

Pengujian *Black Box* pada Sistem Informasi Penjualan HI Shoe Store Menggunakan Teknik *Equivalence Partitions*

Adella Rosalina¹, A A Gede Rai Rassi², Gozali Yusuf Hadi³, Rizki Ubaidillah⁴, Teti Desyani⁵

Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia
e-mail: ¹adellarosalina97@gmail.com, ²liveflat36@gmail.com, ³gozaliyusuf21@gmail.com,
⁴rizkiubayd@gmail.com, ⁵dosen00839@unpam.ac.id

Submitted Date: February 09th, 2020
Revised Date: March 29th, 2020

Reviewed Date: February 19th, 2020
Accepted Date: March 30th, 2020

Abstract

Testing software that is incomplete and ineffective can cause various problems that cause losses, especially for users, this test is carried out to ensure that if it does not match what is expected then the system fails. The sales information system at the HI Shoe Store is then carried out using the Black Box test based on Equivalence Partitions, so it will be known weaknesses in the information system after testing. The system test results show that testing can improve quality and guarantee error free.

Keywords: Systems, Information, Sales, Testing, Black Box

Abstrak

Pengujian perangkat lunak yang tidak lengkap dan tidak efektif dapat mengakibatkan berbagai masalah yang menyebabkan kerugian terutama bagi pengguna, pengujian ini dilakukan untuk memastikan bahwa jika tidak sesuai dengan apa yang diharapkan maka sistem gagal. Sistem informasi penjualan pada HI Shoe Store kemudian dilakukan menggunakan pengujian *Black Box* berbasis *Equivalence Partitions*, maka akan diketahui kelemahan pada sistem informasi setelah dilakukan pengujian. Hasil pengujian sistem menunjukkan bahwa dengan pengujian dapat meningkatkan kualitas dan menjamin software yang dikembangkan bebas dari kesalahan.

Kata Kunci : Sistem, Informasi, Penjualan, Pengujian, *Black Box*

1 Pendahuluan

Pengujian perangkat lunak adalah cara untuk mendapatkan sebuah informasi mengenai kualitas dari perangkat lunak yang diuji (Sulistyanto & SN, 2014), pengujian perangkat lunak dilakukan untuk mendeteksi adanya kesalahan, yang menyebabkan kegagalan perangkat lunak (Irawan, 2017). Proses pengujian untuk mengidentifikasi bagian dari perangkat lunak yang rawan mengalami kegagalan (Hanifah, Ronggo, & Sugiarto, 2016), sebuah perangkat lunak dinyatakan gagal, jika perangkat lunak tersebut tidak memenuhi spesifikasi (Komarudin, 2016).

Tujuan pengujian perangkat lunak dilakukan untuk mencari kesalahan perangkat lunak (Wibisono & Baskoro, 2016). Pengujian perangkat lunak juga bertujuan untuk memperoleh produk yang berkualitas yang memberikan produktivitas tinggi. Pengujian

software diperlukan untuk menjamin bahwa yang dikembangkan dapat mendukung *business process* sesuai dengan persyaratan (*requirement*). Kesalahan pada software dapat menyebabkan *business process* tidak didukung oleh software yang dikembangkan, dan perlu perbaikan atau pengerjaan ulang jika jumlahnya banyak (Saifudin & Wahono, 2015). Dalam proses pengujian perangkat lunak, untuk setiap kasus yang akan diuji harus memiliki identitas dan mempunyai keterhubungan antara sekumpulan masukan dengan hasil yang diinginkan (Komarudin, 2016).

Pengujian yang dilakukan sebelumnya tidak lengkap dan tidak efektif sehingga mengakibatkan berbagai masalah ketika perangkat lunak tersebut digunakan. Masalah yang timbul dalam sebuah perangkat lunak dapat menyebabkan kerugian terutama bagi pemilik, oleh karena itu dilakukan kembali pengujian

untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak (Syaban & Bunyamin, 2015).

Pengujian yang diusulkan harus dirancang dengan baik agar dapat menemukan kesalahan secara sistematis dan dapat diperbaiki dalam waktu dan usaha yang minimal. Pengujian *Black box* berbasis *Equivalence Partitions* berusaha untuk menemukan kesalahan fungsi-fungsi yang hilang atau salah, kesalahan desain antarmuka atau tampilan, kesalahan dalam struktur data atau akses menuju *database*, dan kesalahan performa (Febiharsa, Sudana, & Hudallah, 2018). Dalam proses pengujian, setiap kasus yang akan diuji harus memiliki identitas dan mempunyai keterhubungan antara sekumpulan masukan dengan hasil yang diinginkan.

