

Pengujian Aplikasi Pemesanan Tiket Kereta berbasis Website Menggunakan Metode *Black Box* dengan Teknik *Equivalence Partitioning*

Riro Bregas Trengginaz¹, Ade Yusup², Muhammad Ruhul Jihad³, Daniel Sovian Sunyoto⁴,
Yulianti Yulianti⁵

Teknik Informatika, Universitas Pamulang, Jl. Raya Puspitek No.46, Buaran, Serpong, Tangerang Selatan, Banten, Indonesia, 15310

E-mail: ¹rirobregas@gmail.com, ²adeyusuf94@gmail.com, ³muhammadruhuljihad@gmail.com, ⁴danielsoviaan1@gmail.com, ⁵yulianti@unpam.ac.id

Submitted Date: June 03rd, 2020
Revised Date: August 02nd, 2020

Reviewed Date: July 17th, 2020
Accepted Date: August 03rd, 2020

Abstract

The train ticket booking application is an important application because it involves financial transactions. If something goes wrong can be detrimental to the parties involved. The developer of the train ticket booking application must be able to provide a guarantee that the application has been made according to the requirements and has good quality. To provide a guarantee can be done by testing the application. Testing on a product intends to check if a program has been running as per its capacities or still have blunders that should be adjusted so as to have a decent quality program. A famous testing strategy and broadly utilized by analyzers to investigate the program works out in a good way or not is Black Box Testing and White Box Testing. In this examination, the product that will be surveyed utilizing Black Box Testing is a train ticket booking framework that has two structures: the login structure and the booking type of the train pass to be filled by the administrator. The application will test utilizing Black Box Testing in which the test is just planned to see whether the program is as per the capacity that the program needs without realizing the program code utilized. The sort of testing of the Black Box Testing technique shifts one of which is Equivalence Partitions utilized by the creators in this investigation. Comparability Partitions procedure is a test dependent on entering information on each structure on the framework, each info menu will be tried and assembled by work either legitimate or invalid. Incorrect test results can be found and corrected immediately. So after testing, all requirements have been met and can be guaranteed that the train ticket booking application that has been made is error free and meets all the requirements specified.

Keywords: Application Testing; Blackbox; Equivalence Partitions; Ticket Booking

Abstrak

Aplikasi pemesanan tiket kereta merupakan aplikasi yang penting karena melibatkan transaksi keuangan. Jika terjadi kesalahan dapat merugikan pihak-pihak yang terlibat. Pengembang aplikasi pemesanan tiket kereta harus dapat memberikan jaminan bahwa aplikasi yang dibuat telah sesuai persyaratan dan memiliki kualitas yang baik. Untuk memberikan jaminan dapat dilakukan dengan pengujian aplikasi. Pengujian pada suatu perangkat lunak bertujuan untuk memeriksa apakah sebuah program telah berjalan sesuai dengan fungsinya atau masih memiliki kesalahan yang harus diperbaiki agar program yang dimiliki memiliki kualitas yang baik. Teknik pengujian yang populer dan banyak digunakan oleh para penguji untuk memeriksa program berjalan baik atau tidak adalah *Black Box Testing* dan *White Box Testing*. Pada pengkajian kali ini perangkat lunak yang akan dikaji menggunakan *Black Box Testing* adalah sebuah sistem pemesanan tiket kereta yang memiliki dua form yaitu *form login* dan *form memesan tiket kereta* yang akan diisi oleh admin. Aplikasi tersebut akan melakukan pengujian menggunakan *Black Box Testing* di mana pengujian ini hanya bertujuan untuk melihat program tersebut apakah sesuai dengan fungsi yang diinginkan program tersebut tanpa mengetahui kode program yang digunakan. Jenis pengujian dari metode *Black Box Testing* beragam salah satunya adalah *Equivalence Partitions* yang dipakai oleh

penulis dalam penelitian ini. Teknik *Equivalence Partitions* merupakan pengujian berdasarkan memasukkan data pada setiap *form* yang ada pada sistem, setiap menu masukkan akan dilakukan pengujian dan dikelompokkan berdasarkan fungsinya baik itu bernilai valid maupun tidak valid. Hasil pengujian yang tidak sesuai dapat ditemukan dan langsung diperbaiki. Sehingga setelah pengujian, semua persyaratan telah terpenuhi dan dapat diberikan jaminan bahwa aplikasi pemesanan tiket kereta yang dibuat telah bebas dari kesalahan dan memenuhi semua persyaratan yang ditetapkan.

