



Pengujian *Black Box* pada Aplikasi e-Promkes Berbasis Web Menggunakan Teknik *Equivalence Partitions*

Chelsea Samsi Wijaya¹, Muhammad Dimas Zulfahmi², Antonius Wahyu Sudrajat³

^{1,2,3}Program Studi Sistem Informasi, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang

¹chelseasw@mhs.mdp.ac.id,²dimaszulfahmi7@mhs.mdp.ac.id,³wahyu.sudrajat@mdp.ac.id

Abstract

Software testing is a critical process in information system development, aiming to ensure that the developed software functions according to the expected specifications. In the context of the web-based e-Promkes application, software testing becomes even more significant as this application serves as a primary tool for Mohammad Hoesin Hospital (RSMH) in recording and reporting internal hospital activities. Therefore, comprehensive and structured software testing is essential. This research aims to conduct Black Box Testing on the e-Promkes application using the Equivalence Partitioning (EP) technique, with the primary focus of testing on the PEH form feature. Black Box Testing allows testers to evaluate how the application handles inputs and produces the corresponding outputs. The Equivalence Partitioning technique divides input data into partitions based on valid and invalid values, enabling more effective testing of various input conditions. The research method employed follows the STLC (Software Testing Life Cycle) approach, encompassing various critical stages to ensure that software testing is conducted effectively and meets the predefined requirements. The test results indicate that out of eight test cases, all showed compliance with a 100% conformity rate. This indicates that the PEH form feature on the e-Promkes application functions well according to the established specifications and is capable of providing correct responses to various input conditions.

Keywords: Black Box Testing; Equivalence Partitioning; Information System; Software Testing Life Cycle; Web Application;

1. Introduction

Pada era digital yang terus berkembang seperti saat ini, penggunaan teknologi telah menjadi suatu kebutuhan dalam berbagai bidang, termasuk kesehatan. Teknologi memiliki peran penting dalam meningkatkan efisiensi, mengoptimalkan proses, dan memberikan layanan yang lebih baik kepada masyarakat. Pengembangan sistem informasi adalah salah satu metode untuk memanfaatkan teknologi secara maksimal dalam membantu rumah sakit dalam menjalankan operasionalnya. Pengujian perangkat lunak adalah proses penting dalam pengembangan sistem informasi, yang bertujuan untuk memastikan bahwa perangkat lunak yang dibuat dapat berfungsi sesuai dengan spesifikasi yang diharapkan.

Software Testing melibatkan menjalankan program atau aplikasi yang telah dikembangkan untuk mengidentifikasi masalah seperti *bug*. Pengujian ini sangat penting untuk memastikan perangkat lunak berkualitas tinggi. Karena beberapa aspek sering kali terlewatkan selama pengembangan, pengujian perangkat lunak sangat penting dilakukan untuk menemukan dan memperbaiki masalah tersebut[1].

Dalam konteks aplikasi e-Promkes yang berbasis *web*, pengujian perangkat lunak menjadi lebih signifikan karena aplikasi ini berfungsi sebagai alat utama bagi Rumah Sakit Mohammad Hoesin (RSMH)

dalam proses perekapan dan pelaporan kegiatan *internal* rumah sakit. Oleh karena itu, pengujian yang menyeluruh dan terstruktur menjadi aspek penting untuk memastikan kualitas perangkat lunak yang mencakup elemen utama atau dasar dari desain, spesifikasi, dan pengkodean. Pengujian ini dilakukan dengan menjalankan program untuk mencari kesalahan sintaks, serta melakukan verifikasi program agar sesuai dengan kebutuhan pengguna. Dalam penelitian ini, pengujian dilakukan dengan *Black Box Testing* dengan teknik *Equivalence Partitions*[2].

