

# PERBANDINGAN PORTABILITY, REUSABILITY, DAN INTEROPERABILITY PADA E-COMMERCE BERBASIS OPEN SOURCE

Rifqi Imaduddin Irfan, Aji Saka Nugraha, Rini Angrainingsih

**Abstract**—This paper will focus on comparison of portability, reusability, and interoperability values on open source-based e-commerce. The aim of comparing these three values is to find out which e-commerce has the best portability, reusability, and interoperability values. E-commerce software that will be tested are Magento 2.0, OpenCart 2.3.0.2, and WooCommerce 3.07. It has been chosen because this software are included in “The Top 10 Open Source e-Commerce Platforms” by CMS Critic website. Portability and reusability assessment is based on modularity, self-descriptive, and simplicity sub factor. The result obtained which e-commerce has the best value of the three factors tested is OpenCart with the best scores in the factor of portability and interoperability.

**Index Terms**— E-commerce, Interoperability, Portability, Reusability

## I. PENDAHULUAN

Banyak terjadi kegagalan dalam pembangunan sistem *software*. Salah satunya adalah ketidakmampuan pengembang (*developer*) dalam mengatasi permasalahan pada *software* karena salah dalam merancang kebutuhan. Untuk itu, perlu dibutuhkan rancangan kebutuhan yang komprehensif yang meliputi semua atribut *software* [1].

*Open source* bisa jadi solusi karena *source code* aplikasi tersedia secara gratis dan siapapun bisa mengunduh, memperbaiki dan memodifikasinya [2]. Dengan begitu, pengembang mudah melakukan pemeliharaan hingga pelatihan dan sosialisasi penggunaan *software* kepada pengguna dan pelanggannya.

10 besar *e-commerce* berbasis *open source* yang unggul dalam popularitas dan utilitasnya adalah *Magento*, *PrestaShop*, *OpenCart*, *osCommerce*, *Zen Cart*, *Spree Commerce*, *WooCommerce*, *Jigoshop*, *VirtueMart*, dan *Drupal Commerce* [3]. Dari sepuluh *e-commerce* tersebut, dipilih tiga secara acak untuk dilakukan pengujian, yaitu *Magento*, *OpenCart*, dan *WooCommerce*.

Secara umum tujuan pengujian *software* adalah memastikan sistem *software* bekerja semestinya ketika

digunakan oleh pengguna dan pelanggan yang ditarget pengembang *software* [4]. Pengujian nilai *portability*, *reusability*, dan *interoperability* dilakukan untuk mengetahui seberapa mudah mengadaptasi sebuah *software* digunakan diluar penggunaan aslinya [5]. Cara untuk menilai faktor-faktor tersebut adalah menggunakan *software* pengujian otomatis (*automated testing software*). Penilaian faktor *reusability* menggunakan *ProjectCodeMeter*. Sedangkan penilaian faktor *interoperability* menggunakan *SoapUI*.

Makalah ini bertujuan untuk mengetahui *e-commerce* mana yang paling baik dari segi faktor *portability*, *reusability*, dan *interoperability* setelah diuji dengan *automated testing software*. Diharapkan makalah ini memberikan manfaat bagi para pelaku usaha yang ingin menerapkan *e-commerce* menggunakan *open source* yang mudah diterapkan di berbagai *platform*, pemeliharannya lebih mudah, dan hemat biaya.

## II. LANDASAN TEORI

Faktor kualitas *software* menurut McCall terbagi dalam tiga kategori, yaitu *product operation factors*, *product revision factors*, dan *product transition factors* [1]. *Product transition factors* memiliki tiga faktor, yaitu *portability*, *reusability*, dan *interoperability* [1].

### A. Portability

*Portability* terkait dengan kemampuan adaptasi dari perangkat lunak terhadap lingkungan/*platform* yang terdiri dari *hardware* dan sistem operasi yang berbeda [1]. Faktor *portability* terbagi dalam beberapa subfaktor seperti *modularity*, *Software system independence*, dan *self-descriptive* [1]. *Modularity* merupakan pemecahan *software system* menjadi bagian-bagian kecil yang disebut modul bertujuan agar *software* mudah dipahami dan memiliki struktur yang mudah [6]. *Software system independence* merupakan tingkatan dimana program mandiri terhadap fitur bahasa pemrograman nonstandar, karakteristik sistem operasi, dan batasan-batasan lingkungan lainnya [1]. *Self-descriptive* merupakan dokumentasi dari *code* yang merupakan penjelasan dari implementasi dari fungsi atau *class* yang berbentuk *command* [1].