Kata Kunci: Pengujian Aplikasi; *Black Box*; Pemesanan Tiket; *Equivalence Partitions*

1 Pendahuluan

Pengujian perangkat lunak merupakan pengujian yang bertujuan pada memastikan sebuah perangkat lunak yang dibuat sesuai dengan fungsi dan kebutuhan yang sebelumnya ditentukan. Pengujian perangkat lunak memiliki fungsi yang penting pada pengembangannya untuk menemukan galat (cacat) yang disebabkan oleh perbedaan antara hasil yang diharapkan dengan hasil sebenarnya (Pratama, Ristiano, Prayogo, Nasrullah, & Saifudin, 2020). Aktifitas pengujian tersebut terdiri dari beberapa rangkaian yang berfokus pada tiap desain pada perangkat lunak. Kepuasan pelanggan bergantung pada kualitas perangkat lunak dan kualitas tersebut harus dijaga dengan beberapa alasan (Cholifah, Yulianingsih, & Sagita, 2018).

Hal penting yang perlu diperhatikan pada melakukan pengujian adalah harusnya mendapatkan kesalahan yang tidak bisa terdeteksi sebelumnya dan dapat dikatakan berhasil pengujian tersebut apabila bisa memperbaiki kesalahan tadi sehingga kualitas perangkat lunak tadi semakin lebih baik. Hal lain yang diperhatikan adalah ketika. Saat menemukan kesalahan akan berdampak pada kualitas perangkat lunak yang karenanya dalam memperbaiki kesalahan tersebut membutuhkan perbaikan dengan cepat sehingga menghemat saat pada pengujian.

Pengujian merupakan tahapan penting yang harus dilakukan untuk memberikan jaminan terhadap kualitas perangkat lunak yang dikembangkan (Muslimin, et al., 2020). Pengujian akan dilakukan dalam perangkat lunak yang dipakai sebagai Pemesanan Tiket Kereta secara online melalui website. Pemilihan tersebut karena perlu adanya sebuah sistem yang bisa memesan tiket secara *online* dan dapat diakses dengan mudah oleh pengguna.

Perangkat lunak yang akan dilakukan pengujian adalah aplikasi pemesanan tiket kereta berbasis website. Aplikasi tersebut dikembangkan untuk memudahkan pengguna yang ingin melakukan pemesanan tiket kereta tanpa harus mengantri dan tanpa harus bepergian menuju lokasi

tiket penjualan kereta berada. Kualitas perangkat lunak yang akan diuji masih belum diketahui apakah sesuai dengan fungsinya atau tidak. Karena itulah, penulis ingin melakukan pengujian lebih lanjut dengan menggunakan metode *Black Box*. Proses pengujiannya adalah mengidentifikasi masukan lalu dilakukan pengujian agar mengetahui letak kesalahannya.

Pada sisi kualitas perangkat lunak pun tidak luput dari permasalahan, seperti perangkat lunak masih bisa menyimpan data walaupun form belum terisi, tidak munculnya peringatan Ketika ada kesalahan dalam memasukkan, bahkan fungsionalitas form yang seharusnya untuk masukan angka malah bisa dimasukkan alfabet. Maka dengan dilakukannya pengujian terhadap aplikasi pemesanan tiket kereta ini bisa meminimalisasi bahkan menghilangkan bug atau error dalam aplikasi tersebut.

Metode pengujian yang akan digunakan dalam pengujian ini adalah Metode *Black Box*. Metode *Black Box* adalah metode yang ketika melakukan pengujian tidak perlu melihat dari kode pemrograman, tidak seperti *White Box Testing* yang mengutamakan kode pemrograman yang digunakan beserta dengan kelas dan fungsinya. *Black Box Testing* sendiri memiliki beberapa teknik dalam melakukan pengujian salah satu tekniknya adalah *Equivalence Partitions*. Perancangan kasus uji (test case) pada teknik *Equivalence Partitions* didasarkan pada evaluasi kelas yang menggambarkan kondisi ekuivalen masukan dari kumpulan kadaan yang valid dan tidak valid (Nugraha, Aditama, Arrofi, Ahmad, & Yulianti, 2020).

