

IMPLEMENTASI WHITE BOX TESTING BERBASIS PATH PADA FORM LOGIN APLIKASI BERBASIS WEB

Muhammad Irsyad Shiddiq

Program Studi Informatika Fakultas Teknik Universitas Siliwangi Tasikmalaya
197006044@student.unsil.ac.id

Abstrak

Form login biasanya dirancang sebagai pintuk masuk bagi pengguna untuk mendapatkan hak akses pada suatu aplikasi. Dalam proses pengisian *form login* setidaknya diperlukan *username* dan *password* yang valid. Tujuan penelitian ini untuk melakukan pengujian pada *form login* menggunakan metode *whitebox* dengan teknik penghitungan *cyclomatic complexity* untuk pengukuran kuantitatif terhadap kompleksitas suatu program. Perhitungan *cyclomatic complexity* dilakukan dengan menggunakan pendekatan matematis: $V(G) = R$, $V(G) = E - N + 2$, dan $V(G) = P + 1$. Perhitungan *cyclomatic complexity* ini digunakan untuk membuat jalur independen yang akan digunakan pada tes skenario. Sebuah aplikasi berbasis web yang menyediakan feature login dikembangkan untuk dicoba pada penelitian ini. Hasil percobaan pada penelitian ini *cyclomatic complexity* pada proses *login* dengan nilai 2, artinya struktur program sederhana dengan tingkat kemungkinan kesalahan yang rendah.

Kata kunci : whitebox, pengujian, login, path testing

Abstract

The login form is usually designed as an entry point for users to gain access rights to an application. In the process of filling out the login form, at least a valid username and password are required. The purpose of this research is to test the login form using the whitebox method with the cyclomatic complexity calculation technique for quantitative measurement of the complexity of a program. The cyclomatic complexity calculation is carried out using a mathematical approach: $V(G) = R$, $V(G) = E - N + 2$, and $V(G) = P + 1$. This cyclomatic complexity calculation is used to create independent paths that will be used in scenario test. A web-based application that provides a login feature was developed to be tested in this study. The experimental results in this study are cyclomatic complexity in the login process with a value of 2, meaning that the program structure is simple with a low probability of error.

Keywords : whitebox, testing, login, path testing

I. PENDAHULUAN

Pengujian perangkat lunak merupakan hal yang kritis dari jaminan kualitas perangkat lunak dan penting dilakukan sebelum produk perangkat lunak diserahkan kepada pelanggan[2]. *Testing* perlu dilakukan, karena tidak ada jaminan bahwa suatu perangkat lunak bebas dari kesalahan. Sebuah pengujian yang sukses jika mampu menemukan semua kesalahan yang ada pada program yang belum terjadi sebelumnya[3]. *Testing* merupakan salah satu tahapan yang biasa dilalui dalam proses pengembangan perangkat lunak. Terdapat beberapa tahapan yang biasa dilalui dalam aktifitas ini, diantaranya: *requirement analysis*, *test planning*, *test case development*, *test environment setup*, *test execution*, *test cycle closure*.

Terdapat dua metode *testing* perangkat lunak yang umum digunakan: *whitebox testing*, dan *black box testing*. *Whitebox testing* merupakan pengujian yang

dikembangkan berdasarkan pada kode program, sedangkan pengujian *blackbox* adalah pengujian fungsional yang dibuat berdasarkan spesifikasi dari klien[1]. Pada *whitebox testing* terdapat beberapa teknik dalam pengujiannya yaitu: *Data Flow Testing*, *Control Flow Testing*, *Basis Path* atau *Path Testing*, dan *Loop Testing* [1].

Pada penelitian ini digunakan salah satu teknik *whitebox testing* yaitu *basis path* atau *path testing*. *Basis path* adalah suatu jalur unik yang melintasi alur program dan tidak diperbolehkan terjadinya perulangan lintasan yang sama[4]. Pada penelitian ini dilakukan pengujian terhadap aplikasi web sederhana yang menggunakan bahasa pemrograman *javascript* pada bagian *form login*. Tujuan dari pengujian ini agar proses *login* dapat dilakukan tanpa adanya kesalahan.

II. METODOLOGI PENELITIAN

White box testing

Whitebox testing merupakan pengujian yang dikembangkan berdasarkan pada kode program. Penguji dalam *whitebox testing* harus memiliki pengetahuan tentang kode dan penulisan kasus uji dengan parameter yang sesuai. Hal ini terutama menyangkut dengan aliran kontrol dan aliran data suatu program[1].