Pengujian dilakukan untuk memastikan bahwa masing-masing fungsi beroperasi dengan sepenuhnya, sehingga sistem dikatakan layak

untuk digunakan (Nurudin, Jayanti, Saputro, Saputra, & Yulianti, 2019). Pengujian juga untuk mencari kesalahan pada tiap fungsi yang berfokus pada masukan sistem data, tampilan, pemakaian memori dan kecepatan eksekusi data sehingga jika masukan data tidak sesuai dengan apa yang diharapkan maka sistem gagal (Mustaqbal, Firdaus, & Rahmadi, 2015). Berdasarkan pengujian yang dilakukan maka akan diketahui kelemahan pada sistem informasi setelah dilakukan pengujian menggunakan metode *Black Box* berbasis *Equivalence Partitioning* dan bagaimana untuk mengetahui hasil yang dianggap *valid*. Sedangkan untuk tujuan dari pengujian ini adalah mengetahui kelemahan dari sistem agar data yang dihasilkan sesuai dengan data yang dimasukkan setelah data dieksekusi dan menghindari kekurangan dan kesalahan pada aplikasi sebelum digunakan oleh user.

Tabel 1 Rancangan Test Case

No	Skenario Pengujian	Hasil yang diharapkan
1.	Mengkosongkan <i>E-mail</i> dan <i>Password</i> , lalu menekan tombol " <i>Login</i> "	Sistem akan menampilkan pemberitahuan "Harap diisi bidang ini"
2.	Mengisi <i>e-mail</i> dengan tidak menyertakan '@' lalu menekan tombol " <i>Login</i> "	Sistem akan menampilkan pemberitahuan "Sertakan '@' pada alamat e-mail"
3.	Mengisi <i>E-mail</i> : contohuser@gmail.com dan mengkosongkan <i>Password</i> , lalu menekan tombol " <i>Login</i> "	Sistem akan menampilkan pemberitahuan "Harap diisi bidang ini"
4.	Mengkosongkan <i>E-mail</i> dan mengisi <i>Password</i> : userpass lalu menekan tombol " <i>Login</i> "	Sistem akan menampilkan pemberitahuan "Harap diisi bidang ini"
5.	Masukkan <i>e-mail</i> : testuser@gmail.com, dan <i>Password</i> : userpass, lalu menekan tombol " <i>Login</i> "	Sistem akan menolak dan menampilkan pemberitahuan "E-mail belum terdaftar"
6.	Masukkan <i>e-mail</i> iniuser@gmail.com dengan benar dan data <i>Password</i> : 123, lalu menekan tombol " <i>Login</i> "	Sistem akan menolak dan menampilkan pemberitahuan "Opsss.. Password yang anda masukkan salah"
7.	Masukkan <i>e-mail</i> iniuser@gmail.com dengan benar dan data <i>Password</i> : passuser, lalu menekan tombol " <i>Login</i> "	Sistem akan menerima dan menampilkan pemberitahuan "Login Sukses" Kemudian tampil halaman beranda

2 Metodologi

Pengujian sistem ini bertujuan untuk melihat apakah sistem yang telah dibuat sudah layak untuk dipergunakan dan sesuai dengan tujuan awal pembuatan. Sehingga sangat perlu untuk melakukan pengujian untuk mengurangi terjadinya kesalahan yang merugikan tersebut

(Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019). Dalam tahap pengujian *Equivalence Partitions* yang dilakukan adalah :

1. Membuat rancangan *test case* berdasarkan fungsi yang ada dalam pengujian perangkat lunak.

2. Membuat batasan pengujian *Equivalence Partitions*.
3. Membuat model pengujian dari skenario pengujian dan hasil yang diharapkan.
4. Melakukan pengujian berdasarkan model yang dirancang.

Pada pengujian ini digunakan Teknik *Equivalence Partitions* yaitu masukan dikelompokkan ke dalam kelompok nilai yang valid dan tidak valid. Setiap kelompok diambil satu sample untuk data masukan pada pengujian. Rancangan uji kasus yang dibuat pada pengujian ini ditunjukkan pada Tabel 1.

3 Hasil dan Pembahasan

Pengujian dilakukan pada sistem informasi penjualan dengan tujuan untuk mengetahui kekurangan yang ada pada sistem sebelum sistem digunakan oleh pengguna. Hasil dari pengujian jika ditemukan kelemahan pada bagian sistem maka pengguna bisa mengembangkan pada bagian sistem tersebut yang dianggap lemah. Pada hasil pengujian terdapat tabel *test case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian atau tidak. Pengujian akan dilakukan dengan metode *Black Box* berbasis *Equivalence Partitioning*. Hasil pengujian ditunjukkan pada Tabel 2.

