

Penerapan Teknik *Equivalence Partitioning* untuk Pengujian *Black box*: Studi Kasus Opencart

Gibran Kahlil^{1*}, Afni Puspita Zahra², Muhammad Bintang Nugraha³ dan Ionia Veritawati^{4*}

^{1,2,3,4}Program Studi Teknik Informatika, Fakultas Teknik, Universitas Pancasila, Jakarta

Abstrak. Platform *e-commerce* seperti OpenCart memegang peran krusial dalam bisnis modern, namun sangat rentan terhadap kesalahan fungsional pada modul *input* pengguna. Keandalan fungsional, terutama pada keranjang belanja, berdampak langsung pada kepercayaan pengguna dan kesuksesan transaksi. Penelitian ini bertujuan untuk menerapkan metode pengujian *Black box* dengan teknik *Equivalence Partitioning* (EP) untuk mengevaluasi fungsionalitas modul keranjang belanja OpenCart. Metodologi penelitian berfokus pada perancangan dan eksekusi kasus uji berdasarkan partisi *valid* dan *invalid* pada *field input Quantity* dan *Coupon Code*. Berdasarkan tujuh kasus uji yang dieksekusi penelitian ini menghasilkan tingkat keberhasilan fungsional sebesar 57,14%, dengan mengidentifikasi tiga bug fungsional yang menunjukkan kegagalan terkait *input*, di mana sistem gagal memberikan pesan *error* yang memadai untuk data *invalid*. Sementara itu, 4 kasus uji lainnya menunjukkan keberhasilan fungsional sistem yang berjalan sesuai harapan. Disimpulkan bahwa teknik *Equivalence Partitioning* terbukti efektif dan efisien dalam menemukan kegagalan fungsional pada modul keranjang belanja OpenCart.

Kata kunci— *E-commerce*, Pengujian *Black box*, *Equivalence Partitioning*, OpenCart.

1. PENDAHULUAN

Perkembangan teknologi saat ini mengakibatkan pergeseran yang signifikan di bidang bisnis, hal ini membuat aplikasi *e-commerce* menjadi suatu komponen penting bagi dunia bisnis untuk pedagang maupun pembeli. Salah satu platform *e-commerce* yang populer digunakan adalah OpenCart, sebuah sistem *open-source* berlisensi GNU GPL (*General Public License*) yang diminati karena dapat digunakan secara gratis dan dimodifikasi secara fleksibel oleh pedagang sesuai kebutuhan [1].

Pengujian perangkat lunak sangat penting dalam pengembangan sistem informasi perangkat dan aplikasi *e-commerce*, hal tersebut dilakukan untuk menjamin kualitas aplikasi dalam *e-commerce* karena keandalan fungsional platform berdampak langsung pada kepercayaan dan kepuasan pengguna. Pengujian dilakukan untuk mengevaluasi kemampuan sistem, memastikan bahwa fungsi berjalan sesuai dengan rancangan, dan menemukan kesalahan yang mungkin muncul. Tanpa pengujian, kesalahan mungkin tidak terdeteksi dan akan merugikan di kemudian hari [2], [3], [4].

Black box Testing adalah salah satu metode pengujian yang populer digunakan untuk melakukan verifikasi fungsional. Metode ini berfokus pada pengujian kebutuhan fungsional perangkat lunak. Penguji hanya berfokus pada *input* dan *output* yang dihasilkan tanpa melihat atau mengetahui struktur kode internal program, sehingga pengujian dilakukan murni dari perspektif pengguna [5]. *Equivalence Partitioning* (EP) adalah salah satu teknik turunan *Black box* yang bekerja dengan cara membagi domain *input* program ke dalam beberapa kelas data (partisi), yang kemudian dikategorikan menjadi partisi data yang *valid* dan partisi data yang tidak *valid* [2], [6], [7]. Kemampuannya untuk mengurangi jumlah skenario uji yang perlu dijalankan, sehingga proses pengujian menjadi lebih efisien namun tetap memiliki cakupan pengujian yang luas adalah keunggulan utama dari teknik ini [7], [8].

