$nama_lengkap', level='$level' WHERE id='$id' ";
    $hasil = mysqli_query($conn, $query);

    if ($hasil) {
        echo "<script>
        Swal.fire({
                title: 'Ubah data berhasil!',
```

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

```
                icon: 'success',
                confirmButtonColor: '#3085d6',
                confirmButtonText: 'Ok'
            }).then(result => {
                if (result.isConfirmed) {
                    window.location='view.php'
                }
            });
            </script>";
    } else {
        echo "Error updating record: " . mysqli_error($koneksi);
    }
}
```

3.2 *Flowchart* Aplikasi *Form Users*

Flowchart ini dibuat dengan tujuan untuk dapat dikonversi menjadi bentuk flowgraph. Secara keseluruhan, Flowchart Form User yang telah dibuat dapat dilihat pada Gambar 1.

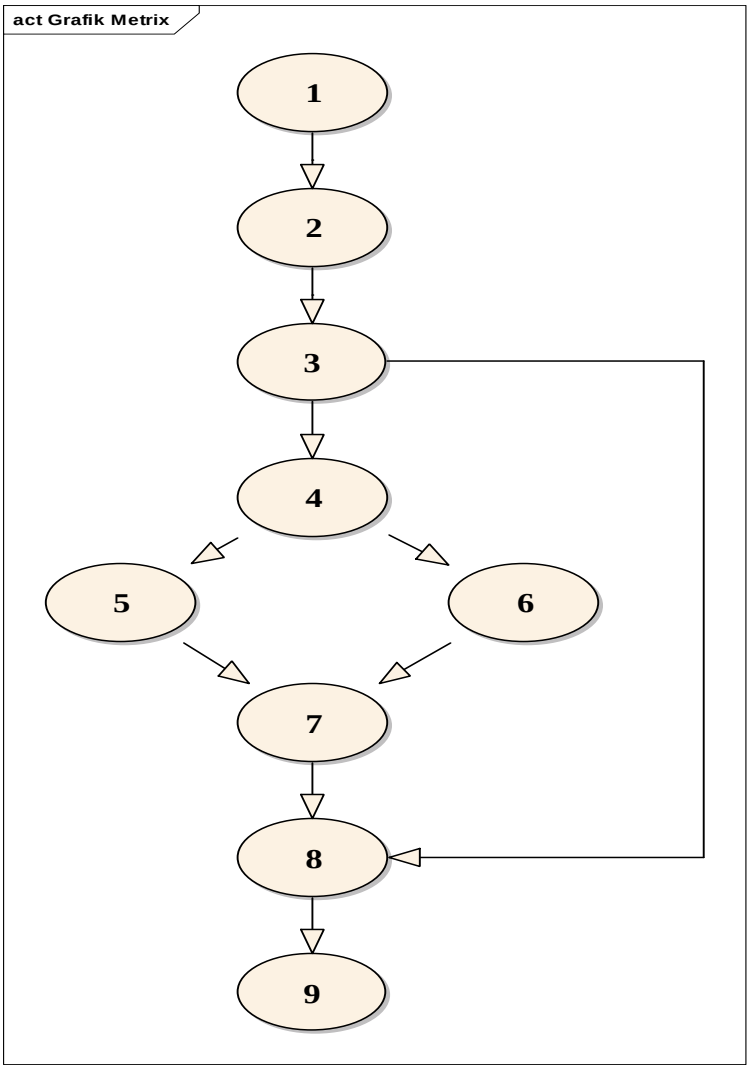


Gambar 1. *Flowchart Form User*

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

3.3 *Flowgraph* Aplikasi *Form Users*

Setelah pembuatan flowchart, langkah selanjutnya adalah membuat flowgraph. Flowgraph ini dibuat dengan tujuan untuk menghitung Cyclomatic Complexity, yang melibatkan perhitungan jumlah node dan jumlah edge dalam flowgraph tersebut.