Black Box Testing memungkinkan pengujian untuk mengevaluasi bagaimana aplikasi menangani *input* dan menghasilkan *output* sesuai dengan yang diharapkan. Pengujian ini sangat efektif untuk menemukan kesalahan dalam fungsionalitas aplikasi, seperti kesalahan dalam proses masukan data, pengolahan data, dan keluaran data, serta memastikan bahwa fungsi-fungsi aplikasi berjalan dengan baik[3]. *Equivalence Partitioning* (EP) adalah teknik pengujian perangkat lunak yang membagi data *input* ke dalam partisi-partisi berdasarkan nilai *valid* dan *invalid*[4]. Langkah pertama dalam fase pengujian *Equivalence Partitions* adalah mengembangkan *test case* berdasarkan karakteristik perangkat lunak yang diuji. Selanjutnya, buat partisi yang setara dengan teknik pengujian. Setelah itu, lakukan pengujian dengan teknik atau deskripsi pengujian untuk mendapatkan hasil yang

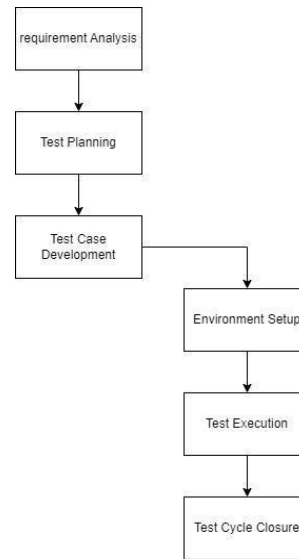
diharapkan. Langkah terakhir adalah membuat uji berdasarkan model yang telah dirancang. Proses ini dilakukan dengan menggunakan hasil dari pengujian *Equivalence Partitions* dan mengevaluasi efektivitasnya.

Berdasarkan hasil penelitian sebelumnya tentang pengujian *Black Box* pada aplikasi e-Commerce OpenCart, dinyatakan bahwa menggunakan teknik *Equivalence Partitioning* dari total 14 tes, sistem menunjukkan tingkat kesesuaian sebesar 57,14%. Artinya, delapan dari 14 tes tersebut berhasil memenuhi persyaratan, sementara enam tes lainnya tidak sesuai[5]. Penelitian sebelumnya juga menunjukkan bahwa teknik *Equivalence Partitioning* bermanfaat dalam mengidentifikasi masalah seperti kesalahan saat menghapus data yang tersimpan dan kesalahan dalam form aplikasi e-Commerce[6]. Studi lain mengenai pengujian *Black Box* pada aplikasi perpustakaan menemukan bahwa penggunaan *Equivalence Partitioning* membantu dalam pembuatan *test case*, uji kualitas, dan penemuan kesalahan yang tersembunyi, seperti kesalahan pengetikan dan celah *error* pada form aplikasi perpustakaan[7].

Tujuan dari penelitian ini adalah untuk melakukan pengujian *Black Box* pada aplikasi e-Promkes dengan menggunakan pendekatan *Equivalence Partitioning* (EP). Fokus utama pengujian adalah fitur form PEH, yang merupakan salah satu proses penting untuk perekapan dan pelaporan aktivitas di rumah sakit. *Form Peh* digunakan untuk menginput data-data *talkshow* edukasi rumah sakit, dan *outputnya* berupa laporan yang dapat dicetak dalam bentuk PDF.

2. Metode Penelitian

Metode penelitian yang digunakan yaitu STLC (*Software Testing Life Cycle*) dengan tahapan alur *Black Box Testing*. Dalam pengujian *Black Box* ini, digunakan data sebagai *input* dengan tujuan untuk memperoleh hasil yang pasti. Artinya, jika data tersebut salah, sistem informasi akan menolaknya atau data *input* tersebut tidak akan tersimpan dalam basis data. Sebaliknya, jika data *input* benar, maka data tersebut akan diterima dan masuk ke dalam basis data sistem informasi[8]. Dari penelitian terdahulu yang telah dibahas sebelumnya, telah dilakukan pengujian *Black Box* dengan teknik *Equivalence Partitions*, tetapi belum ada yang mengimplementasikan STLC. Maka penelitian ini akan melakukan pendekatan STLC sebagai metode pengujian[9]. Untuk memastikan bahwa pengujian perangkat lunak dilakukan secara efisien dan sesuai dengan persyaratan yang telah ditetapkan, proses STLC mencakup berbagai tahapan penting. Tahapan pendekatan STLC ditunjukkan pada Gambar 2.1.

