

Pengujian *black box* pada Website dengan Metode *Robustness Testing* (Studi kasus : Eiger Adventure)

Nabil Fahlevi Abdi¹, S.R. Candra Nursari²

Program Studi Teknik Informatika^{1,2}

Fakultas Teknik Universitas Pancasila Jakarta^{1,2}

Nfahleviabdi8@gmail.com¹, sri.rezeki.candra.n@univpancasila.ac.id²

Abstrak— Pengujian perangkat lunak adalah proses atau serangkaian proses yang tujuannya adalah untuk memastikan bahwa sistem dapat melakukan apa yang seharusnya diharapkan. Jika sistem tidak diuji, maka sistem tersebut tidak dapat diketahui kualitasnya sehingga ketika terjadi kesalahan akan merugikan sistem tersebut. Dengan melakukan pengujian memungkinkan penguji untuk menemukan letak kecacatan pada sistem yang bisa diketahui sejak awal sehingga dapat segera dilakukan perbaikan. Pengujian memiliki beberapa jenis, pada penelitian ini yang digunakan adalah pengujian *black box* dengan menggunakan metode *Robustness Testing*, salah satu jenis pengujian yang bertujuan untuk mengevaluasi bagaimana sebuah sistem atau aplikasi mampu menangani masukan atau kondisi yang tidak diharapkan atau tidak sesuai dengan spesifikasi. Berdasarkan hasil uji dengan teknik *Robustness Testing* menunjukkan bahwa website dapat menangani input yang tidak valid tersebut dan tetap berfungsi dengan baik, maka presentase keberhasilan dari pengujian ini adalah 95%.

Kata Kunci—Pengujian; Black Box; Robustness Testing

I. PENDAHULUAN

Website e-commerce dapat diartikan sebagai transaksi jual beli yang dilakukan secara elektronik melalui media internet. Melalui website e-commerce, konsumen dapat melakukan pencarian barang atau produk yang dibutuhkan menjadi lebih mudah. Untuk memberikan kenyamanan penggunaan website kepada konsumen diperlukan sebuah pengujian untuk dapat menemukan kesalahan ataupun kekurangan pada sistem sehingga dapat mendeteksi kesalahan sejak awal dan kemudian dapat diperbaiki secepatnya.

Salah satu contoh website e-commerce yaitu pada PT Eigerindo Multi Produk Industri, website e-commerce ini menjual produk-produk untuk memenuhi berbagai kebutuhan perlengkapan dan peralatan bagi gaya hidup para pengguna di alam terbuka.

Pengujian adalah suatu proses pelaksanaan suatu program dengan tujuan menemukan suatu kesalahan. Suatu kasus test yang baik adalah apabila test tersebut mempunyai kemungkinan menemukan sebuah kesalahan yang tidak terungkap. Suatu test yang sukses adalah bila test tersebut membongkar suatu kesalahan yang awalnya tidak ditemukan. Tujuan utama dari pengujian adalah untuk

mendesain test yang secara sistematis membongkar jenis kesalahan dengan usaha dan waktu minimum [8].

Dengan dilakukannya pengujian, penguji dengan sadar bahwa kualitas perangkat lunak dapat dilihat dari hasil pengujian yang dilakukan. Semua dilakukan untuk mengukur dan mengantisipasi kesalahan yang bisa saja terlewatkan pada pengujian yang sudah dilewati. Pengujian dilakukan untuk memastikan bahwa masing-masing fungsi beroperasi dengan sepenuhnya, sehingga sistem dikatakan layak untuk digunakan [2].

Terdapat berbagai jenis pengujian salah satunya ialah Black Box. Pengujian jenis ini memiliki beberapa teknik, diantaranya Equivalence Partitioning, Boundary Value Analysis, Robustness Testing, Behavior Testing, dan Cause-Effect Relationship Testing [7]. Berdasarkan penjelasan sebelumnya, diterapkan salah satu teknik yaitu *Robustness Testing* untuk menguji website e-commerce dari PT Eigerindo Multi Produk Industri. Cara yang dapat dilakukan untuk pengujian *Robustness Testing* adalah dengan memasukkan nilai-nilai yang diluar dari kebutuhan yang sudah ditentukan.