Platform *e-commerce* seperti OpenCart sangat bergantung pada seberapa mudah pengguna memasukan data. Keranjang belanja pada *e-commerce* merupakan modul fungsionalitas transaksi yang paling awal dan

* Corresponding author: ionia.veritawati@univpancasila.ac.id

memegang peran krusial. Modul ini tidak hanya dapat menampilkan produk yang dipilih tetapi juga menangani berbagai *input* penting, seperti “Menambahkan kode diskon” dan “Menambahkan kode *voucher* hadiah” yang merupakan fitur yang tersedia dalam alur *checkout*, dan seringkali ditampilkan pada halaman keranjang belanja [1].

Penelitian ini terinspirasi dari penelitian sebelumnya [1] yang telah melakukan pengujian pada fitur *checkout* OpenCart menggunakan kombinasi dua metode yaitu, metode *Equivalence Partitioning* dan *Boundary Value Analysis*. Penelitian ini menunjukkan bahwa metode *Equivalence Partitioning* hanya memiliki tingkat kesesuaian sebesar 57,14%. Hasil penelitian tersebut menunjukkan bahwa sistem OpenCart masih memiliki kelemahan yang signifikan dalam menangani kombinasi *input* dan *output*. Berdasarkan temuan tersebut, penelitian ini akan berfokus pada penerapan teknik *Equivalence Partitioning* untuk pengujian *Black box* pada fungsionalitas modul keranjang belanja OpenCart. Pengujian diharapkan dapat mengidentifikasi kesalahan *validasi input* sedini mungkin dalam alur belanja pengguna, sebelum pengguna melanjutkan ke proses *checkout* akhir.

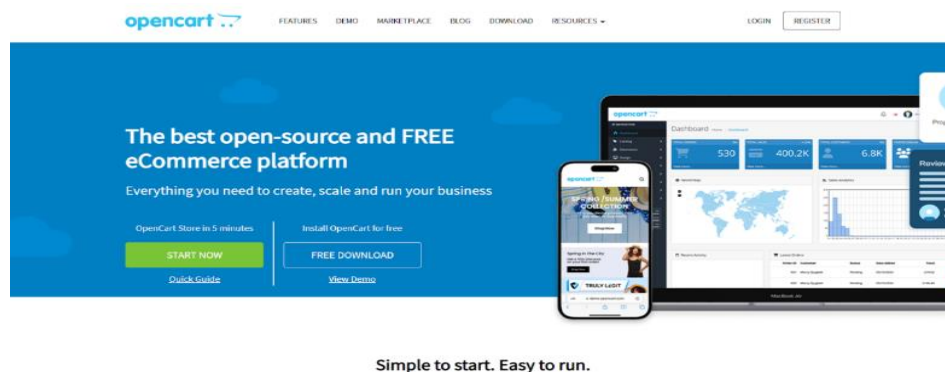
2. METODE PENELITIAN

a. Desain Penelitian

Teknik pengujian yang Digunakan adalah *Black box Testing*. Pemilihan metodologi ini didasarkan pada premis bahwa penelitian tidak mengakomodasi atau secara sengaja mengesampingkan tinjauan terhadap struktur kode sumber internal. *Black box Testing* bersifat eksklusif berorientasi pada validasi eksternal, memfokuskan telah pada kesesuaian fungsi dan relasi kausal antara masukan dan keluaran sistem, guna memastikan pemenuhan spesifikasi kebutuhan fungsional.

b. Objek Penelitian

Objek penelitian adalah sistem OpenCart (demo website publik versi 4.0.2.3), yaitu platform *e-commerce* berbasis web bersifat *open-source*. Sistem ini dipilih karena memiliki fitur lengkap untuk proses transaksi daring. Sesuai dengan rumusan masalah, pengujian difokuskan pada modul keranjang belanja (*Shopping Cart*), karena merupakan bagian fundamental dari pengalaman pengguna dalam sistem *e-commerce*.