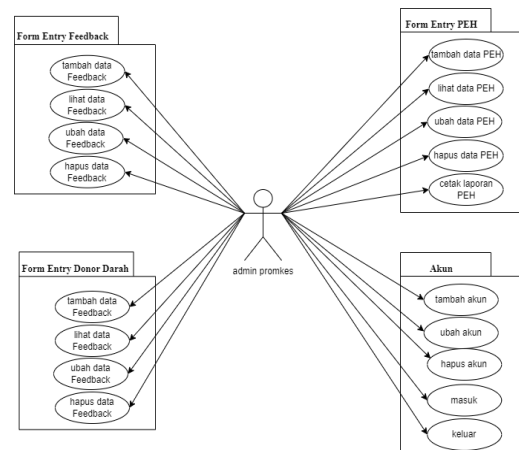


Gambar 2.1 Tahapan *Software Testing Life Cycle*

2.1 Requirement Analysis

Pada tahap ini, analisis kebutuhan perangkat lunak akan dilakukan dari sudut pandang pengguna. Analisis ini akan mencakup identifikasi kebutuhan fungsional dan non-fungsional yang perlu diuji. Persyaratan ini termasuk tampilan pengguna yang mudah dipahami, kemudahan interaksi pengguna dengan sistem, kemampuan pencarian dan penyaringan data, dan responsivitas yang baik terhadap permintaan.

Analisis kebutuhan sistem termasuk perangkat lunak yang akan diuji fiturnya, dengan *use case diagram* dan persyaratan sistemnya. Pada Gambar 2.2 menunjukkan *use case diagram* untuk aplikasi e-Promkes.



Gambar 2.2 *Use Case Diagram*

Peneliti akan menguji salah satu kasus, yaitu formulir PEH, dimulai dengan menambah, mengubah, menghapus, dan mencetak laporan PEH seperti yang

ditunjukkan pada Gambar 2.2. Peneliti juga akan menganalisis persyaratan sistem untuk fitur yang diuji, yang berguna sebagai pedoman untuk pembuatan *test case* pada tahap berikutnya. Tabel 2.1 menunjukkan spesifikasi dan persyaratan sistem untuk fitur *form* PEH.

Tabel 2.1 Persyaratan Sistem pada fitur *form* PEH

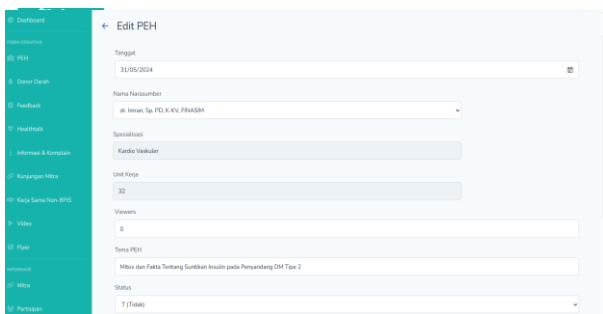
ID	Fitur yang diuji	Kriteria
ID01	Cetak laporan PEH	Mengisi kolom tanggal pada bagian kanan halaman PEH dan memilih fitur cetak
ID02	Tambah data PEH	Menginput data-data yang dibutuhkan
ID03	Lihat data PEH	Melihat data yang telah diinput
ID04	Ubah data PEH	Mengubah data yang diperlukan
ID05	Hapus data PEH	Menghapus data yang sudah ada

2.2 Test Planning

Langkah ini bertujuan untuk merencanakan pengujian dengan mengidentifikasi, menetapkan prioritas, dan menjadwalkan aktivitas yang diperlukan guna memastikan bahwa perangkat lunak memenuhi persyaratan dan berfungsi dengan baik.

