

## Pengujian Terhadap Aplikasi Parking Management Menggunakan Metode Black-Box Berbasis Equivalence Partitions

Fahmi Ahmad Fauzi<sup>1</sup>, Gunawan Eka Putra<sup>2</sup>, Supriyanto<sup>3</sup>, Nurul Asia Saputra<sup>4</sup>, Teti Desyani

Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

E-mail: <sup>1</sup>fahmifauzi07@gmail.com, <sup>2</sup>gunawan.ninetails@gmail.com, <sup>3</sup>supriyanto2412@gmail.com, <sup>4</sup>nurulasia1997@gmail.com, dosen@unpam.ac.id

Submitted Date: March 21<sup>st</sup>, 2020

Reviewed Date: April 04<sup>th</sup>, 2020

Revised Date: April 08<sup>th</sup>, 2020

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

### Abstract

Testing is a process in implementing a program for the purpose of finding / looking for an error. A test case is said to be good is if the test to be performed has the possibility of finding an error that cannot be revealed. While testing is said to be successful if the test to be carried out successfully found an error that originally could not be found. This research was tested by applying the Black Box Testing method. There are several ways in the Black Box Testing method, namely Boundary Value Analysis, Comparison Testing, Sample Testing, Robustness Testing, Equivalence Partitioning and others. In this study using testing by Equivalence Partitioning. Equivalence Partitioning Analysis is a case / problem that will be tested by finding several errors and reducing / minimizing the number of cases that must be made. This test is performed on the Parking Management Application function to facilitate managing parking. The test results indicate the application is in accordance with needs.

Keywords: Testing; Method; Black Box; Equivalence Partitioning; Parking Management.

### Abstrak

Pengujian adalah suatu proses dalam melaksanakan suatu program untuk tujuan menemukan/mencari suatu kesalahan. Kasus pengujian dikatakan baik adalah apabila pengujian yang akan dilakukan memiliki kemungkinan dalam menemukan suatu kesalahan yang tidak dapat terungkap. Sedangkan pengujian dikatakan berhasil adalah apabila pengujian yang akan dilakukan berhasil menemukan suatu kesalahan yang awalnya tidak dapat ditemukan. Penelitian ini dilakukan pengujian dengan menerapkan metode Black Box Testing. Terdapat beberapa cara dalam metode Black Box Testing yaitu Boundary Value Analysis, Comparison Testing, Sample Testing, Robustness Testing, Equivalence Partitioning dan lain-lain. Pada penelitian ini menggunakan pengujian dengan cara Equivalence Partitioning. Equivalence Partitioning Analysis merupakan suatu kasus/permasalahan yang akan dilakukan pengujian dengan ditemukannya beberapa kesalahan dan mengurangi/meminimalisir atas jumlah kasus yang harus dibuat. Pengujian ini dilakukan pada fungsi Aplikasi Parking Management guna memudahkan mengelola parkir. Hasil pengujian menunjukkan aplikasi sudah sesuai dengan kebutuhan.

Kata Kunci: Pengujian; Metode; Black Box; Equivalence Partitioning; Manajemen Parkir.

### 1 Pendahuluan

Perkembangan teknologi diikuti meningkatnya penggunaan dan kompleksitas software, sehingga jaminan kualitas software menjadi kritis dan tantangan yang sulit (Saifudin, Heryadi, & Lukas, 2019), tetapi dalam pembuatan perangkat lunak (software) pastinya kita sering menghadapi "bug" atau kesalahan "error" pada proses-proses tertentu. Maka untuk menghindari banyaknya bug maka diperlukan suatu pengujian di

perangkat lunak (software), sebelum perangkat lunak (software) yang telah kita buat diberikan ke pelanggan atau selama perangkat lunak (software) masih akan terus dikembangkan untuk mengurangi terjadinya kesalahan yang merugikan (Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019).

Pada dasarnya pentingnya pengujian perangkat lunak (software) mengacu pada kualitas perangkat lunak (software) tersebut. Kepuasan pelanggan tergantung pada kualitas perangkat

lunak dan kualitas sejumlah perangkat lunak perlu dijaga dengan beberapa alasan (Cholifah, Yulianingsih, & Sagita, 2018).