Gambar 1. Tampilan Halaman Utama OpenCart

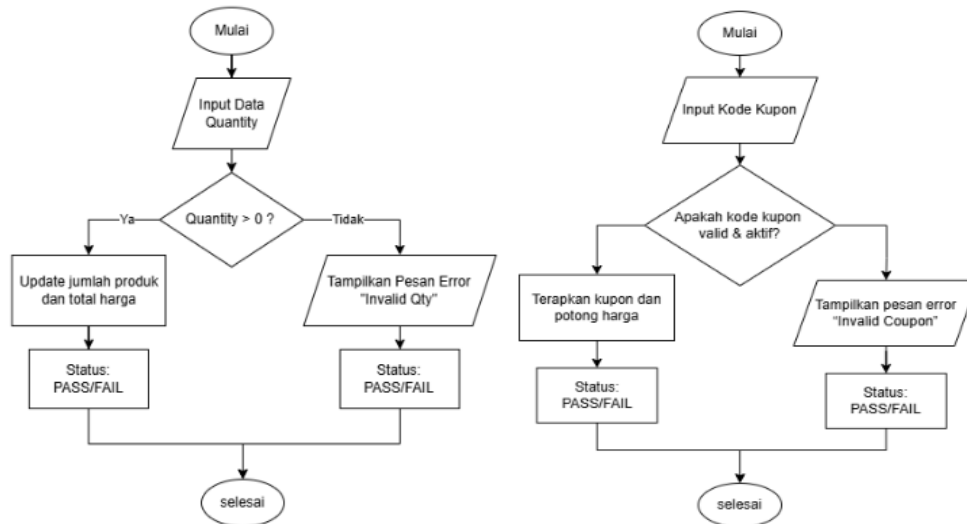
c. Metode Pengujian

Metode pengujian yang digunakan adalah *Black box Testing* dengan teknik *Equivalence Partitioning*. Teknik ini membagi data uji ke dalam beberapa kelas ekivalensi (partisi) berdasarkan kesamaan hasil keluaran yang diharapkan. Setiap kelas mewakili kelompok *input valid* dan tidak *valid*, yang memungkinkan proses pengujian dilakukan secara efisien tanpa menguji seluruh kemungkinan data. Dalam konteks penelitian ini, teknik *Equivalence Partitioning* diterapkan pada Fitur Keranjang Belanja (*Shopping Cart*) untuk menguji fungsionalitas pembaruan jumlah produk dan validasi kode kupon.

d. Tahapan Pengujian

- Identifikasi Modul yang Diuji: Modul yang diuji adalah Modul Keranjang Belanja, dengan fokus pada *field input Quantity* (Jumlah) dan *Coupon Code* (Kode Kupon).

- Perancangan Partisi dan Kasus Uji (*Test Case Design*): Merancang partisi *valid* dan *invalid* untuk *field input* yang diuji, yang kemudian diturunkan menjadi skenario kasus uji seperti terlihat pada Gambar berikut:



Gambar 2. Perancangan Kasus Uji *Quantity* dan Diskon

- Eksekusi Pengujian: Menjalankan kasus uji pada sistem OpenCart.
- Analisis Hasil Pengujian: Hasil pengujian dibandingkan antara keluaran aktual dan hasil yang diharapkan. Jika hasil sesuai, fungsionalitas dinyatakan berhasil (*PASS*); jika tidak, dinyatakan gagal (*FAIL*) dan dianalisis sebagai bug.

e. *Desain Test Case Kanjang Belanja*

Desain Test case keranjang belanja dapat dilihat pada Tabel 1.