2.3. Test Case Development

Proses ini bertujuan untuk menyusun kasus uji yang akan digunakan dalam pengujian otomatis. Ini mencakup serangkaian kasus uji spesifik untuk memastikan kualitas perangkat lunak, dengan fokus pada kepuasan pelanggan[10]. Tabel dan gambar berikut menunjukkan tampilan dan desain *test case*.

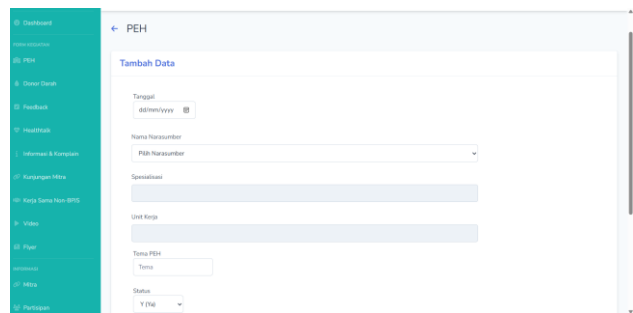


Gambar 2.3 Tampilan *Form* PEH

Berdasarkan pada Gambar 2.3 pada bagian fitur cetak laporan PEH, terdapat rancangan *test case* yang diuji untuk menghapus data kegiatan.

Tabel 2.2 *Test Case Form Edit* Data PEH

Test Case ID	Test Description
TC01	Mengisi kolom tanggal periode waktu dan menekan tombol cetak



Gambar 2.4 Tampilan *Form* Tambah Data PEH

Berdasarkan pada Gambar 2.3 dan Gambar 2.4 *form* tambah PEH, terdapat rancangan *test case* yang diuji untuk menambahkan data kegiatan.

Tabel 2.3 *Test Case Form* Tambah Data PEH

Test Case ID	Test Description
TC02	Memilih fitur tambah data PEH
TC03	Menginput data-data yang dibutuhkan lalu menekan tombol tambah
TC04	Tidak menginput data-data yang dibutuhkan lalu menekan tombol tambah

Gambar 2.5 Tampilan *Form Edit* Data PEH

Berdasarkan pada Gambar 2.3 dan Gambar 2.5 *form edit* PEH, terdapat rancangan *test case* yang diuji untuk mengubah data kegiatan yang ada.

Tabel 2.4 *Test Case Form Edit* Data PEH

Test Case ID	Test Description
TC05	Memilih fitur ubah data PEH

- TC05 Mengubah data yang diperlukan lalu menekan tombol simpan
- TC07 Menghapus data yang sudah ada lalu menekan tombol simpan

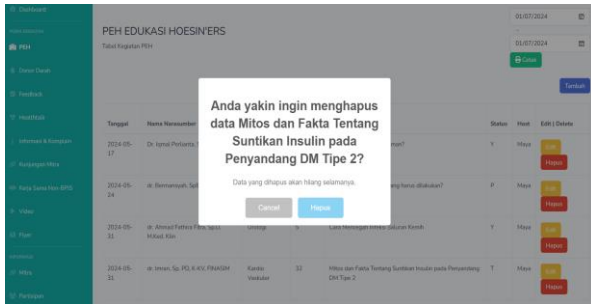
hambatan yang menghalangi proses pengujian serta meningkatkan kualitas STLC di masa mendatang.

3. Hasil dan Pembahasan

Dari kegiatan sebelumnya, yaitu tahap pembuatan *test case*, dihasilkan 8 tes untuk metode *Equivalence Partitioning*. *Test case* yang telah disusun tersebut kemudian akan diuji. *Test case* beserta hasil pengujiannya dapat dilihat pada Tabel 3.1.