Pengujian merupakan suatu rangkaian aktivitas yang terencana dan sistematis untuk menguji atau mengevaluasi kebenaran fungsi aplikasi berdasarkan desain kasus uji (test case) yang spesifik (Yulistina, Nurmala, Supriawan, Juni, & Saifudin, 2020). Pengujian akan direncanakan dengan tahapan awal yaitu menentukan *Test Case* yang akan diuji, lalu membaginya menjadi beberapa partisi masukan dan keluaran. Hal ini dilakukan buat menerima

dokumentasi pengujian (MZ, 2016). Kemudian hasil *Test Case* akan dijadikan sebuah tabel dokumentasi dari perangkat lunak yang dikembangkan sesuai dengan fungsinya.

2 Metodologi

Pengujian perangkat lunak merupakan tahap yang harus dilalui untuk menjamin kualitas software yang dikembangkan (Pratala, Asyer, Prayudi, & Saifudin, 2020). Pengujian *software* sangat penting dilakukan karena setiap orang membuat kesalahan pada saat pembuatan *software*. Kesalahan pada masing-masing *software* akan berbeda pada masing-masing *software* (Akiladevi, Vidhupriya, & Sudha, 2018). Maka dari itu perlu dilakukan pengujian *software* untuk melakukan verifikasi dan validasi bahwa program dibuat sama dengan kebutuhan dari perusahaan. Apabila tidak sama dengan kebutuhan dari perusahaan, maka perlu dilakukan evaluasi agar dapat dilakukan perbaikan pada *software* tersebut (Ardana, 2019). Pengujian yang akan digunakan adalah *Blackbox* agar kualitas *software* lebih baik. Pengujian yang dilakukan tidak lengkap dan efektif dapat menimbulkan masalah dan menyebabkan kerugian ketika *software* digunakan (Rosalina, Rassi, Hadi, Ubaidillah, & Desyani, 2020).

Pengujian *Black Box* bertumpu pada memastikan tiap proses sudah berfungsi sesuai dengan kebutuhan yang diharapkan. Penguji dapat menartikan himpunan kondisi masukan dan menjalankan pengujian pada pengkhususan fungsi dari sistem. Sehingga pengujian merupakan suatu cara pelaksanaan program yang bertujuan menemukan kesalahan atau error kemudian memperbaikinya sehingga sistem dapat dikatakan layak untuk digunakan (Nurudin, Jayanti, Saputro, Saputra, & Yulianti, 2019). Pengujian *Black Box* tidak bisa dikatakan sebagai solusi lain dari pengujian *White Box*, begitu pula pengujian *White Box* bukan solusi alternatif dari pengujian *Black Box*, pengujian *White Box* lebih mencakup kepada kode pemrograman yang telah disusun, lalu kelas dan fungsi yang digunakan dapat diuji untuk melihat kemungkinan menemukan kesalahan pada kode program yang sedang dikembangkan.

Metode *Equivalence Partitions* merupakan metode pengujian yang menggunakan masukan pada setiap menu yang terdapat di sistem pemesanan tiket kereta api Kertajaya, beberapa menu masukan dilakukan pengujian dengan digolongkan dan dikelompokkan berdasarkan fungsinya (Hidayat & Muttaqin, 2018). Dalam penelitian ini akan dilakukan beberapa tahapan.

Tahap pertama diawali dengan menentukan kasus uji (*Test Case*) pada perangkat lunak dengan menggunakan teknik *Equivalence Partitions* melakukan inialisasi standar grade partitions masukan dan keluaran. Hal ini dilakukan agar mendapatkan dataset berupa hasil pengujian dengan menggunakan metode *Equivalence Partitions* yang telah didokumentasikan (MZ, 2016). Hasil pengujian terdapat pada tabel rancangan *Test Case* yang berfungsi menyimpulkan apakah sistem berhasil dalam melakukan pengujian tipe tersebut atau tidak. Rancangan *Test Case* berdasarkan *Equivalence Partitions*.



Gambar 1 Tampilan Form Login

Berdasarkan pada *form* yang ada di Gambar 1 terdapat beberapa rencana pengujian. Pada pengujian *Username* data akan valid jika mengisi *Username* dengan data yang sudah terdapat di database (contoh “*admin*”) dan *password* (“*admin*”), setelah form terisi dan kemudian klik login maka sistem akan menampilkan (“Selamat Datang”), tetapi bila *password* terisi dengan data yang salah maka muncul peringatan (“*Username/Password Salah!*”), dan juga bila *Username* dan *password* kosong maka muncul peringatan (“*please fill out this field*”) pada form tersebut.