Tabel 2. Desain Test Case Keranjang Belanja

ID Test Case	Skenario Pengujian	Partisi yang Diuji	Input Data Uji	Hasil yang Diharapkan
TC-CART-01	Update kuantitas dengan data <i>valid</i>	VP 1 (<i>Quantity</i>)	5	Kuantitas produk berubah menjadi 5, total harga di- <i>Update</i> .
TC-CART-02	Update kuantitas dengan angka nol	IP 1 (<i>Quantity</i>)	0	Muncul pesan error " <i>Quantity must be 1 or more</i> " atau item terhapus dari keranjang.
TC-CART-03	Update kuantitas dengan angka negatif	IP 2 (<i>Quantity</i>)	-1	Muncul pesan error " <i>Quantity must be a positive number</i> ".
TC-CART-04	Update kuantitas dengan teks	IP 4 (<i>Quantity</i>)	dua	Muncul pesan error " <i>Invalid input. Please enter a number</i> ".
TC-CART-05	Menerapkan kode kupon <i>valid</i>	VP 1 (<i>Coupon</i>)	DISKON10 (Kode kupon <i>valid</i> yang telah disiapkan)	Kupon berhasil diterapkan. Muncul baris " <i>Coupon</i> " dan total harga berkurang.

ID Test Case	Skenario Pengujian	Partisi yang Diuji	Input Data Uji	Hasil yang Diharapkan
TC-CART-06	Menerapkan kode kupon tidak <i>valid</i>	IP 1 (<i>Coupon</i>)	SALAHKODE	Muncul pesan <i>error</i> "Warning: Coupon is invalid, expired or reached its usage limit!"
TC-CART-07	Menerapkan kupon kosong	IP 3 (<i>Coupon</i>)	(Kosong)	Muncul pesan <i>error</i> "Warning: Please enter a Coupon Code!"

f. Alat dan Lingkungan Pengujian

Penelitian ini dilakukan dengan menggunakan satu unit laptop yang memiliki spesifikasi perangkat keras prosesor Intel i9, memori RAM 32 GB, dan media penyimpanan 1TB SSD. Perangkat lunak yang digunakan untuk mendukung penelitian mencakup sistem operasi Windows 11 dan peramban Google Chrome versi 118. Objek penelitian adalah platform OpenCart Demo (Versi 4.0.2.3) yang diakses secara publik melalui jaringan internet.

3. HASIL DAN DISKUSI

Modul Keranjang Belanja adalah fitur krusial dalam alur transaksi *e-commerce*. Pengujian pada modul ini difokuskan pada dua fungsionalitas *input* utama: pembaruan jumlah barang (*Quantity*) dan penerapan *Coupon Code*.

a. Perancangan Partisi (*Equivalence Partitioning*)

Untuk merancang kasus uji yang efisien, *field input* pada modul keranjang belanja dibagi menjadi partisi *valid* dan *invalid* sebagai berikut:

- *Field Input: Quantity* (Jumlah Produk)
 - Aturan Bisnis: Kuantitas harus berupa angka bulat lebih besar dari 0.
 - 1 (*Valid*): Angka bulat positif (Data Uji: 5)
 - IP 1 (*Invalid*): Angka nol (Data Uji: 0)
 - IP 2 (*Invalid*): Angka negatif (Data Uji: -1)
 - IP 3 (*Invalid*): Angka desimal/pecahan (Data Uji: 1.5)
 - IP 4 (*Invalid*): Teks / String (Data Uji: dua)
 - IP 5 (*Invalid*): Simbol / Karakter Spesial (Data Uji: !@#)
 - IP 6 (*Invalid*): Kosong (*Input* dihapus lalu di-*Update*)
- *Field Input: Coupon Code* (Kode Kupon)
 - Aturan Bisnis: Kode harus terdaftar dalam sistem dan *valid*.
 - VP 1 (*Valid*): Kode kupon yang *valid* dan aktif (Data Uji: DISKON10)
 - IP 1 (*Invalid*): Kode kupon yang formatnya salah/tidak ada (Data Uji: SALAHKODE)
 - IP 2 (*Invalid*): Kode kupon yang *valid* namun sudah kadaluarsa.
 - IP 3 (*Invalid*): Kosong (*Input* dibiarkan kosong lalu klik "*Apply*")

b. Tabel Kasus Uji Modul Keranjang Belanja

Berdasarkan perancangan partisi di atas, Tabel 2 berikut adalah penjabaran dan hasil kasus uji yang dieksekusi.