### B. Reusability

*Reusability* terkait beberapa manajemen eksplisit seperti isu pembuatan, pengemasan, distribusi, pemasangan, konfigurasi, penyebaran, pemeliharaan, dan pembaruan aplikasi [7]. Berdasarkan model yang dibuat McCall, *reusability* terdapat subfaktor yaitu *modularity*, *document accessibility*, *Software system independence*, *application independence*, *self-descriptive*, *generality*, dan

Rifqi Imaduddin Irfan adalah mahasiswa Program Studi Strata Satu (S-1) Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret (UNS) Surakarta, Jawa Tengah, Indonesia (e-mail: [rifqi.irfan@student.uns.ac.id](mailto:rifqi.irfan@student.uns.ac.id)).

Aji Saka Nugraha adalah mahasiswa Program Studi Strata Satu (S-1) Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret (UNS) Surakarta, Jawa Tengah, Indonesia (e-mail: [ajisaka@student.uns.ac.id](mailto:ajisaka@student.uns.ac.id)).

Rini Angrainingsih adalah dosen Program Studi Strata Satu (S-1) Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret (UNS) Surakarta, Jawa Tengah, Indonesia (phone: +6289673932645; e-mail: [rini.angrainingsih@staff.uns.ac.id](mailto:rini.angrainingsih@staff.uns.ac.id)).

*simplicity* [1]. *Modularity* merupakan pemecahan *software* system menjadi bagian-bagian kecil yang disebut modul bertujuan agar *software* mudah dipahami dan memiliki struktur yang sederhana [2]. *Document accessibility* merupakan kemudahan dokumen dapat diakses oleh berbagai user seperti penyandang disabilitas. *Software system independence* merupakan tingkatan dimana program mandiri terhadap fitur bahasa pemrograman nonstandar, karakteristik sistem operasi, dan batasan-batasan lingkungan lainnya [1]. *Application independence* merupakan atribut *software* yang menentukan apakah sebuah aplikasi tergantung terhadap sistem operasi, *input output* atau bergantung terhadap aplikasi yang lain [8]. *Self-descriptive* merupakan dokumentasi dari *code* yang merupakan penjelasan dari implementasi dari fungsi-fungsi atau *class-class*, dapat berbentuk *command* untuk penjelasan [1]. *Generality* merupakan penilaian *software* untuk dapat menyelesaikan masalah pada domainnya [9]. *Simplicity* adalah *software* dapat dimengerti tanpa kesulitan [9].

### C. Interoperability

*Interoperability* adalah kemampuan dua atau lebih komponen perangkat lunak untuk bekerjasama meskipun terdapat perbedaan dalam bahasa, tampilan, dan *platform* eksekusi program [10]. Faktor *interoperability* terdiri dari beberapa subfaktor yaitu *commonality*, *system compatibility*, *software system independence*, dan *modularity* [1]. *Commonality* adalah penggunaan struktur dan tipe data standar di seluruh program [11]. *System compatibility* adalah kemampuan perangkat lunak yang dikembangkan untuk berjalan pada *hardware*, sistem operasi, aplikasi, ataupun lingkungan jaringan yang berbeda [12]. *Software system independence* merupakan tingkatan dimana program mandiri terhadap fitur bahasa pemrograman nonstandar, karakteristik sistem operasi, dan batasan-batasan lingkungan lainnya [1]. *Modularity* merupakan pemecahan sistem *software* menjadi bagian-bagian kecil yang disebut modul bertujuan agar *software* mudah dipahami dan memiliki struktur yang sederhana [2].