Gambar 2. *Flowgraoh Form User*

3.3.1 Menghitung Jalur Independen Menggunakan *Cyclomatic Complexity*

Setelah pembuatan flowchart, langkah selanjutnya adalah membuat flowgraph. Flowgraph ini dibuat dengan tujuan untuk menghitung Cyclomatic Complexity, yang melibatkan perhitungan jumlah node dan jumlah edge dalam flowgraph tersebut.

Untuk menghitung CC digunakan formula: V (G) = E – N + 2 Dari hasil pembuatan flowgraph sebelumnya, dapat diketahui:

Region (R) = 3
Node (N) = 9
Predicate node (P) = 2

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

Setelah nilai tersebut dimasukan ke dalam formula akan menghasilkan:

a. Perhitungan dengan rumus:
   V(G) = E – N – 2
   V(G) = 10 – 9 + 2
   V(G) = 3
b. Perhitungan dengan rumus:
   V(G) = P + 1
   V(G) = 2 + 1
   V(G) = 3

Berdasarkan hasil perhitungan yang telah dilakukan, dapat diketahui bahwa terdapat 3 jalur independen atau jalur dasar dalam fitur "Tampil Detail Data Guru" pada sisi admin. Berikut adalah rincian dari 3 jalur independen tersebut:

- Path I : 1 – 2 – 3 – 4 – 5 – 7 – 8 – 9 (Skenario tambah data user)
- Path II : 1 – 2 – 3 – 4 – 6 – 7 – 8 – 9 (Skenario update data user)
- Path III : 1 – 2 – 3 – 8 – 9 (Skenario tidak ada nya data yang ditambah atau diupdate)

Setelah menyelesaikan penentuan jalur independen, langkah selanjutnya adalah membuat tabel test case dan membandingkan hasil output aktual dengan hasil yang diharapkan. Tabel test case yang telah dibuat dapat dilihat pada Tabel 3.

Tabel 3. *Table Test Case*

Jalur/ Path	Kegiatan	Hasil yang diharapkan	Hasil	Keterangan
1	Input data user yang akan di tambah	Data yang telah diinput berhasil disampan dan menampilkan data user	Data tersimpan dan menampilkan data user yang telah ditambahkan	Valid
2	Mengambil data user yang akan di update data	Data yang diambil sesuai dengan data sebelumnya dan melakukan update data dan disimpan setelah itu menampilkan halaman data user yang di update	Data tersimpan dan menampilkan data user yang telah di update	Valid
3	Tidak melakukan penambahan data atau perubahan data user	Hanya menampilkan data user yang sudah ada dan sesuai dengan data dalam database	Menampilkan data user yang sesuai dengan data dalam database	Valid

1. Tes Case Jalur 1
   Jalur ini melibatkan pengecekan data pengguna, kemudian pengguna dapat menambahkan data pengguna, melakukan pengecekan untuk mendapatkan data pengguna, memastikan data berhasil disimpan, menampilkan data pengguna, dan kembali ke halaman utama aplikasi.
2. Test Case Jalur 2
   Jalur ini melibatkan pengecekan data pengguna, kemudian pengguna dapat memperbarui data pengguna, melakukan pengecekan untuk mendapatkan data pengguna, memastikan data berhasil diperbarui, menampilkan data guru, dan kembali ke halaman utama aplikasi.
3. Test Case 3
   Jalur ini melibatkan pengecekan data pengguna, kemudian langsung menampilkan data pengguna, dan kembali ke halaman utama aplikasi.

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

4. KESIMPULAN

4.1 Kesimpulan

Otomatisasi pengujian aplikasi POS sistem menggunakan metode white box memberikan solusi yang efektif dalam meningkatkan efisiensi dan akurasi pengujian. Metode white box memungkinkan pengujian mendalam dengan memeriksa elemen-elemen internal aplikasi, Dengan menggunakan metode ini, pengujian aplikasi POS sistem dapat mencakup pengujian unit yang rinci dan pengujian fungsional yang menyeluruh, dengan hasil yang efisien dan akurat.