II. LANDASAN TEORI

A. Pengertian Pengujian

Pengujian adalah suatu proses yang dilakukan untuk melihat apakah sistem berfungsi sesuai harapan dan mencari kesalahan atau error yang bisa muncul pada perangkat lunak tersebut. Pengujian ini juga memperlihatkan secara garis besar mengenai kualitas perangkat lunak secara terstruktur yang berguna dalam kegiatan bisnis untuk mengetahui tingkat manajemen risiko pada penggunaannya. Tujuan dilakukannya pengujian ini untuk menemukan error, failure, bug pada sistem yang belum terdeteksi (kesalahan atau cacat lainnya). Pengujian ini sangat diperlukan, gunanya adalah memastikan perangkat lunak yang sedang dibangun dapat berfungsi sesuai dengan apa yang direncanakan. Pengembang dan tim penguji harus menyisihkan waktu tersendiri guna melakukan pengujian perangkat lunak yang sedang dikembangkan supaya kesalahan dan kekurangan dapat diketahui lalu diperbaiki secepatnya. Ini merupakan bagian penting dari gambaran baik atau buruknya suatu perangkat lunak serta menggambarkan bagian yang tetap ada pada roda kehidupan para pengembang. [8]

B. Pengertian Pengujian Black Box

Pengujian Black Box bertumpu pada pengkhususan fungsi dari perangkat. Penguji dapat mengartikan himpunan kondisi masukan dan menjalankan pengujian pada pengkhususan fungsi dari perangkat lunak. Pengujian Black Box tidak bias dikatakan sebagai solusi jalur lain dari pengujian White Box melainkan sebagai pelengkap untuk menguji fungsi yang tidak didapat dari pengujian White Box, begitupula pengujian White Box bukanlah solusi alternatif dari pengujian Black Box, pengujian White Box lebih mencakup kepada kode program yang telah dibangun, lalu kelas dan fungsi yang dibuat dapat diuji untuk melihat kemungkinan menemukan kesalahan pada kode program yang sedang dikembangkan [1].

C. Pengertian Robustness Testing

Robustness Testing adalah pengujian untuk mengetahui sejauh mana suatu sistem dapat beroperasi dengan benar ketika diberi data atau masukan yang di luar batas sistem atau dengan kondisi lingkungan yang penuh tekanan. Karena pengujian ini perlu dipicu dengan skenario terburuk bahkan yang dapat merusak sistem, sehingga dapat mendeteksi jenis kesalahan pada sistem dan itulah hal yang penting dalam pengujian ini [9].

Tujuan dari robustness testing adalah untuk mengetahui seberapa tahan sebuah sistem terhadap gangguan atau kegagalan. Hal ini bertujuan untuk menjamin bahwa sistem dapat beroperasi dengan baik walaupun terjadi gangguan atau kegagalan. Berikut tahapan-tahapan dalam *robustness testing*, yaitu :

1. Identifikasi input yang mungkin menyebabkan gangguan atau kegagalan pada sistem.
2. Buat skenario pengujian yang menggunakan input-input tersebut.
3. Lakukan pengujian dengan menggunakan skenario yang telah dibuat.
4. Evaluasi hasil pengujian dan catat setiap gangguan atau kegagalan yang terjadi.
5. Perbaiki sistem sesuai dengan masalah yang teridentifikasi pada tahap sebelumnya.
6. Ulangi tahap-tahap di atas sampai sistem dapat beroperasi dengan baik meskipun terjadi gangguan atau kegagalan.

III. METODOLOGI PENELITIAN

Robustness testing adalah salah satu jenis pengujian yang bertujuan untuk mengevaluasi bagaimana sebuah sistem atau website mampu menangani masukan atau kondisi yang tidak diharapkan atau tidak sesuai dengan spesifikasi.

Pada website e-commerce ini, kami melakukan pengujian dengan melakukan uji sebagai berikut:

1. Memasukkan input yang tidak valid seperti karakter simbol atau huruf kapital pada form login, untuk mengecek apakah website tersebut dapat menangani

input yang tidak valid tersebut dan tetap berfungsi dengan baik.

2. Mencoba membeli produk dengan jumlah yang melebihi stok yang tersedia, untuk mengecek apakah website tersebut dapat menangani situasi ini dengan benar dan memberikan pesan error yang tepat kepada pengguna.
3. Mencoba memasukkan alamat pengiriman yang tidak valid atau tidak lengkap, untuk mengecek apakah website tersebut dapat menangani situasi ini dengan benar dan memberikan pesan error yang tepat kepada pengguna.
4. Mencoba mengakses halaman atau fitur tertentu tanpa login terlebih dahulu, untuk mengecek apakah website tersebut dapat menangani situasi ini dengan benar dan memberikan pesan error yang tepat kepada pengguna.

