

## PERBANDINGAN SUBFAKTOR *REUSABILITY* PADA *OPEN SOURCE LEARNING MANAGEMENT SYSTEM*

**Endar Suprih Wihidayat<sup>1</sup>, Tiyas Sulistyoningrum<sup>2</sup>, Rini Anggrainingsih<sup>2\*</sup>**

<sup>1</sup>Program Studi Pendidikan Teknik Informatika dan Komputer,  
Fakultas Keguruan dan Ilmu Pendidikan

<sup>2</sup>Program Studi Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam,  
Universitas Sebelas Maret

Jl. Ir. Sutami 36A Ketingan Surakarta 57126

\*Email: rini.anggrainingsih@staff.uns.ac.id

### Abstrak

*Reusability pada software dinilai untuk mengetahui tingkat penggunaan ulang code program software. Paper ini fokus pada perbandingan nilai Reusability pada Learning Management System (LMS). Perbandingan yang dilakukan ini bertujuan untuk mengetahui LMS mana yang memiliki nilai reusability yang lebih baik sehingga dapat digunakan ulang oleh programmer serta dapat meningkatkan produktifitas dan menghemat cost dan maintainnya lebih mudah. LMS yang dipilih adalah Moodle, Dokeos dan ATutor karena termasuk ke dalam Top 8 Open Source LMS berdasarkan komunitas online terbesar eLearning Industry. Penilaian Reusability berdasarkan sub faktor: Modularity, Document Accessibility, Software System Independence, Application independence, Self descriptive, Generality, dan Simplicity. Metode yang digunakan dalam perhitungan modularity, self descriptive, dan simplicity menggunakan tools ProjectCodeMeter. Sedangkan untuk document accessibility, software system independence, application independence dan generality berdasarkan informasi website resmi terkait. Hasil akhir yang didapat adalah Moodle memiliki nilai reusability yang lebih baik dibandingkan dengan Dokeos dan ATutor.*

**Kata kunci :** *Learning Management System(LMS), ProjectCodeMeter, Sub Faktor Reusability*

### 1. PENDAHULUAN

Tanpa adanya pengukuran kualitas *software*, tingkat kualitas *software* tidak dapat dipastikan (Tuteja dan Dubey, 2012). Salah satu faktor kualitas *software* adalah *reusability* yang merupakan cara yang baik untuk meningkatkan produktifitas dan *maintainability software* yang dapat membantu programmer pada saat mendevlop karena code-code tersebut dapat digunakan kembali pada project lain yang memiliki *requirement* yang sama (Paliwal dkk.,2014). Ketika dilakukan penambahan fitur pada *LMS* biasanya mengeluarkan cost yang banyak (Bickford, 2013). Manfaat yang didapatkan ketika diketahui *LMS* mana yang bernilai *reusability* baik adalah kita dapat mengetahui apakah *LMS* tersebut mudah dilakukan *maintain* dan *cost* yang dikeluarkan dapat berkurang.

Berdasarkan McCall model *reusability* terdapat sub faktor yaitu *modularity*, *document accessibility*, *software system independence*, *application independence*, *self descriptive*, *generality* dan *simplicity* (Galín, 2004). *Modularity* merupakan pemecahan *software system* menjadi bagian bagian kecil yang disebut modul bertujuan agar *software* mudah dipahami dan memiliki struktur yang simple (Emanuel dkk., 2011). *Document accessibility* merupakan kemudahan dokumen dapat diakses oleh berbagai user seperti penyandang disabilitas yang dijelaskan pada <http://accessibility.psu.edu/accommodations/definition/>. *Software system independence* merupakan tingkatan dimana program mandiri terhadap fitur bahasa pemrograman nonstandar, karakteristik sistem operasi, dan batasan-batasan lingkungan lainnya (Galín, 2004). *Application independence* merupakan atribut *software* yang menentukan apakah sebuah aplikasi tergantung terhadap *Operating System*, *input output* atau bergantung terhadap aplikasi yang lain seperti yang dijelaskan pada <http://www.oocities.org/hidonovan/Exam2studyguide.htm>. *Self descriptive* merupakan dokumentasi dari *code* yang merupakan penjelasan dari implementasi dari fungsi fungsi atau *class*, dapat berbentuk *command* untuk menjelaskan (Galín, 2004). *Generality* merupakan penilaian *software* untuk dapat menyelesaikan masalah pada domainnya (Bachtiar dkk., 2013). *Simplicity* adalah *software* dapat dimengerti tanpa kesulitan (Bachtiar dkk., 2013).