Pada pengujian whitebox terdapat beberapa teknik yang dapat dilakukan diantaranya: *Data Flow Testing*, *Control Flow Testing*, *Basic Path* atau *Path Testing*, dan *Loop Testing*[1].

Basis Path Testing

Basis path testing adalah teknik *whitebox testing* yang pertama dikemukakan oleh Tom McCabe[5]. Metode *basis path* memungkinkan perancang *test case* untuk membuat pengukuran kompleksitas logikal dari rancangan prosedural dan menggunakan pengukuran ini sebagai panduan untuk mendefinisikan himpunan basis dari jalur eksekusi. *Test case* yang dibuat untuk menguji himpunan basis dijamin akan mengeksekusi setiap *statement* di dalam program sekurangnya sekali pada saat pengujian[6]. Langkah-langkah untuk melakukan pengujian *basis path* adalah[2]:

- Menggambarkan grafik alir (*flow graph*) berdasarkan algoritma perancangan prosedur atau fungsi.
- Menentukan *cyclomatic complexity*.
- Menentukan jalur-jalur dasar sesuai dengan jumlah dari *cyclomatic complexity*.
- Mendefinisikan kasus-kasus uji untuk setiap jalur dasar yang telah ditentukan.

Flow Graph

Grafik alir (*Flow Graph*) adalah sebuah notasi sederhana yang merepresentasikan aliran kontrol dari sebuah struktur program. Dalam sebuah grafik alir, anak panah disebut sebagai sisi (*edge*, E) yang merepresentasikan aliran kontrol, kemudian lingkaran disebut sebagai simpul (*node*, N) yang merepresentasikan satu atau lebih aksi/ Pernyataan logis, lalu daerah yang dibatasi oleh sisi dan simpul disebut area (*region*, R). Simpul yang mengandung keputusan disebut sebagai (*predicate node*, P) yaitu simpul yang mengeluarkan lebih dari satu sisi[7].

Cyclomatic Complexity

Cyclomatic complexity, $V(G)$ adalah sebuah besaran perangkat lunak yang menyatakan ukuran tingkat kompleksitas sebuah program. Angka ini menentukan jumlah jalur dasar yang harus diuji minimal sekali dari sebuah program[5]. Secara matematis, $V(G)$ dapat ditentukan salah satu cara dari beberapa pendekatan berikut:

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

Berdasarkan studi yang telah dibuat oleh beberapa industri menyatakan semakin besar nilai $V(G)$ maka semakin besar probabilitas terjadinya kesalahan program[2].

HTML

Hyper Text Markup Language (HTML) merupakan bahasa pemrograman dasar dalam pembuatan website[8]. HTML terdiri dari *Head*, *Body* dan di dalamnya terdapat *Tag* dan *Attribute*. walaupun dikatakan sebagai bahasa pemrograman, tetapi HTML belum dapat dikatakan sebagai bahasa pemrograman karena HTML tidak memiliki hal-hal yang dibutuhkan oleh bahasa pemrograman yaitu logika, HTML hanya memberikan output, maka dari itu HTML di ibaratkan sebagai pondasi atau struktur dari Web dan yang menjadi bahasa pemrograman nya adalah *Javascript*, *PHP*, dan lain – lain[9].

CSS

Cascading Style Sheet (CSS) adalah suatu aturan untuk mengatur tampilan dari website sehingga tampilan dalam web lebih terstruktur. CSS sendiri bukanlah bahasa pemrograman, CSS lebih seperti konfigurasi tampilan dari suatu tag pada website. CSS dapat merubah teks, warna, latar belakang dan posisi dari suatu *tag*[9].

Javascript

Javascript adalah bahasa pemrograman untuk sisi klien atau *client side*[10]. *Javascript* dibuat untuk memperkaya fitur atau fungsi pada *website* agar lebih dinamis, seperti untuk menampilkan dan menghilangkan objek-objek pada *website* kemudian dengan fungsi *javascript* dapat memanggil kembali objek yang di hilangkan tersebut[9].

Tujuan dari penelitian ini untuk menerapkan *whitebox testing* dengan teknik *basis path* atau *path testing* dengan menggambarkan *flowgraph* dan menentukan *cyclomatic complexity*.