Setelah dilakukan pengujian, dibuatlah analisis pada berbagai uji yang sudah dilakukan berupa tabel test case dengan membandingkan hasil keluaran dari pengujian dengan tabel test case.

IV. HASIL & PEMBAHASAN

Tahap pengujian dilakukan untuk mengidentifikasi input yang mungkin menyebabkan gangguan atau kegagalan pada sistem di dalam situs web Eiger Adventure. Kondisi *input* sendiri didefinisikan sebagai kasus uji untuk menghasilkan *output* yang dapat dievaluasi.

A. Pengujian Pada Modul Login dan Register

Tabel 1 - Tabel Pengujian Modul Login

No.	Deskripsi pengujian	Hasil yang diharapkan	Hasil
1	Pencobaan login dengan username atau password yang salah	Sistem dapat menampilkan pesan kesalahan yang sesuai jika username atau password yang dimasukkan salah	Berhasil
2	Pencobaan login tanpa mengisi username atau password	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika username atau password tidak diisi	Berhasil
3	Pencobaan login dengan username yang sudah terdaftar	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika username yang dimasukkan sudah terdaftar	Berhasil

Tabel 2 - Tabel Pengujian Modul Register

No.	Deskripsi pengujian	Hasil yang diharapkan	Hasil
1	Pencobaan registrasi tanpa mengisi data lengkap	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika ada data yang tidak diisi	Berhasil
2	Pencobaan registrasi dengan email yang tidak valid	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika data yang dimasukkan tidak valid	Berhasil
3	Pencobaan registrasi dengan password yang lemah	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika password yang dimasukkan tidak aman	Berhasil

B. Pengujian Pada Modul Halaman Produk

Tabel 3 - Tabel Pengujian Modul Halaman Produk

No.	Deskripsi pengujian	Hasil yang diharapkan	Hasil
1	Pencobaan mencari produk yang tidak ada dalam daftar	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika produk yang dicari tidak ada dalam daftar	Berhasil
2	Pencobaan menambahkan produk yang stoknya habis ke keranjang	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika stok produk yang ditambahkan sudah habis	Berhasil
3	Pencobaan pemuatan halaman yang lama	Aplikasi dapat memuat halaman dengan cepat dan tanpa error	Berhasil

C. Pengujian Pada Modul Keranjang Barang

Tabel 4 - Tabel Pengujian Modul Keranjang

No.	Deskripsi pengujian	Hasil yang diharapkan	Hasil
1	Pencobaan menambahkan jumlah barang yang melebihi stok yang ada	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika jumlah barang yang ditambahkan melebihi stok yang ada	Berhasil
2	Pencobaan menambahkan barang dengan jumlah yang banyak secara cepat	Aplikasi dapat memuat dan menambahkan barang dengan cepat dan tanpa error	Tidak berhasil

D. Pengujian Pada Modul Checkout

Tabel 5 - Tabel Pengujian Modul Checkout

No.	Deskripsi pengujian	Hasil yang diharapkan	Hasil
1	Pencobaan checkout tanpa mengisi data pembeli	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika ada data pembeli yang tidak diisi	Berhasil
2	Pencobaan checkout dengan data pembeli yang tidak valid	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika data pembeli yang dimasukkan tidak valid	Berhasil
3	Pencobaan checkout tanpa mengisi metode pembayaran	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika metode pembayaran tidak diisi	Berhasil
4	Percobaan checkout dengan mengisi voucher yang tidak valid	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika voucher yang digunakan tidak sesuai	Berhasil

E. Pengujian Pada Modul Pembayaran

Tabel 6 - Tabel Pengujian Modul Pembayaran

No.	Deskripsi pengujian	Hasil yang diharapkan	Hasil
1	Pencobaan pembayaran tanpa login	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika belum melakukan login	Berhasil
2	Pencobaan membayar tanpa mengisi field nomor kartu kredit	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika field nomor kartu kredit tidak diisi	Berhasil
3	Pencobaan pembayaran dengan nomor kartu kredit yang tidak valid	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika nomor kartu kredit yang dimasukkan tidak valid	Berhasil
4	Pencobaan membayar tanpa mengisi field tanggal masa berlaku	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika field tanggal kadaluarsa tidak diisi	Berhasil
5	Pencobaan pembayaran dengan CVV yang tidak valid	Aplikasi dapat menampilkan pesan kesalahan yang sesuai jika kode CVV yang dimasukkan tidak valid	Berhasil