### III. TINJAUAN PUSTAKA

Penelitian yang terkait dalam paper ini antara lain

1) Penelitian yang berjudul “*Using use case patterns to estimate reusability in software systems*” menjelaskan penggunaan pola *use case* sebagai dasar penghitungan biaya pembuatan *software*. Untuk melakukan penelitian ini, diperlukan pembuatan katalog pola *use case*. Katalog dianalisis untuk memperkirakan potensi *reusability* dalam *software* yang berbeda [13].

2) Penelitian yang berjudul “*Issues in the Specification and Measurement of Software Portability*” menjelaskan tentang *portability* sebagai subfaktor yang paling diminati dalam kualitas *software*. Akan tetapi, *portability* belum dibahas secara eksplisit baik oleh kurikulum rekayasa perangkat lunak maupun metodologi pengembangan *software* pada umumnya [14].

3) Penelitian yang berjudul “*A generic interoperability testing framework and a systematic development process for automated interoperability testing*”, menjelaskan pengujian *interoperability* otomatis dengan pemeriksaan pesan, yang menilai kecocokan pesan yang ditukarkan antar sistem. Pengujian ini juga menjelaskan penilaian, otomatisasi, dan keterbatasan sistem yang diuji [15].

### IV. METODOLOGI PENELITIAN

Pengukuran faktor *portability* melalui dua tahap pengumpulan data dan membandingkan hasil studi literatur [16].

#### 1) Pengumpulan data

Pengumpulan data dilakukan dengan metode studi literatur dan *technical report* tiap *e-commerce*. Pencarian data dilakukan dengan melihat dokumentasi *software* yang ada di setiap website *e-commerce* antara lain sebagai berikut

- *OpenCart* (<http://docs.opencart.com>)
- *Magento* (<https://magento.com/resources/technica>)
- *WooCommerce* (<https://docs.woocommerce.com>).

#### 2) Membandingkan hasil studi literatur

Data yang diperoleh selanjutnya dievaluasi. Data-data akan dimasukkan ke dalam sebuah tabel perbandingan untuk membandingkan ketiga *open source e-commerce*. Evaluasi dilakukan pada tiga subfaktor *portability* berdasarkan model McCall yaitu *modularity*, *software system independence*, dan *self-descriptive*.

Sedangkan pengukuran faktor *reusability* dan *interoperability* melalui dua tahap yaitu *instal* dan konfigurasi *open source software* dan pengujian menggunakan *automated testing software* [6].

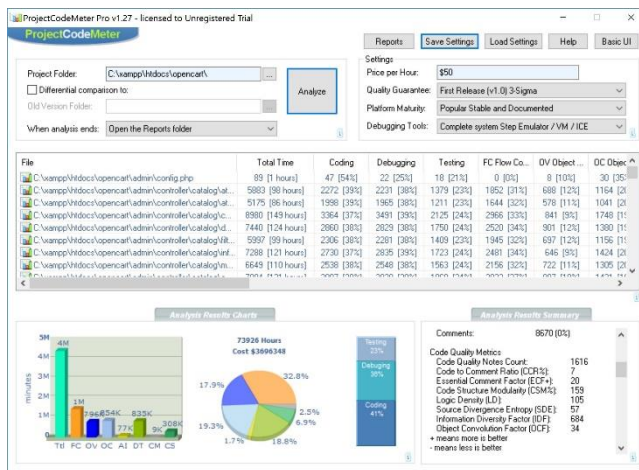
#### 1) Instal *open source software*. *Open source* dapat diunduh di masing-masing *website e-commerce software*.

- <https://magento.com> untuk *software e-commerce Magento*
- <https://www.opencart.com> untuk *software e-commerce OpenCart*
- <https://woocommerce.com> untuk *software e-commerce WooCommerce*

Konfigurasi database dan server dapat dilakukan dengan menggunakan *open source webserver* seperti XAMPP, WAMP, dan LAMP.