Tabel 1 Rancangan *Test Case* Form Login

Id	Rincian Pengujian	Hasil yang Diinginkan
L01	Mengisi <i>Username</i> dengan “Admin” lalu <i>Password</i> diisi “Admin” terus klik Login	Web menampilkan “Selamat Datang” dan kemudian user berhasil masuk.

L02	Mengisi <i>Username</i> dengan “Admin” dan <i>Password</i> diisi “asal” lalu klik Login	Tampil peringatan “ <i>Username / Password Salah!!</i> ”
L03	Mengkosongkan <i>Username</i> dan <i>Password</i> kemudian klik Login	Tampil Peringatan “ <i>please fill out this field</i> ” disalah satu <i>form</i>

Tabel 2 Rancangan Test Case *Form* Pemesanan Tiket

Id	Rincian Pengujian	Hasil yang Diinginkan
P01	Kode tiket terisi secara otomatis dan selanjutnya Memilih nama kereta dengan pilihan “Kertajaya”, Mengisi nama pembeli dengan “Ruhul”, Keberangkatan dengan “Rangkasbitung”, Tujuan dengan “Rawa Buntu”, Tanggal berangkat dengan “12-12-2020”, Harga dengan “16000”, dan memilih kelas kereta dengan pilihan “Ekonomi” kemudian klik Tambah	Muncul pemberitahuan “Berhasil” dan data sukses masuk ke database
P02	Mengosongkan bagian nama pembeli, keberangkatan, tujuan, dan Harga dan kemudian klik Tambah.	Akan muncul pemberitahuan “ <i>fill out this field</i> ” dibagian form tersebut

3 Hasil dan Pembahasan

Pengujian yang dilakukan pada sistem pemesana tiket kereta api Kertajaya tujuannya untuk mengetahui kekurangan yang ada di dalam sistem sebelum bisa digunakan oleh admin. Hasil pengujian jika ditemukan kelemahan dalam sistem maka bisa untuk dikembangkan pada bagian yang dianggap kelemahannya itu. Pada hasil pengujian terdapat tabel *Test Case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dan sesuai dengan rencana pengujian atau tidak. Pengujian akan menggunakan dengan metode *Black Box* berbasis Teknik *Equivalence Partitions*.

Gambar 2 Tampilan Form Pemesanan Tiket Kereta Api Kertajaya

Berdasarkan form di Gambar 2 terdapat beberapa rencana pengujian. Pada rencana pengujian *form* pemesanan tiket, admin harus memilih nama kereta, mengisi nama pembeli, keberangkatan, Tujuan, Tanggal Berangkat, Harga Tiket dan memilih Kelas Kereta. Untuk kode tiket akan terisi secara otomatis sesuai sistem. Kemudian ketika klik Tambah maka sistem akan memunculkan pemberitahuan (“Berhasil”) dan data berhasil masuk ke database. Tetapi jika *form* yang harusnya terisi tetapi masih kosong maka muncul peringatan (“*fill out this field*”) di *form* yang kosong tersebut.

Berdasarkan rencana pengujian yang telah disusun, maka dilakukan pengujian. Hasil pengujian ditunjukkan pada Tabel 3.

Tabel 3 Hasil Pengujian Equivalence Partitions

Id	Rincian Pengujian	Hasil yang Diinginkan	Hasil yang Didapatkan	Keterangan
L01	Mengisi <i>Username</i> dengan "Admin" lalu <i>Password</i> diisi "Admin" terus klik Login	Web menampilkan "Selamat Datang" dan kemudian user berhasil masuk.	Web menampilkan "Selamat Datang" dan kemudian user berhasil masuk.	Sesuai
L02	Mengisi <i>Username</i> dengan "Admin" dan <i>Password</i> diisi "asal" lalu klik Login	Tampil peringatan " <i>Username / Password Salah!!</i> "	Tampil peringatan " <i>Username / Password Salah!!</i> "	Sesuai
L03	Mengosongkan <i>Username</i> dan <i>Password</i> kemudian klik Login	Tampil Peringatan " <i>please fill out this field</i> " disalah satu <i>form</i>	Tampil Peringatan " <i>please fill out this field</i> " disalah satu <i>form</i>	Sesuai
P01	Kode tiket terisi secara otomatis dan selanjutnya Memilih nama kereta dengan pilihan "Kertajaya", Mengisi nama pembeli dengan "Ruhul", Keberangkatan dengan "Rangkasbitung", Tujuan dengan "Rawa Buntu", Tanggal berangkat dengan "12-12-2020", Harga dengan "16000", dan memilih kelas kereta dengan pilihan "Ekonomi" kemudian klik Tambah	Muncul pemberitahuan "Berhasil" dan data sukses masuk ke database	Muncul pemberitahuan "Berhasil" dan data sukses masuk ke database	Sesuai
P02	Mengosongkan bagian nama pembeli, keberangkatan, tujuan, dan Harga dan kemudian klik Tambah.	Akan muncul pemberitahuan " <i>fill out this field</i> " dibagian form tersebut	Akan muncul pemberitahuan " <i>fill out this field</i> "	Sesuai

