

## Pengujian Black Box pada Aplikasi Perpustakaan Berbasis Web Menggunakan Teknik Equivalence Partitioning

Deny Setiawan<sup>1</sup>, Muhammad Ardiansyah Fadhillah<sup>2</sup>, Anggun Wibawa<sup>3</sup>, Irwan Sugiarto<sup>4</sup>, Agus Mulyana<sup>5</sup>, Irpan Kusyadi<sup>6</sup>

Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Banten, Indonesia, 15416  
e-mail: <sup>1</sup>denysetiawan04@hotmail.com, <sup>2</sup>dmafd03@gmail.com, <sup>3</sup>anggunwibawa004@gmail.com,  
<sup>4</sup>irwanjavier21@gmail.com, <sup>5</sup>agusmulyana@gmail.com, <sup>6</sup>dosen00673@unpam.ac.id

Submitted Date: January 05<sup>th</sup>, 2020

Reviewed Date: January 06<sup>th</sup>, 2020

Revised Date: January 18<sup>th</sup>, 2020

Accepted Date: April 29<sup>th</sup>, 2020

### Abstract

*In determining the level of quality of a software it is necessary to test the software. Tests tested can include specifications, design, and coding of the software. This test is conducted to test web-based library software. This test uses the Black Box Testing method using the Equivalence Partitioning technique. Black Box Testing is applied in order to find errors in software related to erroneous functions, errors in interface display, data structure errors or access rights in external databases, sub-optimal performance in a process, and errors in initialization or termination. The Equivalence Partitioning technique defines the cases to be tested by finding some errors and minimizing the number of cases that must be made. The Equivalence Partitioning test case design is based on the premise of the input and output of a component which is partitioned into classes according to the specifications of that component, which are treated equally (equivalence). It can also be assumed that the results of an input produce the same response. The expected result of testing this software is to ensure that this software is free from errors so that in its use there will be no obstacles that prevent the user from doing his work. And inform that the software is in accordance with predetermined specifications.*

*Keywords: Software Testing; Black Box; Equivalence Partitioning.*

### Abstrak

Pengujian pada perangkat lunak dilakukan sebagai penentuan tingkat kualitas suatu perangkat lunak.. Pengujian yang diujikan dapat meliputi spesifikasi, desain, dan koding dari perangkat lunak tersebut. Pengujian ini dilakukan untuk menguji perangkat lunak perpustakaan berbasis web. Pengujian ini menggunakan metode *Black Box* dengan menggunakan teknik *Equivalence Partitioning*. Metode pengujian *Black Box* diterapkan guna untuk mencari kesalahan dalam perangkat lunak yang berkaitan dengan fungsi-fungsi yang keliru, adanya kesalahan dalam tampilan antarmuka, kesalahan struktur data atau hak akses dalam database eksternal, kurang optimal dalam suatu proses pengerjaan, serta kesalahan dalam inialisasi atau terminasi. Design uji kasus *Equivalence Partitioning* berdasarkan pada premis *input* dan *output* dari suatu komponen yang di partisi ke dalam kelas-kelas sesuai dengan spesifikasi dari komponen itu, yang diperlakukan sama (*equivalence*). Dapat juga dianggap bahwa hasil dari suatu input menghasilkan respon yang sama. Hasil yang di harapkan dari pengujian perangkat lunak ini adalah untuk memastikan bahwa perangkat lunak ini sudah bebas dari kesalahan-kesalahan sehingga dalam penggunaannya nanti tidak terjadi kendala yang menghambat pengguna dalam mengerjakan tugasnya. Serta menginformasikan bahwa perangkat lunak sudah sesuai dengan spesifikasi yang sudah ditentukan sebelumnya.

Kata Kunci: Pengujian Perangkat Lunak; Black Box; Equivalence Partitioning.