Aplikasi Parking Management merupakan platform rekam data transaksi parkir berbasis aplikasi komputer untuk menghasilkan laporan realtime. Aplikasi ini diperuntukkan bagi pengelola parkir yang ingin meningkatkan layanan parkir lebih mudah dan transparan. Aplikasi ini perlu diuji untuk menemukan kesalahan yang tidak disengaja. Maka dilakukan langkah pemeriksaan untuk memastikan bahwa data tersebut telah sesuai dengan yang ditetapkan dan bertujuan untuk memastikan bahwa data akan dimasukkan ke dalam basis data itu telah diketahui dan dapat dijelaskan kebenaran datanya (MZ, 2016).

Pengujian pada penelitian ini diusulkan menguji menggunakan pengujian dengan metode Black Box. Black Box merupakan suatu teknik pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak (software), penguji dapat mengumpulkan kondisi masukan dan melakukan pengujian pada spesifikasi fungsional program. Equivalence partitioning adalah metode black box testing yang membagi domain masukan dari suatu program ke dalam kelas-kelas data, di mana test cases dapat diturunkan.

Equivalence partitioning berdasarkan pada premis masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang akan diperlakukan sama (ekuivalen) oleh komponen tersebut. Dapat juga diasumsikan bahwa masukan yang sama akan menghasilkan respon yang sama pula. Di mana rencana pengujian dengan menganalisa kebutuhan dan spesifikasi dalam pemasukan data, melakukan pemilihan input, melakukan pemilihan output. Semua tahapan akan dibahas 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 merupakan suatu proses pelaksanaan program yang bertujuan menemukan kesalahan dan memperbaikinya sehingga sistem dikatakan layak untuk digunakan (Nurudin, Jayanti, Saputro, Saputra, & Yulianti, 2019). Pengujian merupakan salah satu tahapan yang harus dilalui untuk menghasilkan sebuah perangkat

lunak atau sistem yang berkualitas. Melalui pengujian, kami dapat mengetahui kekurangan dan kesalahan yang terdapat di dalam sebuah sistem sehingga dapat segera memperbaikinya. Tujuannya adalah menyesuaikan fungsi-fungsi dengan spesifikasi yang diinginkan.

Pengujian *software* sangat diperlukan untuk memastikan aplikasi/*software* yang sedang dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Software yang mengandung kesalahan dapat menyebabkan *business process* tidak didukung oleh software yang dihasilkan, dan perlu perbaikan atau pengerjaan ulang jika jumlahnya banyak (Saifudin & Wahono, 2015). Pada aktivitas pengujian perangkat lunak harus memperhatikan berbagai hal termasuk mengidentifikasi keterhubungan antara sekumpulan masukan dengan hasil yang diinginkan (MZ, 2016).

Pengujian yang dilakukan tidak lengkap dan efektif dapat menimbulkan masalah dan menyebabkan kerugian ketika software digunakan (Rosalina, Rassi, Hadi, Ubaidillah, & Desyani, 2020). Penguji *software* harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan dapat dideteksi sejak awal dan dikoreksi secepatnya (Shi Mingtao, 2010).

Pengujian *Black Box* berusaha untuk menemukan klasifikasi kesalahan dalam beberapa kategori, yaitu fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses database eksternal, kesalahan performa, kesalahan dan terminasi (Agarwal, Tayal, & Gupta, 2010). Dalam pengujian *Black Box Testing* digunakan alat untuk pengumpulan data yang disebut dengan *user acceptance test*, dokumen ini terdiri deskripsi indikator dari prosedur – prosedur pengujian fungsionalitas dari perangkat lunak. Dengan metode *Black Box Testing* ini, pengembang aplikasi parking management dapat mengelola parkir yang ingin meningkatkan layanan parkir lebih mudah dan transparan.

Metode *Equivalence Partitions* merupakan pengujian berdasarkan masukan setiap menu yang terdapat pada sistem seleksi pemenang tender, setiap menu masukan dilakukan pengujian melalui klasifikasi dan pengelompokan berdasarkan fungsinya (Hidayat & Muttaqin, 2018).

Dalam penelitian ini akan dilakukan beberapa tahapan. Pada tahapan pertama diawali dengan menentukan *Test Case* perangkat lunak yang akan diuji dengan metode *Equivalence*

Partitions kemudian menginisialisasi *standar grade partition* masukan dan keluaran. Hal ini dilakukan untuk mendapatkan dataset berupa dokumentasi pengujian dengan metode

*Equivalence Partitions* dan nilai tingkat efektifitas metode *Equivalence Partitions*.

Tabel 1. Rancangan Test Case Form Parkir Keluar testing form *Data Parkir Masuk*.