Dari pengujian yang sudah dilakukan dapat dilihat bahwa semua skenario yang dijalankan berhasil dengan baik dan telah sesuai harapan penguji. Dengan ini dipastikan bahwa aplikasi pemesanan tiket kereta api Kertajaya ini sudah berjalan dengan baik dan dapat segera digunakan.

4 Kesimpulan

Pengujian yang telah dilakukan bertujuan untuk memastikan bahwa fungsi sistem yang terdapat di aplikasi bekerja dengan baik tanpa ada masalah. Setelah dilakukan pengujian *Black Box* berbasis Teknik *Equivalence Partitions* terhadap aplikasi pemesanan tiket kereta api Kertajaya dapat disimpulkan dalam pengujian tidak ditemukan kesalahan.

5 Saran

Aplikasi pemesanan tiket kereta api Kertajaya diharapkan bisa dikembangkan dengan lebih baik lagi agar memberikan kemudahan dan kenyamanan dalam pemesanan tiket kereta. Dengan adanya pengujian perangkat lunak ini, peneliti lain dapat melakukan pengujian serupa dengan objek yang berbeda. Hasil pengujian ini tidak akan persis

sama dengan pengujian pada objek lainnya. Pengujian dalam penelitian ini juga dapat dilakukan dengan teknik-teknik lain misalnya menggunakan pendekatan *white box testing*. Selain itu juga jenis-jenis pengujian lainnya dapat pula diterapkan seperti *unit testing*, *regression testing*, *user acceptance testing* dan lain-lain.

Referensi

- Akiladevi, R., Vidhupriya, P., & Sudha, V. (2018). A Study and Analysis on Software Testing Tools. *International Journal of Pure and Applied Mathematics*, 118(18), 1783-1800.
- Ardana, I. S. (2019). Pengujian Software Menggunakan Metode Boundary Value Analysis dan Decision Table Testing. *Jurnal Teknologi Informasi*, XIV(03), 40-47.
- Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *Jurnal String*, 206-210.
- 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. *Jutis (Jurnal Teknik Informatika)*, 6(1), 25-29.

- Muslimin, D. B., Kusmanto, D., Amilia, K. F., Ariffin, M. S., Mardiana, S., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 5(1), 19-25. doi:10.32493/informatika.v5i1.3778
- MZ, M. K. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions Pada Aplikasi Sistem Informasi Sekolah. *Jurnal Mikrotik*, 2-8.
- Nugraha, B. F., Aditama, F., Arrofi, M., Ahmad, S. U., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Penghitungan Parkir Swalayan ADA Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 5(2), 146-151. doi:10.32493/informatika.v5i2.5350
- Nurudin, M., Jayanti, W., Saputro, R. D., Saputra, M. P., & Yulianti, Y. (2019). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 4(4), 143-148.
- Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. (2020). Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path. *Jurnal Informatika Universitas Pamulang*, 5(2), 111-119. doi:informatika.v5i2.4713
- Pratama, B. P., Ristiano, I. B., Prayogo, I. A., Nasrullah, & Saifudin, A. (2020). Pengujian Perangkat Lunak Sistem Informasi Penilaian Mahasiswa dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing. *Journal Of Artificial Intelligence And Innovative Applications*, 32-36.
- Rosalina, A., Rassi, A. A., Hadi, G. Y., Ubaidillah, R., & Desyani, T. (2020). Pengujian Black Box pada Sistem Informasi Penjualan HI Shoe Store Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 5, 26-29. doi:10.32493/informatika.v5i1.3964
- Yulistina, S. R., Nurmala, T., Supriawan, R. M., Juni, S. H., & Saifudin, A. (2020). Penerapan Teknik Boundary Value Analysis untuk Pengujian Aplikasi Penjualan Menggunakan Metode Black Box Testing. *Jurnal Informatika Universitas Pamulang*, 5(2), 129-135. doi:10.32493/informatika.v5i2.5366