V. PENUTUP

Berdasarkan hasil pengujian dengan menggunakan metode *Robustness Testing* pada website Eiger Adventure dapat disimpulkan dari 20 pengujian terdapat 19 pengujian yang berhasil dalam menangani kesalahan input atau gangguan pada sistem. Website ini dapat menangani kesalahan input yang tidak valid tersebut dan masih tetap berfungsi dengan baik, maka presentase keberhasilan pengujian ini adalah 95%.

DAFTAR PUSTAKA

[1] A. S. Anggana, A. Shiddiq, A. A. Samui, C. Kodri, F. Ramadhan, and A. Saifudin, "Pengujian Black Box pada Aplikasi Admin PT. World Trans Berbasis Web Menggunakan Metode Equivalence Partitioning," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 1, p. 95, Mar. 2020, doi: 10.32493/informatika.v5i1.3866.

[2] M. Nurudin, W. Jayanti, R. D. Saputro, M. P. Saputra, and Y. Yulianti, "Pengujian Black Box pada Aplikasi Penjualan Berbasis Web

Menggunakan Teknik Boundary Value Analysis," *Jurnal Informatika Universitas Pamulang*, vol. 4, no. 4, p. 143, Dec. 2019, doi: 10.32493/informatika.v4i4.3841.

[3] A. Ijudin and Aries Saifudin, "Pengujian Black Box pada Aplikasi Berita Online dengan Menggunakan Metode Boundary Value Analysis," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 1, pp. 8–12, 2020, Accessed: Nov. 29, 2022. [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika/article/view/3717/pdf>

[4] A. Rosalina, A. Gede, Gozali Yusuf Hadi, Rizki Ubaidillah, and Teti Desyani, "Pengujian Black Box pada Sistem Informasi Penjualan HI Shoe Store Menggunakan Teknik Equivalence Partitions," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 1, pp. 26–29, 2020, Accessed: Nov. 29, 2022. [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika/article/view/3964/pdf>

[5] S. Anwar, Pandu Gumilang, T. Wati, Yohanes Philipus Andersen, and Teti Desyani, "Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions," *Jurnal Informatika Universitas Pamulang*, vol. 4, no. 4, pp. 131–136, 2019, Accessed: Nov. 29, 2022. [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika/article/view/3834/pdf>

[6] Dwi Bahar Muslimin, Dwi Kusmanto, Karina Femi Amilia, Muhammad Syamsul Ariffin, Siti Mardiana, and Yulianti Yulianti, "Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 1, pp. 19–25, 2020, Accessed: Nov. 29, 2022. [Online]. Available: <http://openjournal.unpam.ac.id/index.php/informatika/article/view/3778/pdf>

[7] Nadya Safitri and Rully Pramudita, "Pengujian Black Box Menggunakan Metode Cause Effect Relationship Testing," *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS : Journal of Information System*, vol. 3, no. 1, pp. 101–110, 2018, Accessed: Nov. 29, 2022. [Online]. Available: <https://ejournal-binainsani.ac.id/index.php/ISBI/article/view/1050>

[8] M. Sidi Mustaqbal, Roeri Fajri Firdaus, and Hendra Rahmadi, "PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (STUDI KASUS : APLIKASI PREDIKSI KELULUSAN SMNPTN)," *Jurnal Ilmiah Teknologi Infomasi Terapan*, vol. 1, no. 3, 2015, doi: 10.33197/jitter.vol1.iss3.2015.62.

[9] Zoltan Micskei, H. Madeira, A. Avritzer, and N. Antunes, "Robustness Testing Techniques and Tools," *ResearchGate*, Nov. 02, 2012. https://www.researchgate.net/publication/278692960_Robustness_Testing_Techniques_and_Tools

[10] A. Shah, D. Sundmark, Birgitta Lindström, and S. F. Andler, "Robustness Testing of Embedded Software Systems: An Industrial Interview Study," *ResearchGate*, 2016. https://www.researchgate.net/publication/299375286_Robustness_Testing_of_Embedded_Software_Systems_An_Industrial_Interview_Study