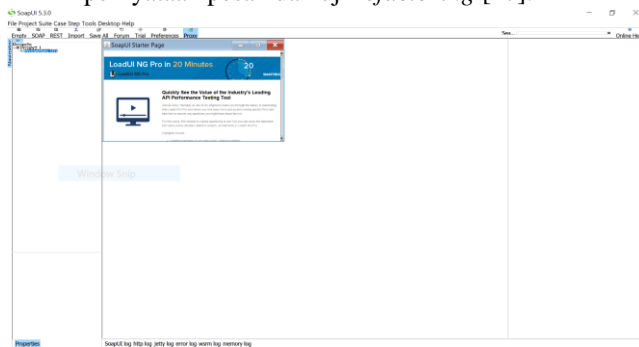
#### 2) Pengujian dengan *automated testing software*

Untuk pengukuran faktor *reusability* menggunakan *software ProjectCodeMeter* versi 1.27, sedangkan untuk pengukuran faktor *interoperability* menggunakan *software SoapUI* versi 5.3.0.



Gambar 1 Tampilan Software ProjectCodeMeter

Software *ProjectCodeMeter* merupakan software untuk project manager yang digunakan untuk mengukur estimasi waktu, biaya, kompleksitas, dan kualitas dari proyek *software* dengan baik [6]. Software *SoapUI* merupakan *software* untuk melakukan *functional testing* terhadap REST API dan SOAP *web service*, WSDL *coverage*, uji pernyataan pesan dan uji *refactoring* [17].



Gambar 2 Tampilan SoapUI

## V. HASIL DAN PEMBAHASAN

Berikut ini adalah hasil dan pembahasan perbandingan *portability*, *reusability*, dan *interoperability* pada *open source e-commerce Magento*, *OpenCart*, dan *WooCommerce*.

### A. Portability

Dengan merujuk pada dokumentasi *e-commerce* yang terdapat di *website* resminya didapatkan hasil seperti pada Tabel 1 dibawah [18]–[20].

Open Source	Modularity	Self-descriptive	Simplicity
Magento	Komponen file-file program sudah terstruktur & disusun	1. Berjalan di semua platform dan sistem operasi	Menyediakan dokumentasi yang jelas 1. deskripsi aplikasi

	<p>sesuai fungsi masing-masing.</p> <p>index.php user.ini</p>	<p>lainnya yang mendukung PHP dan database.</p> <p>2. Membutuhkan aplikasi <i>web server</i> dan <i>database MySQL</i>.</p> <p>3. Disarankan menggunakan <i>php_xdebug2.2.0</i> dan <i>PHPUnit</i>.</p>	<p>2. <i>source code</i> program</p> <p>3. informasi rilis</p> <p>4. <i>end user, developer, enterprise</i></p> <p>5. cara migrasi data dan <i>code</i></p> <p>6. <i>security patches</i></p>
OpenCart	<p>Komponen file-file program sudah terstruktur &amp; disusun sesuai fungsi masing-masing.</p> <p>config.php index.php php.ini</p>	<p>1. Berjalan di semua platform dan sistem operasi lainnya yang mendukung PHP dan database.</p> <p>2. Membutuhkan aplikasi <i>web server</i> dan <i>database MySQL</i>.</p>	<p>Menyediakan dokumentasi yang jelas</p> <p>1. deskripsi aplikasi</p> <p>2. <i>source code</i> program</p> <p>3. informasi rilis</p> <p>4. <i>end user, developer, enterprise</i></p> <p>5. cara migrasi data dan <i>code</i></p> <p>6. <i>security patches</i></p>
WooCommerce	<p>Komponen file-file program sudah terstruktur &amp; disusun sesuai fungsi masing-masing.</p> <p>index.php wp-config.php wp-load.php wp-login.php wp-settings.php xmlrpc.php</p>	<p>1. Berjalan di semua platform dan sistem operasi lainnya yang mendukung PHP dan database.</p> <p>2. Membutuhkan aplikasi <i>web server</i> dan <i>database MySQL</i>.</p> <p>3. Menambahkan CMS</p>	<p>Menyediakan dokumentasi yang jelas</p> <p>1. deskripsi aplikasi</p> <p>2. <i>extensions</i></p> <p>3. <i>codex (snippets, third-party plugins, theming, version notes)</i></p> <p>4. <i>developer documentation</i></p> <p>5. dokumentasi tema</p>

		Word-press	6. cara menjalankan program
--	--	------------	-----------------------------