Tabel 2 Hasil Pengujian *Equivalence Partitioning* pada *form login* pembeli

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Status
1.	Mengkosongkan <i>E-mail</i> dan <i>Password</i> , lalu menekan tombol " <i>Login</i> "	Sistem akan menampilkan pemberitahuan "Harap diisi bidang ini"	Sistem menampilkan pemberitahuan "Harap diisi bidang ini"	Berhasil
2.	Mengisi <i>e-mail</i> dengan tidak menyertakan '@' lalu menekan tombol " <i>Login</i> "	Sistem akan menampilkan pemberitahuan "Sertakan '@' pada alamat e-mail"	Sistem menampilkan pemberitahuan "Sertakan '@' pada alamat e-mail"	Berhasil
3.	Mengisi <i>E-mail</i> : contohuser@gmail.com dan mengkosongkan <i>Password</i> , lalu menekan tombol " <i>Login</i> "	Sistem akan menampilkan pemberitahuan "Harap diisi bidang ini"	Sistem menampilkan pemberitahuan "Harap diisi bidang ini"	Berhasil
4.	Mengkosongkan <i>E-mail</i> dan mengisi <i>Password</i> : userpass lalu menekan tombol " <i>Login</i> "	Sistem akan menampilkan pemberitahuan "Harap diisi bidang ini"	Sistem menampilkan pemberitahuan "Harap diisi bidang ini"	Berhasil
5.	Masukkan <i>e-mail</i> : testuser@gmail.com, dan <i>Password</i> : userpass, lalu menekan tombol " <i>Login</i> "	Sistem akan menolak dan menampilkan pemberitahuan "E-mail belum terdaftar"	Sistem menolak dan menampilkan pemberitahuan "E-mail belum terdaftar"	Berhasil
6.	Masukkan <i>e-mail</i> iniuser@gmail.com dengan benar dan data <i>Password</i> : 123, lalu menekan tombol " <i>Login</i> "	Sistem akan menolak dan menampilkan pemberitahuan "Opsss.. Password yang anda masukkan salah"	Sistem menolak dan menampilkan pemberitahuan "Opsss.. Password yang anda masukkan salah"	Berhasil
7.	Masukkan <i>e-mail</i> iniuser@gmail.com dengan benar dan data <i>Password</i> : passuser, lalu menekan tombol " <i>Login</i> "	Sistem akan menerima dan menampilkan pemberitahuan "Login Sukses" Kemudian tampil halaman beranda	Sistem menerima dan menampilkan pemberitahuan "Login Sukses" Kemudian tampil halaman beranda	Berhasil

4 Kesimpulan

Pengujian pada sistem ini dilakukan untuk memastikan masing-masing fungsi beroperasi

dengan sepenuhnya dan mendapatkan hasil yang diharapkan serta mencari kesalahan dalam sistem tersebut. Setelah dilakukan pengujian

Black Box berbasis *Equivalence Partitions* terhadap sistem informasi penjualan pada *Hi Shoe Store* dapat disimpulkan dalam pengujian ini tidak ditemukan kesalahan.

5 Saran

Pengujian harus dirancang dengan baik agar dapat menemukan kesalahan secara sistematis dan dapat diperbaiki dalam waktu dan usaha yang minimal. Bagi pembaca semoga dapat menambah ilmu dan wawasan tentang metode *Black Box testing* dalam aplikasi berbasis web.

Referensi

- Febiharsa, D., Sudana, I. M., & Hudallah, N. (2018). Uji Fungsionalitas (Blackbox Testing) Sistem Informasi Lembaga Sertifikasi Profesi (SILSP) Batik dengan AppPerfect Web Test dan Uji Pengguna. *Joined Journal (Journal of Informatics Education)*, 1(2), 117-126.
- Hanifah, Ronggo, A., & Sugiarto. (2016). Penggunaan Metode Black Box pada Pengujian Sistem Informasi Surat Keluar Masuk. *SCAN - Jurnal Teknologi Informasi dan Komunikasi*, 11(2), 33-40.
- Irawan, Y. (2017). Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus dengan Metode Whitebox Testing. *Sentra Penelitian Engineering dan Edukasi*, 1.
- Komarudin, M. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence. *MIKROTIK: Jurnal Manajemen Informatika*, 6(1), 1-18.
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), 31-36. doi:10.33197/jitter.vol1.iss3.2015.62
- 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. (2019). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 4(4), 143-148.
- Saifudin, A., & Wahono, R. S. (2015). Penerapan Teknik Ensemble untuk Menangani Ketidakseimbangan Kelas pada Prediksi Cacat Software. *Journal of Software Engineering*, 1(1), 28-37.
- Sulistiyanto, H., & SN, A. (2014). Urgensi Pengujian pada Kemajemukan Perangkat Lunak dalam Multi Perspektif. *Komuniti: Jurnal Komunikasi dan Teknologi Informasi*, 6(1), 65-74. doi:10.23917/komuniti.v6i1.2944
- Syaban, R. M., & Bunyamin, H. (2015). Pengembangan sistem informasi pengelolaan surat masuk dan keluar berbasis web di dinas sosial tenaga kerja dan transmigrasi kabupaten garut menggunakan framework php. *Jurnal Algoritma*, 12(2), 301-311. doi:10.33364/algoritma/v.12-2
- Wibisono, W., & Baskoro, F. (2016). Pengujian Perangkat Lunak dengan Menggunakan Model Behavior UML. *JUTI (Jurnal Ilmiah Teknologi Informasi)*, 1(1), 43-50. doi:10.12962/j24068535.v1i1.a95