*LMS* yang dibandingkan pada paper ini merupakan *Open Source LMS* karena *Open Source* memiliki *user documentation* yang lengkap dan merupakan *software* dengan tingkat *accessibility* yang baik (Heron dkk., 2013). *Open Source LMS* yang termasuk ke dalam kategori populer adalah *Moodle*, *ATutor*, *Eliademy*, *Forma LMS*, *Dokeos*, *ILIAS*, *Opigno*, dan *OLAT* (Pappas, 2015). Pada paper ini *LMS* yang dibandingkan adalah *Moodle* versi 3.0.3, *ATutor* versi 2.2.1 dan *Dokeos* versi 2.1.1 yang merupakan versi terbaru pada saat ini dan stabil. Penilaian *reusability* pada paper ini menggunakan *software ProjectCodeMeter*. *Software ProjectCodeMeter* merupakan *software* untuk *project manager* yang digunakan untuk mengukur *time estimation*, *cost*, *complexity*, dan *quality* dari *software project* dengan baik. Cara menggunakannya adalah dengan input direktori *project* yang akan diukur lalu pilih *Analyze* (*ProjectCodeMeter*).

## 2. METODOLOGI

Metodologi yang digunakan dalam paper ini menggunakan *software ProjectCodeMeter* dan dari informasi *website* resmi terkait. Pengumpulan data menggunakan *source code* yang diunduh dari masing - masing *website* resmi. Berikut ini merupakan tahapan yang dilakukan dalam penilaian sub faktor *Reusability* :

- Mengunduh *source code* dari *website* tiap *LMS* dan *software ProjectCodeMeter*. *Website* dari *Moodle* yaitu [www.moodle.org](http://www.moodle.org), *ATutor* yaitu [www.atutor.ca](http://www.atutor.ca), *Dokeos* yaitu [www.dokeos.com](http://www.dokeos.com), dan *software ProjectCodeMeter* yaitu [www.ProjectCodeMeter.com](http://www.ProjectCodeMeter.com).
- Menilai *modularity*, *self descriptive* dan *simplicity* menggunakan *ProjectCodeMeter* dengan cara memasukkan direktori file *source code* tiap *LMS* pada kolom *project folder* dan pilih *Analyze*. Setelah itu hasil nilai yang didapatkan dapat dilihat di folder bernama *.PCMReports* yang terdapat di dalam folder *source code* tiap *LMS* yang telah dianalisis dengan *ProjectCodeMeter*. Nilai hasil terletak pada file *ProjectCodeMeter\_Summary*. Nilai *modularity* terletak pada nilai *Code Structure Modularity (CSM)*, nilai *simplicity* terletak pada nilai *Flow Complexity (FC)* dan nilai *self descriptive* terletak pada nilai *Code to Command Ratio (CCR)*.
- Menilai *document accessibility*, *software system independence*, *application independence* dan *generality* dengan menggunakan informasi dari *website* resmi terkait.
- Mengambil kesimpulan dari masing-masing hasil penilaian sub faktor *reusability* tiap *LMS*.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Pembahasan

#### a. Modularity

Nilai *modularity* terletak pada nilai *Code Structure Modularity (CSM)* dari hasil pengukuran menggunakan *software ProjectCodeMeter* seperti pada Tabel 1. *CSM* merupakan nilai dari sejauh mana *code* terbagi menjadi fungsi dan *class*. Nilai *CSM* yang baik adalah mendekati 100 yang artinya nilai *code* pada setiap modulnya *balance*. Sedangkan nilai yang kurang, memiliki modularitas yang kecil dan nilai yang besar memiliki *code* yang terlalu terfragmentasi atau terpecah.

**Tabel 1. Hasil sub faktor Modularity**

<i>LMS</i>	Modularity	Keterangan
Moodle	162	Memiliki selisih 62 dengan 100
Atutor	141	Memiliki selisih 41 dengan 100
Dokeos	81	Memiliki selisih 19 dengan 100

Nilai *modularity* dari *Moodle* dan *ATutor* melebihi angka 100. Jika angka melebihi 100 maka *code* untuk tiap modulnya terlalu terfragmentasi atau terlalu terpecah-pecah menjadi bagian yang terlalu banyak yang akan mengakibatkan *maintenance* menjadi sulit. Dari ketiga hasil, yang memiliki *modularity* paling mendekati 100 adalah *Dokeos*.

## b. Self descriptive

Dengan menggunakan *ProjectCodeMeter*, nilai *self descriptive* terletak pada nilai *Code to Command Ratio (CCR)*. *CCR* merupakan nilai keseimbangan antara garis komentar atau *Command Line* dengan *code statements* atau *Logical Lines of Code (LLOC)*. Nilai sempurna adalah 100 yang artinya adalah setiap baris *code statement* 100% terdapat baris komentar.

**Tabel 2. Hasil sub faktor self descriptive**

LMS	Hasil
Moodle	25 %
Atutor	16 %
Dokeos	14 %

Dari hasil, yang memiliki nilai *CCR* yang tinggi adalah *Moodle* yaitu 25%. *ATutor* memiliki nilai 16 % dan *Dokeos* yang paling sedikit yaitu 14%.

## c. Simplicity

Hasil pengukuran dapat dilihat pada Tabel 3. Dengan menggunakan *tool ProjectCodeMeter*, yang menjadi acuan adalah nilai *Flow Coplexity (FC)*. Jika nilai *FC* kecil, maka tingkat *simplicity*nya tinggi dan berlaku sebaliknya. Nilai *FC* pada *ProjectCodeMeter* dihitung berdasarkan nilai *linear independent path* pada codenya seperti pada teori McCabe.