Tabel 1 Hasil Pengujian Portability

Dari hasil perbandingan *portability* masing-masing *e-commerce* dapat disimpulkan *OpenCart* memiliki hasil yang lebih baik. Alasannya pada subfaktor *software system independence* terlihat *OpenCart* tidak membutuhkan aplikasi tambahan selain *webserver* dan *database MySQL*. Selain itu pada subfaktor *self-descriptive OpenCart* juga menyediakan dokumentasi dalam bentuk *e-book* gratis dan berbayar yang bisa diunduh secara bebas dan dijadikan rujukan *user* dan *developer* dalam mengelola *e-commerce OpenCart*.

### B. Reusability

Nilai *modularity* didapatkan dari nilai *Code Structure Modularity (CSM)* yang merupakan nilai ketelitian *code* terbagi menjadi fungsi dan *class*. Nilai CSM yang kurang dari 100 menandakan modularitas yang kecil, sementara nilai CSM yang lebih dari 100 menandakan modularitas yang besar. Nilai CSM terbaik adalah yang mendekati 100 atau *code* di setiap modul seimbang.

Nilai *self descriptive* didapatkan dari nilai *Code to Command Ratio (CCR)* yang merupakan nilai keseimbangan antar baris komentar (*command line*) dan *Logical Lines of Code (LLOC)*. Semakin mendekati nilai 100, setiap baris *code statement* 100% terdapat baris komentar.

Nilai *simplicity* didapatkan dari nilai *Flow Complexity (FC)*. Nilai FC yang kecil menandakan tingkat *simplicity* yang tinggi, sebaliknya nilai FC yang besar menandakan tingkat *simplicity* yang rendah. Nilai FC dihitung berdasarkan nilai *linear independent path* pada *code*.

Berikut ini hasil pengujian *reusability* pada ketiga *E-commerce*.

Open Source	Modularity	Self-descriptive	Simplicity
Magento	159	23%	26%
OpenCart	159	7%	33%
WooCommerce	159	18%	41%

Tabel 2 Hasil Pengujian Reusability

Dari hasil pengujian diatas, didapatkan hasil *reusability* terbaik adalah *Magento* dengan *modularity* sebesar 159, *self-descriptive* sebesar 23%, dan *simplicity* sebesar 26%. Hasil *reusability* menunjukkan tingkat modul pada suatu *code* lebih mudah untuk digunakan kembali untuk kepentingan *development* pada *requirement* yang sama. Selain itu juga lebih mudah untuk dilakukan pemeliharaan (*maintenance*) sehingga dapat mengurangi biaya (*cost*) yang dikeluarkan.

### C. Interoperability

Hasil *interoperability* di semua yang diujikan menunjukkan tidak ada kesalahan tipe data, metode, yang mengganggu kinerja *open source e-commerce*. Sehingga disimpulkan semua *e-commerce* mampu dijalankan di *hardware*, sistem operasi, dan *platform* yang berbeda.

Berikut ini hasil *interoperability*.

Open Source	Action	Result
Magento	...	PASS
OpenCart	...	PASS
WooCommerce	...	PASS

Tabel 3 Hasil Pengujian Interoperability

Dari hasil pengujian diatas, *OpenCart* tidak ditemukan kegagalannya sama sekali. Sementara pada *WooCommerce* dan *Magento* terjadi kegagalan proses pengujian masing-masing satu dan dua kali dalam.

## VI. PENUTUP

*E-commerce OpenCart* memiliki nilai *portability* yang lebih baik dibandingkan *Magento* dan *WooCommerce*. *OpenCart* berjalan di semua *platform* tanpa aplikasi tambahan dan mempunyai dokumentasi sistem yang jelas. *E-commerce* dengan nilai *reusability* terbaik adalah *Magento*. *Magento* lebih mudah untuk digunakan kembali untuk kepentingan *development* pada *requirement* yang sama. *E-commerce* dengan *interoperability* terbaik adalah *OpenCart*. *OpenCart* dapat digunakan lebih mudah bersama dengan aplikasi-aplikasi lain yang berbeda dari segi bahasa, tampilan, dan *platform*.