Tabel 3.1 *Test Case Form PEH*

Test Case ID	Test Description	Nilai	Hasil yang diharapkan	Hasil uji coba
TC01	Memilih fitur tambah data PEH	Valid	Menampilkan detail <i>input form</i>	Sesuai
TC02	Menginput data-data yang dibutuhkan di <i>form</i> lalu menekan tombol tambah	Valid	Akan muncul notifikasi bahwa data tersebut berhasil ditambahkan	Sesuai
TC03	Tidak menginput data-data yang dibutuhkan di <i>form</i> lalu menekan tombol tambah	Invalid	Akan muncul notifikasi bahwa data tersebut kosong dan wajib untuk diisi	Sesuai
TC04	Memilih fitur ubah data PEH	Valid	Menampilkan detail <i>input form</i> dan dapat diubah oleh admin	Sesuai
TC05	Mengubah data yang diperlukan lalu menekan tombol simpan	Valid	Akan muncul notifikasi bahwa data tersebut berhasil diperbarui	Sesuai



Gambar 2.6 Tampilan *Form* Hapus Data PEH

Berdasarkan pada Gambar 2.3 dan Gambar 2.6 *form* hapus PEH, terdapat rancangan *test case* yang diuji untuk menghapus data kegiatan.

Tabel 2.4 *Test Case Hapus* Data PEH

Test Case ID	Test Description
TC08	Memilih fitur hapus data PEH

2.4 Environment Setup

Pada tahap ini, tujuan utamanya adalah memastikan bahwa lingkungan pengujian, termasuk perangkat lunak dan perangkat keras yang digunakan, sudah siap. Tahap ini juga mencakup persiapan semua yang diperlukan untuk pengujian, seperti pengaturan lingkungan pengujian dan data uji yang dibutuhkan

2.5 Text Execution

Pada tahap ini, hasil pengujian dicatat dan setiap ketidaksesuaian dengan persyaratan ditulis sebagai "Tidak Sesuai dengan Harapan." Jika ada *bug*, tim pengembang akan diberitahu untuk memperbaikinya, dan pengujian otomatis akan diulangi. Hasil pengujian juga dilaporkan, dengan catatan *bug*.

2.6 Test Cycle Closure

Di akhir siklus pengujian, laporan hasil pengujian otomatis disusun dan data hasil pengujian dikumpulkan untuk evaluasi dan analisis. Tahap ini bertujuan untuk mengidentifikasi dan mengatasi

TC06	Menghapus data yang sudah ada lalu menekan tombol simpan	<i>Invalid</i>	Akan muncul notifikasi bahwa data tersebut kosong dan wajib untuk diisi	Sesuai
TC07	Memilih fitur hapus data PEH	<i>Valid</i>	Akan muncul notifikasi bahwa data tersebut akan dihapus	Sesuai
TC08	Mengisi kolom tanggal pada bagian kanan halaman PEH dan memilih fitur cetak	<i>Valid</i>	Form tersebut akan otomatis tercetak dalam bentuk PDF sesuai periode waktu yang ditentukan	Sesuai

Berdasarkan hasil pengujian menggunakan metode *Equivalence Partitioning* pada Tabel 3.1, dari total 8 tes yang dilakukan, sistem menunjukkan kesesuaian sebesar 100%. Ini berarti bahwa seluruh delapan tes yang dilakukan menunjukkan hasil yang sesuai dengan harapan dan persyaratan yang telah ditetapkan. Tidak ada ketidaksesuaian atau *bug* yang ditemukan selama pengujian ini. Metode *Equivalence Partitioning* digunakan untuk membagi data uji menjadi beberapa partisi yang mewakili berbagai kondisi *input* yang mungkin terjadi. Hasil ini menunjukkan bahwa sistem telah berhasil memenuhi semua kondisi *input* yang diuji, sehingga dapat disimpulkan bahwa sistem berfungsi dengan baik dalam skenario yang diuji. Dokumentasi hasil pengujian ini mencakup detail dari setiap tes yang dilakukan, hasil yang diperoleh, dan konfirmasi bahwa semua hasil sesuai dengan kriteria yang diharapkan

4. Kesimpulan

Pengujian fitur *form* PEH pada aplikasi e-Promkes dilakukan menggunakan teknik *Equivalence Partitioning*. Berdasarkan hasil pengujian yang tercantum dalam Tabel 3.1, terdapat delapan kasus uji yang diuji, dan hasil pengujian menunjukkan bahwa semua kasus uji tersebut berhasil memenuhi ekspektasi. Dari delapan kasus uji tersebut, sistem menunjukkan kesesuaian 100% dengan semua tes dinyatakan sesuai. Hal ini menandakan bahwa fitur *form* PEH pada aplikasi e-Promkes berfungsi dengan baik sesuai dengan harapan dan spesifikasi yang telah ditentukan.