### 1. Pendahuluan

Saat ini perkembangan teknologi dan komputer sudah semakin cepat dan berdampak

pada kehidupan manusia. Sekarang ini hampir seluruh kegiatan manusia sudah melibatkan berbagai macam perangkat teknologi termasuk

dalam melakukan pengelolaan data – data informasi. Sudah banyak perangkat lunak baik yang berbasis *desktop* maupun *web* dibuat agar dapat memudahkan suatu pekerjaan manusia. Tentunya, perangkat lunak harus menjalankan tugasnya dengan baik sehingga tidak mengalami kendala yang menghambat pekerjaan pengguna. Perkembangan teknologi diikuti meningkatnya penggunaan dan kompleksitas software, sehingga jaminan kualitas software menjadi kritis dan tantangan yang sulit (Saifudin, Heryadi, & Lukas, 2019). Maka dari itu untuk menjamin kualitas suatu perangkat lunak diperlukan pengujian, pengendalian, dan pengelolaan sebagai acuan pada kualitas perangkat lunak .

Menurut Fournier dalam (MZ & Komarudin, 2016), pengujian sebaiknya menemukan kesalahan yang tidak disengaja dan pengujian dinyatakan sukses jika berhasil memperbaiki kesalahan tersebut. Selain itu, pengujian juga bertujuan untuk menunjukkan kesesuaian fungsi-fungsi perangkat lunak dengan spesifikasinya. Sebuah perangkat lunak dinyatakan gagal, jika perangkat lunak tersebut tidak memenuhi spesifikasi. Pengujian ini dilakukan pada aplikasi perpustakaan berbasis *web*. Aplikasi ini dirancang agar dapat membantu mempermudah petugas perpustakaan dalam melakukan kegiatan pengelolaan data – data perpustakaan seperti pendaftaran anggota, peminjaman dan pengembalian buku. Sehingga pelaksanaan kegiatan pengelolaan data perpustakaan menjadi lebih mudah dan cepat sehingga menghemat waktu dan menghindari kesalahan akibat pemrosesan data secara manual.

Sebelum perangkat lunak ini digunakan, maka diperlukan pengujian guna menemukan kesalahan yang tidak diinginkan. Aplikasi perpustakaan berbasis *web* ini akan diuji dengan menggunakan metode *Black Box Testing* dengan teknik *Equivalence Partitioning*. Metode tersebut dipilih dikarenakan metode dan teknik pengujian tersebut dapat memastikan fitur – fitur yang terdapat dalam perangkat lunak dapat berfungsi dengan baik. *Black Box Testing* merupakan pengujian yang berkaitan dengan hal – hal yang tidak tercakup dengan menggunakan metode *White Box Testing*.

Banyak desain rancangan yang sudah dibuat dalam melakukan pengujian perangkat lunak. Dalam pengujian ini akan ada beberapa tahap. Tahapan – tahapan tersebut akan dijelaskan pada bab selanjutnya.

## 2. Metodologi

Untuk menjamin software yang dihasilkan telah bebas dari kesalahan umumnya dilakukan dengan pengujian yang merupakan tahapan paling mahal dalam pengembangan software (Saifudin & Yulianti, 2020). Pengujian adalah satu kegiatan yang direncanakan secara teratur dan terencana untuk menguji atau mengevaluasi kebenaran yang diinginkan (Zuriati, Widyawati, Sitanggung, & Buowo, 2018). Pengembang atau penguji software harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan dapat dideteksi sejak awal dan dikoreksi secepatnya. (Sutanto, Utomo, & Perbawa, 2018). Menurut Shi dalam (Mustaqbal, Firdaus, & Rahmadi, 2015) pengujian atau testing sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan *software* seperti halnya analisis, desain, dan pengkodean.