Namun, penting untuk diingat bahwa otomatisasi pengujian tidak dapat menggantikan sepenuhnya pengujian manual dan keahlian pengujian manusia. Pengujian manual tetap diperlukan untuk mengidentifikasi masalah yang kompleks, menguji pengalaman pengguna yang lebih intuitif, dan melakukan validasi yang mendalam. Oleh karena itu, otomatisasi pengujian dengan metode white box harus diimbangi dengan pengujian manual yang disesuaikan agar memastikan kualitas keseluruhan aplikasi POS sistem. Dalam keseluruhan, otomatisasi pengujian dengan menggunakan metode dan alat yang tepat dapat memberikan manfaat signifikan dalam meningkatkan efisiensi dan akurasi pengujian aplikasi POS sistem, dengan penekanan yang tepat pada peran penting pengujian manual.

4.2 Saran

1. Melakukan pengujian menggunakan Metode Black Box, untuk membandingkan efektivitas dan kelebihan masing-masing metode white box dan black box dalam konteks otomatisasi pengujian aplikasi POS sistem.
2. Integrasi dengan alat pengujian lainnya, Selenium IDE adalah alat otomatisasi pengujian yang populer, tetapi ada berbagai alat pengujian lainnya yang tersedia seperti alat manajemen uji coba, kerangka kerja pengujian, atau alat pelaporan.

REFERENCES

Dewi, E. H., Pratama, I. S., Putera, A. S., & Carudin. (2022). Black Box Testing pada Aplikasi Pencatatan Peminjaman Buku Menggunakan Boundary Value Analisis. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 315-324.

Hanifah, U., Alit, R., dan Sugiarto. (2016). Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk. *SCAN, VOL. XI* NOMOR 2.

Mulyati, S., Kusyadi, I., Ashara, M. I., Widodo, A. P., & Wahyudin. (2022). Pengujian Black Box ada Aplikasi Hitung Nilai Mahasiswa Menggunakan Metode Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 83-88.

Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 125-130.

Nugraha, E. V., Ariyana, R. Y., Nurnawati, dan Kumalasari, E. (2022). Uji Black Box Test Aplikasi Software Developmen System Information (SODEVI) PT. Dimata Sorajayate Menggunakan Katalon Studio. *Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST)* (pp. 1-6). Yogyakarta: Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST).

Nurudin, M., Jayanti, W., Saputro, R. W., Saputra, M. P., dan Yulianti. (2019). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 143-148.

OKTAL : Jurnal Ilmu Komputer dan Science
Volume 2, No. 6, Juni 2023
ISSN 2828-2442 (media online)
Hal 1769-1779

Pratama, B. P., Ristianto, V., Prayogo, I. A., Nasrullah, dan Saifudin, A. (2020). Pengujian Perangkat Lunak Sistem Informasi Penilaian Mahasiswa dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing. *Journal of Artificial Inteligence and Innovative Applications*, 32-36.

Shaleh, I. A., Prayogi, J., Pirdaus, P., Syawal, R., dan Saifudin, A. (2021). Pengujian Black Box pada Sistem Informasi Penjualan Buku BerbasisWeb dengan Teknik Equivalent Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 38-45.

Sudana, I. M., Febiharsa, D., dan Hudallah, N. (2018). Uji Fungsionalitas (Blackbox Testing) Sistem Informasi Lembaga Sertifikasi Profesi (SILSP) Batik dengan AppPerfect Web Test dan Uji Pengguna. *Joined Journal, Volume 1*, Nomor 2.

Wulandari, A. S., Saepudin, A., Kinanti, M. P., Sudesi, Z., Saifudin, A., & Yulianti. (2022). Pengujian Aplikasi Sistem Informasi Akademik Berbasis Web Menggunakan Metode Black Box Testing Equivalence Partitioning. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 102-109.