Terdapat beberapa tahap dalam proses penelitian yaitu: tahap analisis data, proses pengerjaan, prosedur cara kerja sistem, dan skenario pengujian sistem.

Analisis Data

Data – data yang digunakan dalam penelitian ini berasal dari *source code* yang digunakan dalam pembuatan prototype aplikasi berbasis web, terdiri dari 4 file sebagai berikut:

- index.html
- style.css
- script.js
- success.html

Pengujian difokuskan pada *file script.js* yang dibuat dengan bahasa pemrograman *javascript* untuk diuji fungsinya.

Proses Pengerjaan

Proses yang diuji diambil dari *file script.js* dengan tujuan menguji fungsi – fungsi yang ada pada file tersebut tahapan pada proses pengerjaan ini yaitu[1]:

- membuat *flowchart* dari *source code*
- membuat *flowgraph* dari *flowchart*
- menentukan nilai dari *flowgraph*
- menghitung *cyclomatic complexity*
- membuat jalur independen
- membuat *test case* dari skenario.

Alur atau proses tersebut sesuai dengan teknik *path testing* pada *whitebox testing*.

Prosedur Cara Kerja Sistem

Prosedur cara kerja sistem didapat dengan panduan dari pembuatan program aplikasi yang akan diujikan[1].

Skenario Pengujian Sistem

Skenario pengujian sistem diambil dari hasil jalur independen dari perhitungan *cyclomatic complexity*, format tabel skenario pengujian sistem ditampilkan pada tabel 1.

Tabel 1. Contoh tabel test case

No	kegiatan	Hasil yang diharapkan	hasil	Keterangan
1	-	-	-	-
2	-	-	-	-
3	-	-	-	-
4	-	-	-	-
5	-	-	-	-

III. HASIL DAN PEMBAHASAN

Pengujian difokuskan pada form login (*script.js*), dengan potongan kode program seperti ditampilkan pada gambar 1.

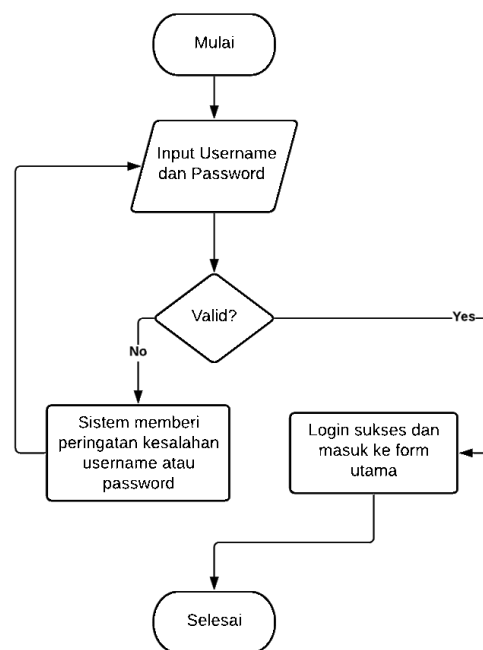
```
function validate() {
    var username =
document.getElementById("username").value
;
    var password =
document.getElementById("password").value
;
    if (username == "irsyad" && password ==
"irsyad123") {
        alert("Login berhasil");
        window.Location = "success.html";
        return false;
    } else {
        alert("Login gagal username atau
password salah");
        window.Location = "index.html";
        if (attempt == 0) {
document.getElementById("username").disab
led = true;

document.getElementById("password").disab
led = true;

document.getElementById("submit").disab
led = true;
            return false;
        }
    }
}
```

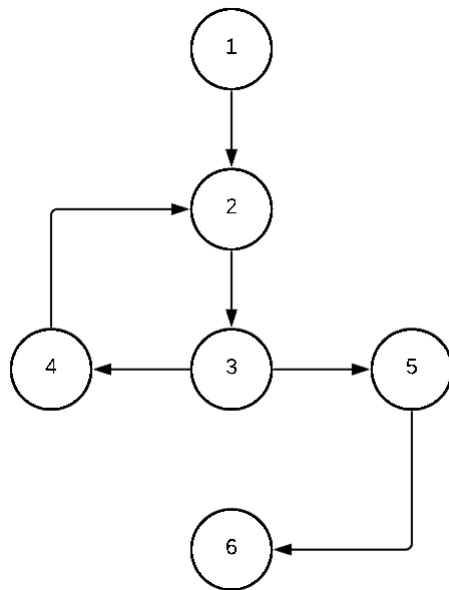
Gambar 1. Potongan kode program fungsi login.