## DAFTAR PUSTAKA

- [1] D. Galin, *Software Quality Assurance: From Theory to Implementation*. 2004.
- [2] A. Emanuel, R. Wardoyo, J. Istiyanto, and K. Mustofa, "Modularity Index Metrics for Java-Based Open Source Software Projects," *Development*, vol. 2, no. 11, pp. 52–58, 2011.
- [3] S. Fyffe, "Top 10 Open Source eCommerce Platforms," *CMS Critic*, 2015. [Online]. Available: <https://www.cmscritic.com/top-10-open-source-ecommerce-platforms/>. [Accessed: 11-Jun-2017].
- [4] J. Tian, *Software quality engineering: testing, quality assurance, and quantifiable improvement*. John Wiley & Sons, 2005.
- [5] M. Dannelly, Stephen; El Alliali, "Quality Factors," *Strategic Direction*, vol. 29, no. 1. pp. 27–29, 2012.
- [6] E. Wihidayat, "Perbandingan Subfaktor Reusability Pada Open Source Learning Management System," *Pros. SNST ke-7 Tahun 2016 Fak. Tek. Univ. Wahid Hasyim Semarang* 225, pp. 225–229, 2016.
- [7] M. K. Kaur, M. R. Rani, and M. Jagdeep, "Code Reuse & Reusability of the Software," pp. 28–30, 2013.
- [8] OOCities, "Exam2studyguide," 2009. [Online]. Available: <http://www.oocities.org/hidonovan/Exam2studyguide.htm>. [Accessed: 28-May-2017].
- [9] A. M. Bachtiar, D. Dharmayanti, M. K. Sabariah, A. M. Bachtiar, D. Dharmayanti, and M. K. S., "Vol.11 No. 2," vol. 11, no. 2, pp. 224–233.
- [10] P. Wegner, "Interoperability," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 285–287, 1996.
- [11] V. E. M. Gultom, "SOFTWARE QUALITY MODEL (MODEL PERANGKAT LUNAK)," 2010. [Online]. Available: <https://vaskoedo.wordpress.com/2010/09/30/software-quality-model-modelperangkat-lunak/>.
- [12] U. Fatimah, "Aspek Compatibility," 2014. [Online]. Available: <http://fatimahumi.blogspot.co.id/2014/03/compatibility-testing.html>. [Accessed: 29-May-2017].
- [13] A. Issa, M. Odeh, and D. Coward, "Using use case patterns to

- estimate reusability in software systems,” *Inf. Softw. Technol.*, vol. 48, no. 9, pp. 836–845, 2006.
- [14] J. D. Mooney, “Issues in the Specification and Measurement of Software Portability,” pp. 1–16, 2016.
- [15] T. Rings, P. Poglitsch, S. Schulz, L. Serazio, and T. Vassiliou-Gioles, “A Generic Interoperability Testing Framework and a Systematic Development Process for Automated Interoperability Testing,” *Int. J. Softw. Tools Technol. Transf.*, vol. 16, no. 3, pp. 295–313, 2014.
- [16] T. Anggrainingsih, Rini; Wahyuningsih Dessy; Asmari, Ersi Indah; Khusyaini, Achmad; Rejekiningsih, “Perbandingan Faktor Correctness, Efisiensi dan Portability ... (Anggrainingsih dkk.),” *Pros. SNST ke-7 Tahun 2016 Fak. Tek. Univ. Wahid Hasyim Semarang*, pp. 208–213, 2016.
- [17] SmartBear, “SoapUI Open Source,” 2017. [Online]. Available: <https://www.soapui.org/open-source.html>. [Accessed: 15-Jun-2017].
- [18] Magento, “Documentation and Resources,” 2017. [Online]. Available: <https://magento.com/resources/technical>. [Accessed: 29-May-2017].
- [19] Opencart.com, “System Requirements - OpenCart Documentation,” *OpenCart*, 2017. [Online]. Available: <http://docs.opencart.com/requirements/>. [Accessed: 29-May-2017].
- [20] WooCommerce, “WooCommerce Docs,” 2017. [Online]. Available: <https://docs.woocommerce.com>. [Accessed: 29-May-2017].