Tabel 2. Hasil Test Case Keranjang Belanja

ID Test Case	Skenario Pengujian	Partisi yang Diuji	Input Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC-CART-01	<i>Update</i> kuantitas dengan data <i>valid</i>	VP 1 (<i>Quantity</i>)	5	Kuantitas produk berubah menjadi 5, total harga di- <i>Update</i> .	Kuantitas produk berubah menjadi 5, total harga di- <i>Update</i> .	<i>PASS</i>

ID Test Case	Skenario Pengujian	Partisi yang Diuji	Input Data Uji	Hasil yang Diharapkan	Hasil Aktual	Status
TC-CART-02	Update kuantitas dengan angka nol	IP 1 (Quantity)	0	Muncul pesan error " <i>Quantity must be 1 or more</i> " atau item terhapus dari keranjang.	Halaman me-refresh, kuantitas item kembali menjadi 1. Tidak ada pesan error.	FAIL
TC-CART-03	Update kuantitas dengan angka negatif	IP 2 (Quantity)	-1	Muncul pesan error " <i>Quantity must be a positive number</i> ".	Halaman me-refresh, kuantitas item kembali menjadi 1. Tidak ada pesan error.	FAIL
TC-CART-04	Update kuantitas dengan teks	IP 4 (Quantity)	dua	Muncul pesan error " <i>Invalid input. Please enter a number</i> ".	Halaman me-refresh, kuantitas item kembali menjadi 1. Tidak ada pesan error.	FAIL
TC-CART-05	Menerapkan kode kupon valid	VP 1 (Coupon)	DISKON10 (Kode kupon valid yang telah disiapkan)	Kupon berhasil diterapkan. Muncul baris " <i>Coupon</i> " dan total harga berkurang.	Kupon berhasil diterapkan. Total harga berkurang.	PASS
TC-CART-06	Menerapkan kode kupon tidak valid	IP 1 (Coupon)	SALAHKODE	Muncul pesan error " <i>Warning: Coupon is invalid, expired or reached its usage limit!</i> "	Muncul pesan error " <i>Warning: Coupon is invalid...</i> "	PASS
TC-CART-07	Menerapkan kupon kosong	IP 3 (Coupon)	(Kosong)	Muncul pesan error " <i>Warning: Please enter a Coupon Code!</i> "	Muncul pesan error " <i>Warning: Coupon is invalid...</i> "	PASS

c. Diskusi temuan

Dari tujuh kasus uji yang dieksekusi pada Modul Keranjang Belanja, empat kasus uji berstatus *PASS* dan tiga kasus uji berstatus *FAIL*. Temuan ini menjawab rumusan masalah kedua dan ketiga, yang menunjukkan bahwa teknik *Equivalence Partitioning* efektif dalam mengidentifikasi bug fungsional pada validasi input. Berikut merupakan analisis dari kasus uji ini.

- TC-CART-02, TC-CART-03, dan TC-CART-04 (*FAIL*): Ditemukan kelemahan fungsionalitas yang konsisten pada validasi field *Quantity*. Ketika pengguna memasukkan data dari partisi *invalid* (nol, negatif, atau teks), sistem tidak memberikan *feedback* atau pesan *error* yang jelas. Sistem hanya me-refresh halaman dan mengembalikan kuantitas ke nilai 1.

- Dampak Bug: Perilaku sistem ini membingungkan pengguna (*bad usability*). Pengguna tidak tahu mengapa *input* mereka ditolak. Selain itu, pada kasus TC-CART-02 (*input* '0'), pengguna mungkin bermaksud menghapus item, namun sistem gagal memproses ekspektasi tersebut.
- Rekomendasi: Sistem seharusnya mengimplementasikan validasi *front-end* atau *back-end* yang menampilkan pesan *error* spesifik, seperti "Kuantitas harus berupa angka positif" atau "Masukkan minimal 1".
- Pengujian pada *field Coupon Code* (TC-CART-05, 06, 07) menunjukkan fungsionalitas yang diharapkan dan berhasil lolos pengujian (status *PASS*), di mana sistem memberikan *feedback error* yang jelas saat *input* tidak *valid*.