Alur pengujian fungsi login ditampilkan seperti pada gambar 2.



Gambar 2. Flowchart fungsi login.

Flowgraph dari Fungsi login ditampilkan seperti pada gambar 3.



Gambar 3. Flowgraph Fungsi Login.

Gambar 3 menampilkan flowgraph dari flowchart fungsi login pada gambar 2. Dari gambar 3 diketahui node dari fungsi login berjumlah 6, edge berjumlah 6, region berjumlah 2, dan predicate node berjumlah 1. Setelah diketahui nilai dari flowgraph, maka untuk perhitungan flowgraph adalah :

1. $V(G) = R$
 $= 2$
2. $V(G) = E - N + 2$
 $= 6 - 6 + 2$
 $= 2$
3. $V(G) = P + 1$
 $= 1 + 1$
 $= 2$

Dari perhitungan flowgraph, didapatkan jalur indepen sebagai berikut :

1. 1-2-3-5-6 ($if(3) = true$)
2. 1-2-3-4-2-3-5-6 ($if(3) = false$)

Setelah dibuat jalur independen selanjutnya buat tabel test case dengan format yang sudah dijelaskan sebelumnya, test case seperti ditampilkan pada tabel 2.

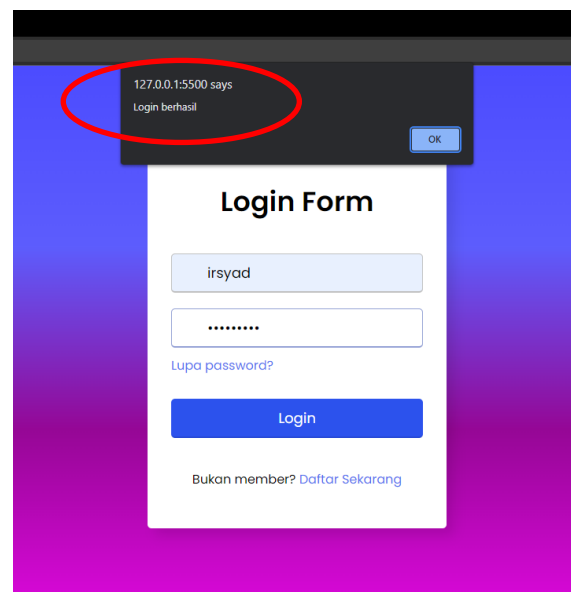
Tabel 2. Test Case

No	Aktifitas	Hasil yang diharapkan	hasil	Kete rangan
1	Memasukan username dan password yang sesuai	Masuk ke halaman form utama	Masuk ke halaman form utama	valid
2	Memasukan username dan password yang tidak sesuai (username salah, password benar),(username benar password salah), (tidak mengisi username dan password)	Muncul peringatan kesalahan username dan password dan tetap di halaman form login	Muncul peringatan kesalahan username dan password dan tetap di halaman form login	valid

Dari tabel 2, berdasarkan jalur independen yang didapat dari hasil perhitungan Cyclomatic Complexity dengan melakukan dua pengujian, didapatkan hasil keduanya valid, artinya tidak ditemukan kesalahan logika pada fungsi login.

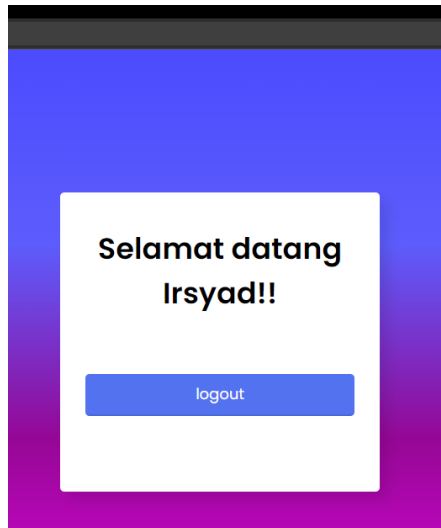
Hasil Uji Coba

Pada bagian ini ditampilkan hasil dari test case yang telah disiapkan sebelumnya sesuai dengan tabel 2. Aktifitas pertama, setelah **memasukan username dan password yang sesuai**, maka tampil pesan “login berhasil” seperti tampil pada gambar 4.