Pengujian software haruslah dilakukan dalam proses rekayasa perangkat lunak atau software engineering. Sejumlah strategi pengujian software telah diusulkan dalam literatur. Semuanya menyediakan template untuk pengujian bagi pembuat software. Dalam hal ini, semuanya harus memiliki karakteristik umum berupa (Sutanto, Utomo, & Perbawa, 2018) :

- a. Testing dimulai pada level modul dan bekerja keluar ke arah integrasi pada sistem berbasis komputer.
- b. Teknik testing yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya.
- c. Testing diadakan oleh pembuat/pengembang software dan untuk proyek yang besar oleh group testing yang independent.
- d. Testing dan Debugging adalah aktivitas yang berbeda tetapi debugging harus diakomodasikan pada setiap strategi testing.

Metode *Blackbox Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan, estimasi banyaknya data uji dapat dihitung melalui banyaknya *field* data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi (Cholifah, Yulianingsih, & Sagita, 2018). *Black Box Testing* sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional

dari perangkat lunak (Hanifah, Alif, & Sugiarto, 2016).

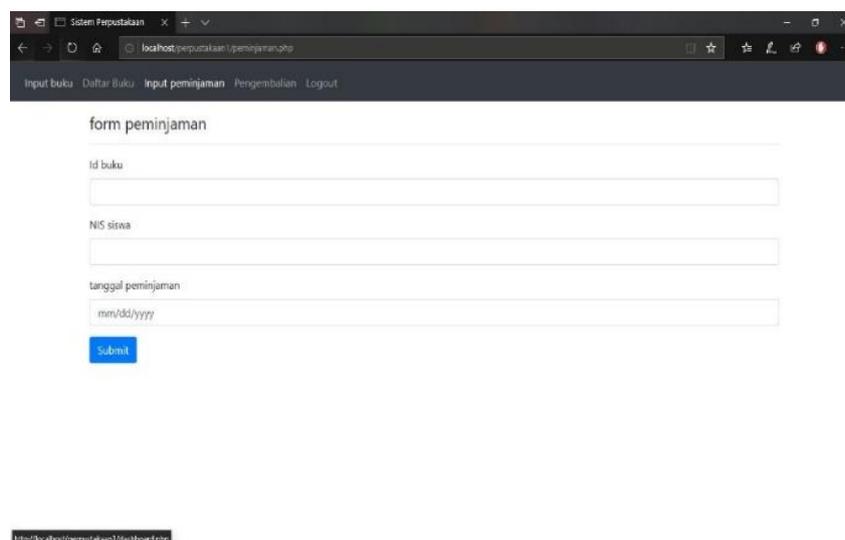
*Black Box Testing* berfokus pada pengujian dari masing-masing spesifikasi fungsional perangkat lunak. Ujicoba *Black-Box* dilakukan untuk menemukan kesalahan dalam beberapa kategori yaitu: (1) Fungsi-fungsi yang hilang atau salah; (2) Kesalahan desain antarmuka (*interface*) atau tampilan; (3) Kesalahan dalam struktur data atau akses *database* eksternal; (4) Kesalahan performa; dan (5) Kesalahan inisialisasi dan terminasi (Febiharsa, Sudana, & Hudallah, 2018).

Keuntungan penggunaan metode *Blackbox Testing* adalah : (1) Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu; (2) Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan; (3) *Programmer* dan *tester* keduanya saling bergantung satu sama lain (Jaya, 2018). Pengujian *Equivalence Partitioning* merupakan pengujian berdasarkan inputan setiap menu yang terdapat pada sistem, setiap menu inputan dilakukan pengujian melalui klasifikasi dan pengelompokan berdasarkan fungsinya (Hidayat & Muttaqin, 2018). Teknik ini (*Equivalence Partitioning*) dapat di terapkan jika sekumpulan data pada kelas kesetaraan (*equivalence class*) telah disusun dan memiliki batasan yang dapat diidentifikasi (Andriansyah, 2018). Langkah – langkah yang harus disiapkan dalam menganalisa partisi pada teknik ini, yaitu :

- a. Menyiapkan model komponen berupa partisi dari nilai *input* dan nilai *output* komponen.