4. KESIMPULAN

Penelitian ini telah menerapkan metode pengujian *Black box* dengan teknik *Equivalence Partitioning* pada Modul Keranjang Belanja platform *e-commerce* OpenCart. Berdasarkan analisis sebelumnya, rancangan partisi yang dibuat telah berhasil digunakan untuk membuat tujuh kasus uji fungsional. Hasil dari eksekusi tujuh kasus uji tersebut menunjukkan empat kasus uji berstatus *PASS* dan tiga kasus uji berstatus *FAIL*. Telah diidentifikasi tiga bug fungsional, yang seluruhnya terkait dengan kelemahan validasi pada *field Quantity*. Sistem ditemukan gagal memberikan *feedback error* yang memadai untuk *input invalid* (nol, negatif, dan teks), yang berpotensi membingungkan pengguna.

Dari temuan tersebut, dapat disimpulkan bahwa teknik *Equivalence Partitioning* terbukti efektif dan efisien untuk menemukan kegagalan fungsional pada modul keranjang belanja OpenCart. Saran untuk penelitian selanjutnya agar pengujian dapat dilanjutkan dengan teknik *Boundary Value Analysis* untuk melengkapi *Equivalence Partitioning*, serta memperluas cakupan pengujian ke modul krusial lainnya seperti *checkout*.

DAFTAR PUSTAKA

- [1] 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*," *Jurnal Ilmiah Informatika*, vol. 8, no. 1, pp. 54–64, Jun. 2023.
- [2] A. Krismadi, A. F. Lestari, A. Pitriyah, I. W. P. A. Mardangga, M. Astuti, and A. Saifudin, "Pengujian *Black box* berbasis *Equivalence Partitions* pada Aplikasi Seleksi Promosi Kenaikan Jabatan," vol. 2, no. 4, pp. 155–161, Oct. 2019.
- [3] M. Sholeh, I. Gisfas, Cahiman, and M. A. Fauzi, "*Black box Testing* on ukmbantul.com Page with *Boundary Value Analysis* and *Equivalence Partitioning* Methods," *J Phys Conf Ser*, 2021.
- [4] P. Huriati, H. Azmi, Y. Wati, D. Meidelfi, and T. Lestari, "*Black box Testing* on the online quiz application using the *Equivalence Partitions* method," *International Journal of Advanced Science Computing and Engineering*, vol. 2, no. 2, pp. 51–56, Aug. 2020.
- [5] F. K. Kartono *et al.*, "Pengujian *Black box Testing* Pada Sistem Website Osha Snack: Pendekatan Teknik *Boundary Value Analysis*," *JURNAL KRIDATAMA SAINS DAN TEKNOLOGI*, vol. 6, no. 2, Dec. 2024.
- [6] R. Pramudita, "Pengujian *Black box* pada Aplikasi Ecampus Menggunakan Metode *Equivalence Partitioning*," *Informatics for Educators and Professionals*, vol. 4, no. 2, pp. 193–202, Jun. 2020.
- [7] I. Maulidya *et al.*, "Pengembangan Sistem Pencatatan Pesanan Berbasis Website dengan Pendekatan *Equivalence Partitioning* pada Tahap Pengujian *Black box*," *BINA INSANI ICT JOURNAL*, vol. 11, no. 2, pp. 154–166, Dec. 2024.
- [8] Y. Irawan, S. Muzid, N. Susanti, and R. R. Setiawan, "System Testing using *Black box Testing Equivalence Partitioning* (Case Study at Garbage Bank Management Information System on Karya Sentosa)," Oct. 2018.