Gambar 4. Uji coba memasukan username dan password yang sesuai (valid)

Aktifitas pertama berhasil **masuk ke halaman form login**, maka dinyatakan **valid** seperti ditampilkan pada gambar 5.

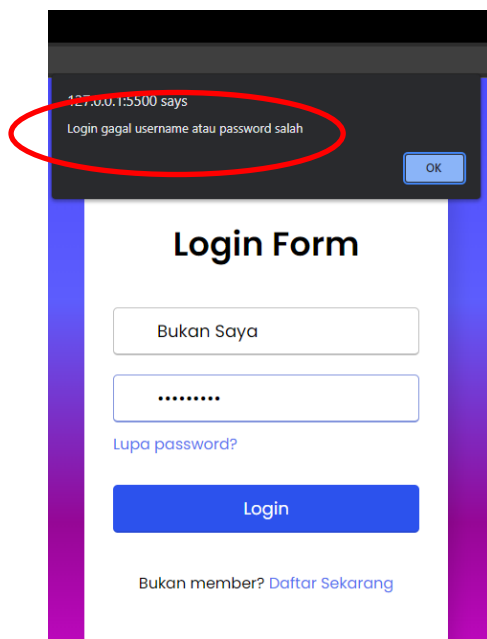


Gambar 5. Hasil uji coba *username* dan *password* yang valid

Pada Aktifitas kedua, **memasukan username dan password yang tidak sesuai**, terdapat tiga skenario yang dilakukan, diantaranya:

- a. username salah, password benar
- b. username benar password salah
- c. tidak mengisi username dan password

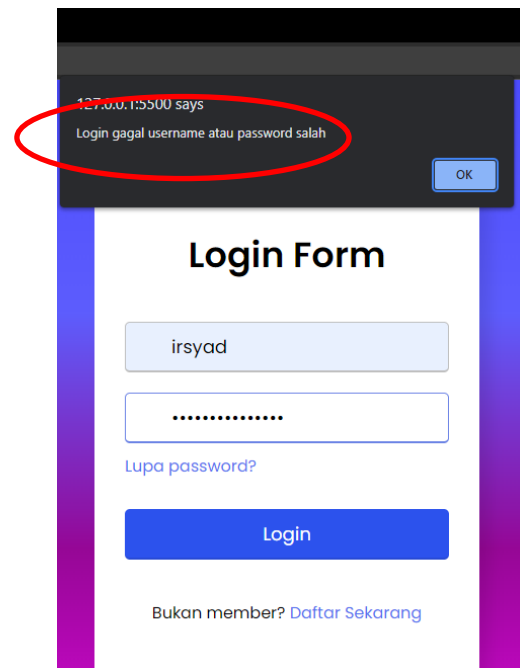
Gambar 6 merupakan tampilan ketika **memasukan username salah dan password benar**, maka tampil pesan “login gagal user atau password salah”.



Gambar 6. Hasil uji coba jika *username* salah dan *password* benar (tidak valid)

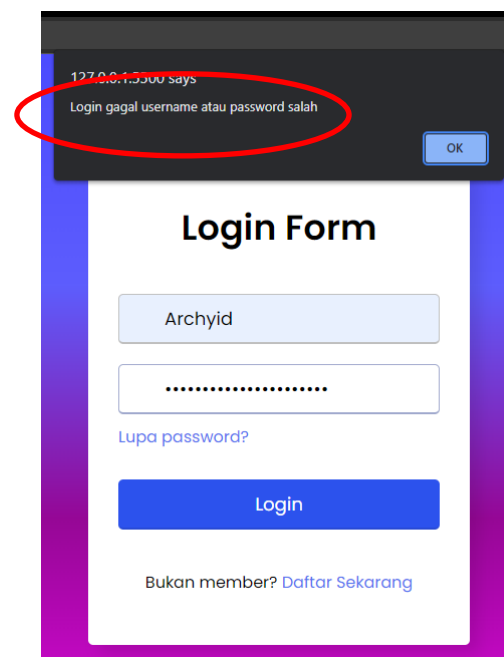
Gambar 7 merupakan tampilan ketika **memasukan username benar dan password salah**,

maka tampil pesan “login gagal user atau password salah”.