Test Id	Skenario pengujian	Hasil yang diharapkan
PM01	Mengisi No Polisi dengan Huruf dan Angka "B6087CWY" serta memilih jenis kendaraan lalu mengklik "Simpan".	Sistem akan secara otomatis menyimpan data kendaraan.
PM02	Mengisi No Polisi dengan Huruf saja serta memilih jenis kendaraan lalu mengklik "Simpan".	Sistem tidak akan menyimpan data dan akan menampilkan alert "Nomor Polisi Harus diisi dengan benar!".
PM03	Mengisi No Polisi dengan Angka "12345678" serta memilih jenis kendaraan lalu mengklik "Simpan".	Sistem tidak akan menyimpan data dan akan menampilkan alert "Nomor Polisi Harus diisi dengan benar!".

Pada pengujian terdapat tabel rancangan *Test Case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian tipe tersebut atau tidak. Berikut terdapat penjelasan dari beberapa tabel Rancangan *Test Case* berdasarkan *Equivalence Partition*.

Dari hasil testing, terdapat data tabel dari rancangan Test Case yang berguna menyimpulkan apakah aplikasi berhasil dalam pengujian tipe yang ditetapkan atau tidak.

Pada form Gambar 1 ada beberapa rencana testing. Rencana Testing *No.Pol* akan valid jika *No.Pol* diisi dengan memasukkan angka dan huruf "ABCD1234" yang terdiri dari 8 digit dan tidak

boleh kosong.

Sebaliknya, data tidak valid jika memasukkan abjad saja "ABCDEFGH" dan juga angka saja "12345678".

Pada form Gambar 2 ada beberapa rencana testing. Rencana Testing *No.Pol* akan valid jika *No.Pol* diisi dengan memasukkan angka dan huruf "ABCD1234" yang terdiri dari 8 digit dan tidak boleh kosong.

Sebaliknya, data tidak valid jika memasukkan abjad saja "ABCDEFGH" dan juga angka saja "12345678".

No. Tiket	No. Pol	Jenis	Tanggal Masuk	Jam Masuk
36	A1B2C3D4	Motor	2020-03-13	21:43:04
35	12345678	Mobil	2020-03-13	21:42:27
34	ABCDEFGH	Motor	2020-03-13	21:42:01

Gambar 1 Form Parkir Keluar

No. Tiket	No. Pol	Jenis	Tgl Keluar	Jam Keluar	Biaya
-----------	---------	-------	------------	------------	-------

Gambar 2 Form Parkir Masuk

Tabel 2. Rancangan Test Case Form Parkir Keluar testing form *Data Parkir Keluar*

Test Id	Skenario pengujian	Hasil yang diharapkan
PK01	Memasukkan No Polisi Huruf dan Angka "B6087CWY" serta memilih jenis kendaraan lalu langsung mengklik tombol "cari".	Sistem akan secara otomatis menampilkan data masuk kendaraan serta total biaya parkir.

PK02	Memasukkan No Polisi Huruf "ABCDEFGH" serta memilih jenis kendaraan lalu langsung mengklik tombol "cari".	Sistem akan menampilkan alert "Nomor Polisi Salah Data Tidak Ditemukan!"
PK03	Memasukkan No Polisi Angka "12345678" serta memilih jenis kendaraan lalu langsung mengklik tombol "cari".	Sistem akan menampilkan alert "Nomor Polisi Salah Data Tidak Ditemukan!"

### 3 Hasil dan Pembahasan

berikut.

Berdasarkan rencana pengujian yang telah disusun dapat dilakukan pengujian sebagai