- b. Spesifikasi dari komponen untuk membuat *input* dan *output*.
- c. Partisi merupakan kumpulan nilai yang dipilih dan diharapkan akan diperlakukan dengan cara yang sama oleh komponen.
- d. Harus menentukan partisi yang bernilai valid dan tidak valid.

Pada pengujian kali ini dilakukan pengujian pada halaman form peminjaman buku di perangkat lunak *web* perpustakaan (Gambar 1). Di halaman form tersebut, terdapat 3 kolom yang harus diisi oleh pengguna jika ingin melakukan pendataan peminjaman buku yaitu kolom ID Buku, NIS Siswa, serta tanggal peminjaman. Pada rencana pengujian kolom ID Buku hanya akan valid jika kolom ID Buku diisi dengan memasukkan angka “123” yang maksimal jangkauan datanya adalah 3 digit dan tidak boleh kosong. Data tidak valid jika memasukan selain dari angka. Pada rencana pengujian kolom Nis Siswa juga sama seperti kolom ID Buku yaitu data akan valid jika kolom Nis Siswa diisi angka “123” tetapi harus di isi dengan minimal 5 – 10 digit. Kolom Nis Siswa juga tidak akan valid jika diisi dengan selain angka dan tidak boleh kosong. Pada rencana pengujian kolom Tanggal Peminjaman sama seperti pada kolom – kolom sebelumnya yaitu data akan valid jika diisi dengan angka dengan format bulan/tanggal /tahun “mm/dd/yyyy”. Kolom Tanggal Peminjaman juga tidak valid jika diisi dengan selain angka dan format yang sudah ditentukan dan juga tidak boleh kosong.



The image shows a screenshot of a web browser window. The address bar shows the URL 'localhost:pepusakaan/peminjaman.php'. The page title is 'form peminjaman'. There are three input fields: 'Id buku', 'NIS siswa', and 'tanggal peminjaman'. The 'tanggal peminjaman' field has a placeholder 'mm/dd/yyyy'. Below the fields is a blue 'Submit' button. The browser's navigation bar shows 'Input buku', 'Daftar buku', 'Input peminjaman', 'Pengembalian', and 'Logout'.

Gambar 1. Form Peminjaman Buku

Sesuai dengan tahapan pengujian menggunakan teknik Equivalence Partitioning, maka ditentukan terlebih dahulu test-case sebelum akhirnya dilakukan pengujian. Rancangan test case akan dijabarkan pada tabel dibawah ini :

Tabel 1. Rancangan Test Case Pengujian Form Peminjaman

Id	Deskripsi pengujian	Hasil yang diharapkan
P01	Mengisi kolom ID Buku dengan "001", kolom Nis Siswa dengan "123456", dan kolom Tanggal Peminjaman dengan "01/10/2020" kemudian klik tombol Submit.	Sistem berhasil menyimpan penambahan data ke database dan menampilkan pesan sukses.
P02	Mengisi kolom ID Buku dengan "abc001" atau tidak diisi, kolom Nis Siswa dengan "567" atau tidak diisi, dan kolom Tanggal Peminjaman dengan "20201221" atau tidak diisi kemudian klik tombol Submit.	Sistem menolak untuk menyimpan data

### 3. Hasil dan Pembahasan

Berdasarkan pada penjelasan yang sudah dijelaskan sebelumnya bahwa pengujian ini akan dilakukan menggunakan teknik *Black Box* dengan metode *Equivalence Partitioning*. Pengujian dilakukan pada halaman form peminjaman buku. Setelah diterapkan pengujian dengan metode tersebut, berikut hasilnya dijabarkan pada tabel di bawah ini:

Tabel 2. Hasil Pengujian

Id	Deskripsi pengujian	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
P01	Mengisi kolom ID Buku dengan "001", kolom Nis Siswa dengan "123456", dan kolom Tanggal Peminjaman dengan "01/10/2020" kemudian klik tombol Submit.	Sistem berhasil menyimpan penambahan data ke database dan menampilkan perubahan tampilan pada halaman sistem.	Sistem menampilkan pop up pesan "Data Sukses Disimpan" serta kolom – kolom pada form tidak terisi atau sudah ter-reset.	Sesuai
P02	Mengisi kolom ID Buku dengan "abc001" atau tidak diisi, kolom Nis Siswa dengan "567" atau tidak diisi, dan kolom Tanggal Peminjaman dengan "20201221" atau tidak diisi kemudian klik tombol Submit.	Sistem menolak untuk menyimpan data	Sistem menampilkan pesan peringatan "Mohon lengkapi kolom – kolom form sesuai dengan yang sudah ditentukan".	Sesuai

### 4. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan menggunakan tahapan-tahapan yang ada pada metode *Equivalence Partitioning*, dapat terlihat bahwa perangkat lunak *web* perpustakaan yang diujikan pada form halaman peminjaman memiliki fungsi-fungsi yang sudah berjalan dengan baik sesuai dengan yang diharapkan. Ini pun membuktikan bahwa metode *Equivalence*

*Partitioning* dapat digunakan dengan cukup mudah dalam melakukan pengujian aplikasi.

## 5. Saran

Untuk lebih memastikan aplikasi tersebut terhindar dari kesalahan sehingga terjamin kualitasnya, maka disarankan agar dapat melakukan pengujian menggunakan metode *Black Box* yang lainnya sehingga dapat meminimalkan celah kesalahan pada aplikasi *website* tersebut.

## Referensi

- Andriansyah, D. (2018). Pengujian Kotak Hitam Boundary Value Analysis Pada Sistem Informasi Manajemen Konseling Tugas Akhir. *Indonesian Journal on Networking and Security - Vol.7 No.1*, 13-18.
- Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *Jurnal String Vol. 3 No.2 Desember 2018*, 206-210.
- Febiharsa, D., Sudana, I. M., & Hudallah, N. (2018). Uji Fungsionalitas ( Black Box Testing ) Sistem Informasi Lembaga Sertifikasi Profesi (SILSP) Batik dengan Appperfect Web Test dan uji Pengguna. *Joined Journal Vol.1 No.2*, 117-126.
- Hanifah, U., Alif, R., & Sugiarto. (2016). Penggunaan Metode Black Box pada Pengujian Sistem Informasi Surat Keluar Masuk. *SCAN VOL. XI NOMOR 2 JUNI 2016*, 33-40.
- Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jurusan Teknik Informatika FT Universitas Islam Syekh Yusuf*, 25-29.
- Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Black Box Testing Boundary Value Analysis. *Jurnal Pengembangan IT(JPIT)*, 45-48.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Ilmiah Teknologi Informasi Terapan*, 31-36.
- MZ, & Komarudin, M. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions Aplikasi Sistem Informasi Sekolah. *Jurnal Mikrotik*.
- Saifudin, A., & Yulianti, Y. (2020). Dimensional Reduction on Cross Project Defect Prediction. *Journal of Physics: Conference Series. 1477*, hal. 022030. Tangerang: IOP Publishing. doi:10.1088/1742-6596/1477/3/032011
- Saifudin, A., Heryadi, Y., & Lukas. (2019). Ensemble Undersampling to Handle Unbalanced Class on Cross-Project Defect Prediction. *IOP Conference Series: Materials Science and Engineering. 662*, hal. 1-8. Bandung: IOP Publishing. doi:10.1088/1757-899X/662/6/062012
- Sutanto, Y., Utomo, A., & Perbawa, D. S. (2018). Pengujian Aplikasi Website Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal SAINSTECH Politeknik Indonusa Surakarta*, 52-57.
- Zuriati, Widyawati, D. K., Sitanggang, I. S., & Buowo, A. (2018). Teknik Pengujian Boundary Value Analysis pada Aplikasi Learning Management System Polinea. *Jurnal TAM (Technology Acceptance Model) Vol.9, No.2*, 86-92.