Gambar 7. Uji coba jika *username* sesuai dan *password* salah (tidak valid)

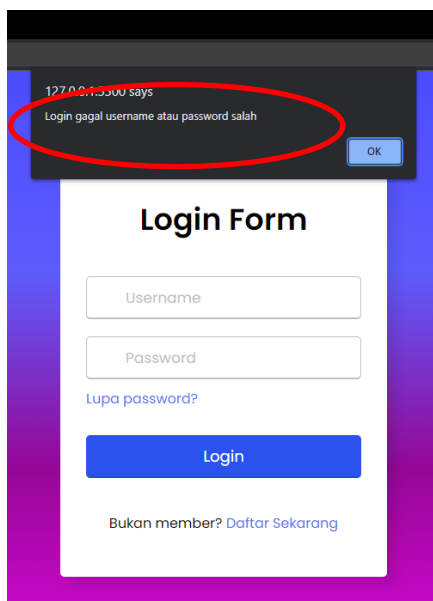
Gambar 8 merupakan tampilan ketika **memasukan username salah dan password salah**, maka tampil pesan “login gagal user atau password salah”.



Gambar 8. Hasil uji coba jika *username* dan *password* tidak sesuai (tidak valid)

Gambar 9 merupakan tampilan ketika **tidak**

mengisi username dan password, dan langsung menekan tombol login, maka tampil pesan “login gagal user atau password salah”.



Gambar 9. Hasil uji coba jika username dan password kosong (tidak valid)

IV. KESIMPULAN

Berdasarkan hasil percobaan yang dilakukan pada penelitian ini telah dilakukan whitebox testing berbasis path pada form login. Hasil perhitungan *Cyclomatic complexity* pada form login diperoleh nilai 2, artinya struktur program tergolong sederhana dengan tingkat kemungkinan kesalahan yang rendah.

DAFTAR PUSTAKA

- [1] R. Subagia, R. Alit, and F. A. Akbar, “Penguujian white box pada sistem informasi monitoring skripsi program studi informatika,” *J. Inform. dan Sist. Inf.*, vol. 01, no. 2, pp. 539–547, 2020.
- [2] T. A. Kurniawan, “Penguujian Struktur Program Dengan Penguujian Jalur Dasar (Basis Path Testing): Teori Dan Aplikasi,” *Eeccis*, vol. 1, no. 1, pp. 29–32, 2007, [Online]. Available: <http://jurnaleeccis.ub.ac.id/index.php/eccis/article/viewFile/357/266>.
- [3] R. Mubarak, “Implementasi Metode White Box Testing Pada Proses Quality Assurance Perangkat Lunak Berbasis Web Dan Mobile Collection System,” *J. Teknol. Inf. ESIT*, vol. XV, no. 10, pp. 57–63, 2020.
- [4] C. T. Pratala, E. M. Asyer, I. Prayudi, and A. Saifudin, “Penguujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path,” *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 111, 2020, doi: 10.32493/informatika.v5i2.4713.
- [5] M. E. Khan, “Different approaches to white box testing technique for finding errors,” *Int. J. Softw. Eng. its Appl.*, vol. 5, no. 3, pp. 1–14, 2011, doi: 10.5121/ijsea.2011.2404.
- [6] Handy and J. Susilo, “Aplikasi Penguujian White-Box Ibi Online Judge,” *J. Inform. dan Bisnis*, vol. 3, pp. 56–68, 2014.
- [7] A. Widiyanto, “Penguujian Perangkat Lunak,” *FASILKOM Univ. Indones.*, no. September, pp. 1–4, 2008, [Online]. Available: http://lib.ui.ac.id/file?file=digital/123518-SK-740-Penguujian_perangkat-Pendahuluan.pdf.
- [8] A. Yani, B. Saputra, and R. T. Jurnal, “Rancang Bangun Sistem Informasi Evaluasi Siswa Dan Kehadiran Guru Berbasis Web,” *Petir*, vol. 11, no. 2, 2018, doi: 10.33322/petir.v1i2.344.
- [9] Marlina, Masnur, and M. Dirga.F, “Aplikasi E-Learning Siswa Smk Berbasis Web,” *J. SINTAKS Log. Vol.*, vol. 1, no. 1, pp. 2775–412, 2021.
- [10] O. Pahlevi, A. Mulyani, and M. Khoir, “Sistem Informasi Inventori Barang Menggunakan Metode Object Oriented Di Pt. Livaza Teknologi Indonesia Jakarta,” *J. PROSISKO*, vol. 5, no. 1, 2018, [Online]. Available: <https://livaza.com/>.