Tabel 3 Hasil Pengujian

Test Id	Skenario pengujian	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
PM01	Mengisi No Polisi dengan Huruf dan Angka "B6087CWY" serta memilih jenis kendaraan lalu mengklik "Simpan".	Sistem akan secara otomatis menyimpan data kendaraan.	Data Kendraan tersimpan dalam database dan Terlihat nomor kendaraan	Valid
PM02	Mengisi No Polisi dengan Huruf saja serta memilih jenis kendaraan lalu mengklik "Simpan".	Sistem tidak akan menyimpan data dan akan menampilkan alert "Nomor Polisi Harus diisi dengan benar!".	Data Kendraan tidak tersimpan dalam database dan tidak Terlihat nomor kendaraan	Valid
PM03	Mengisi No Polisi dengan Angka "12345678" serta memilih jenis kendaraan lalu mengklik "Simpan".	Sistem tidak akan menyimpan data dan akan menampilkan alert "Nomor Polisi Harus diisi dengan benar!".	Data Kendraan tidak tersimpan dalam database dan tidak Terlihat nomor kendaraan	Tidak Valid
PK01	Meng input No Polisi Huruf dan Angka "B6087CWY" serta memilih jenis kendaraan lalu langsung mengklik tombol "cari".	Sistem akan menampilkan alert "Nomor Polisi Harus diisi dengan benar!".	Sistem secara otomatis menampilkan data masuk kendaraan serta total biaya parkir.	Tidak Valid
PK02	Meng input No Polisi Huruf "ABCDEFGH" serta memilih jenis kendaraan lalu langsung mengklik tombol "cari".	Sistem akan menampilkan alert "Nomor Polisi Harus diisi dengan benar!".	Sistem menampilkan alert "Nomor Polisi Salah Data Tidak Ditemukan!"	Tidak Valid
PK03	Meng input No Polisi Angka "12345678" serta memilih jenis kendaraan lalu langsung mengklik tombol "cari".	Sistem akan menampilkan alert "Nomor Polisi Harus diisi dengan benar!".	Sistem menampilkan alert "Nomor Polisi Salah Data Tidak Ditemukan!"	Tidak Valid

### 4 Kesimpulan

Pengujian aplikasi dengan menggunakan metode *Black Box* memberikan dokumentasi hasil pengujian yang memberikan informasi kesesuaian aplikasi yang diuji dengan spesifikasi yang telah ditentukan. Maka berdasarkan pengujian aplikasi Parking Management, dapat diambil kesimpulan bahwa pengujian dengan metode *Black Box* berbasis *Equivalence Partitions* dapat membantu proses pembuatan *case* pengujian, uji kualitas dan menemukan kesalahan yang tidak disengaja. Dalam pengujian aplikasi Parking Management ditemukan kesalahan yang mana ketika menginput

nomor kendaraan dengan kondisi huruf semua, angka semua, huruf dan angka berselingan serta angka dan huruf berselingan, aplikasi masih dapat menyimpan data kendaraan tersebut, tidak sesuai dengan ekspektasi. Pengujian ini menunjukkan bahwa terdapat celah *error* pada suatu form aplikasi Parking Management. Untuk menjaga kestabilan fungsi dari perangkat lunak ini maka diperlukan prosedur penggunaan dan batasan pengguna yang jelas. Sehingga dapat memberikan gambaran bagaimana serangkaian skenario diujikan untuk memperoleh nilai yang valid, sehingga hasil masukkan sesuai dengan yang

diinginkan.

## 5 Saran

Berdasarkan proses pengujian kualitas aplikasi Parking Management, penulis memberikan saran pada penelitian selanjutnya agar melakukan mengkaji dengan lebih banyak sumber maupun referensi terkait agar hasil penelitian memberikan hasil yang lebih baik lagi dan informasi yang lebih lengkap lagi. Diharapkan pada penelitian selanjutnya untuk lebih melakukan persiapan yang matang dalam aplikasi Parking Management serta menggunakan metode pengujian lain agar hasilnya sesuai dengan yang diharapkan.

## Referensi

- Agarwal, B. Tayal, S.P. dan Gupta, M., 2010, Software Engineering & Testing, Sudbury, Johanes and Bartlett Publishers, Massachusetts.
- Cholifah, W., Yulianingsih, & Sagita, S. (2018). Pengujian Black Box Testing Pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *Jurnal String*, 3(2), 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. *Jurnal Teknik Informatika UNIS*, 6(1), 25-29.
- Hanifah, U., Alit, R., & Sugiarto. (2016). Penggunaan Metode Black Box pada Pengujian Sistem Informasi Surat Keluar Masuk. *SCAN*, 11(2), 33-40.
- Jaya, M. S., Gumilang, P., Wati, T., Andersen, Y. P., & Desyani, T. (2019). Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 131-136.
- MZ, M. K. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions Pada Aplikasi Sistem Informasi Sekolah. *Jurnal Mikrotik*, 2-8.
- 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*, 4(4), 125-130.
- 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.
- 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
- Saifudin, A., & Yulianti, Y. (2020). Dimensional Reduction on Cross Project Defect Prediction. *Journal of Physics: Conference Series*. 1477, p. 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, pp. 1-8. Bandung: IOP Publishing. doi:10.1088/1757-899X/662/6/062012
- Shi, Mingtao, 2010, Software Functional Testing from the Perspective of BusinessPractice Computer and Information Science, www.ccsenet.org/cis