Teknik *Equivalence Partitioning* berhasil menguji berbagai kondisi input, baik *valid* maupun *invalid*, serta memastikan bahwa sistem memberikan *respons* yang tepat untuk setiap kondisi tersebut.

References

- [1] A. Fahrezi, F. N. Salam, G. M. Ibrahim, R. R. Syaiful, and A. Saifudin, "Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia," *Log. J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 1, pp. 1–5, 2022, [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic>
- [2] A. H. Albaqir, A. R. Putra, and A. Saifudin, "Pengujian Aplikasi Pembelian Tiket Online menggunakan Metode Pengujian Black Box Testing Berbasis Equivalence Partitions," *JRIIN J. Ris. Inform. dan Inov.*, vol. 1, no. 4, pp. 1–9, 2023.
- [3] D. Setiawan, M. A. Fadhillah, A. Wibawa, I. Sugiarto, A. Mulyana, and I. Kusyadi, "Pengujian Black Box pada Aplikasi Perpustakaan Berbasis Web Menggunakan Teknik Equivalence Partitioning," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 2, p. 95, 2020, doi: 10.32493/jtsi.v3i2.3955.
- [4] A. R. S. M. Z. R. B. M. Z. S. A. Saifudin, "Pengujian Black Box pada Sistem Aplikasi Informasi Data Kinerja Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, no. Vol 3, No 1 (2020): Jurnal Teknologi Sistem Informasi dan Aplikasi, pp. 9–14, 2020, [Online]. Available: <http://openjournal.unpam.ac.id/index.php/JTSSI/article/view/4303/pdf>
- [5] F. I. Pratama, E. M. N. Subroto, R. M. Haira, and M. A. Yaqin, "Pengujian Black Box pada Aplikasi E-Commerce OpenCart dengan Metode Equivalence Partitioning dan Boundary Value Analysis," *J. Ilm. Inform.*, vol. 8, no. 1, pp. 54–64, 2023, doi: 10.35316/jimi.v8i1.54-64.
- [6] M. Ilham, D. Apriansyah, M. R. Nurhakiki, N. Apriani, and A. Saifudin, "Pengujian Black Box pada Aplikasi Ecommerce Berbasis Website Menggunakan Teknik Equivalence Partitions," *OKTAL J. Ilmu Komput. dan Sci.*, vol. 2, no. 7, pp. 1958–1966, 2023, [Online]. Available: <https://journal.mediapublikasi.id/index.php/oktal>
- [7] B. A. Priyaungga, D. B. Aji, M. Syahroni, N. T. S. Aji, and A. Saifudin, "Pengujian Black Box pada Aplikasi Perpustakaan Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 3, p. 150, 2020, doi: 10.32493/jtsi.v3i3.5343.
- [8] Uminingsih, M. Nur Ichsanudin, M. Yusuf, and S. Suraya, "Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan Dengan Metode Black Box Testing Bagi Pemula," *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123/storage.v1i2.270.
- [9] A. Arfan and Hendrik, "Penerapan STLC dalam Pengujian Automation Aplikasi Mobile (Studi kasus: LMS Amikom Center PT.GIT Solution)," *Jur. Inform. Univ. Islam Indones.*, vol. 3, no. 2, pp. 1–6, 2022, [Online]. Available: <https://jurnal.uui.ac.id/AUTOMATA/article/view/24127/>
- [10] M. S. A. K. Jaya, P. Gumilang, T. Wati, Y. P. Andersen, and T. Desyani, "Pengujian black box pada aplikasi sistem penunjang keputusan seleksi calon pegawai negeri sipil menggunakan teknik equivalence partitions," *J. Inform. Univ. Pamulang*, vol. 4, no. 4, pp. 131–136, 2019.