**Tabel 3. Hasil pengukuran Simplicity**

LMS	Hasil	Keterangan
Moodle	36%	Baik
ATutor	39%	Baik
Dokeos	30%	Baik

Nilai yang baik adalah kurang dari 50%. Dari hasil yang diperoleh, yang paling baik adalah *Dokeos* karena nilai *FC* nya paling kecil.

## d. Document accessibility

Hasil dari penilaian *document accessibility* dapat dilihat pada Tabel 4. Pada Tabel 4, terdapat kriteria apa saja yang harus ada dalam *software* terkait *accessibility* seperti yang dijelaskan pada <http://elearningfacultymodules.org/images/AccessibilityStandards.pdf>. Hasil yang diperoleh berdasarkan dari informasi *website* resmi *Moodle* yaitu [www.moodle.org](http://www.moodle.org), *ATutor* yaitu [www.atutor.ca](http://www.atutor.ca), dan *Dokeos* yaitu [www.dokeos.com](http://www.dokeos.com). Penilaian dengan menggunakan **Ya** apabila memenuhi kriteria dan **Tidak** apabila tidak memenuhi kriteria.

**Tabel 4. Hasil penilaian sub faktor document accessibility**

Kriteria	Moodle	Atutor	Dokeos
Web accessibility	Ya	Ya	Tidak
Word file accessibility	Ya	Ya	Ya
PDF accessibility	Ya	Ya	Ya
PowerPoint accessibility	Ya	Ya	Ya
Software accessibility	Ya	Ya	Ya
Hasil *	100%	100%	80%

\* nilai perbandingan dalam presentase yang memenuhi kriteria

*Standard* yang digunakan dalam *web accessibility* adalah menggunakan *WCAG (Web Content Accessibility Guidelines)* standard. Untuk *ATutor* menggunakan *WCAG*

v 1.0 dan v 2.0 untuk *Moodle*, menggunakan standard WCAG v2.0 sedangkan *Dokeos* belum memiliki *standard accessibility*.

e. Software system independence

Hasil penilaian *software system independence* dapat dilihat pada Tabel 5. Ketiga *LMS* merupakan *system independence* yang tidak membutuhkan *software* lain ketika dilakukan instalasi.

**Tabel 5. Hasil sub faktor system independence**

<i>LMS</i>	Hasil	Keterangan
Moodle	Independence	Moodle tidak memerlukan software lain untuk diinstall
ATutor	Independence	ATutor tidak memerlukan software lain untuk diinstall
Dokeos	Independence	Dokeos tidak memerlukan software lain untuk diinstall

f. Application independence

Hasil dari *application independence* dapat dilihat pada Tabel 6. Moodle memiliki aplikasi *mobile* yang *independence*.

**Tabel 6. Hasil sub faktor application independence**

<i>LMS</i>	Hasil	Keterangan
Moodle	Memiliki aplikasi mobile	Baik, aplikasi dikatakan independent karena tidak bergantung aplikasi lain.
ATutor	ATutor tidak memiliki app	-
Dokeos	Dokeos tidak memiliki app	-

g. Generality

Hasil penilaian *generality* dapat dilihat pada Tabel 7. Kolom fungsional pada Tabel 7 merupakan fungsional *requirement* yang harus ada di dalam *LMS* (Amiroh, 2012).

**Tabel 7. Hasil penilaian sub faktor generality**

Fungsional	Moodle	ATutor	Dokeos
Pengelolaan hak akses	Ada	Ada	Ada
Pengelolaan courses	Ada	Ada	Ada
Pengelolaan aktivitas	Ada	Ada	Ada
Pengelolaan nilai	Ada	Ada	Ada
Pengelolaan bahan ajar (resources)	Ada	Ada	Ada
Pengelolaan visualisasi e-learning	Ada	Ada	Ada

Semua *LMS* yang dibandingkan memiliki semua kriteria fungsional *LMS*. Maka Moodle, ATutor dan Dokeos memiliki kriteria *Generality* yang baik.

### 3.2 Hasil

Kesimpulan hasil dari setiap sub faktor untuk masing masing *LMS* dapat dilihat pada Tabel 8.

**Tabel 8. Hasil sub faktor masing - masing LMS**

Sub faktor	Moodle	Atutor	Dokeos	LMS yang lebih baik
Modularity	162	141	81	Dokeos
Document Accessibility	100%	100%	80%	Moodle, Atutor
Software system independence	Independence	Independence	independence	Moodle, ATutor, Dokeos
Application independence	Independence	-	-	Moodle
Self descriptive	25 %	16 %	14 %	Moodle
Generality	100 %	100%	100%	Moodle, ATutor, Dokeos
Simplicity	36%	39%	30%	Dokeos

#### 4. KESIMPULAN

Dari perhitungan antara *LMS Moodle*, *ATutor* dan *Dokeos* yang memiliki nilai *reusability* yang lebih baik adalah Moodle. Nilai *reusability* yang baik memiliki arti bahwa tingkat modul pada codenya lebih mudah untuk digunakan kembali untuk kepentingan *development* pada *requirement* yang sama. Selain itu juga lebih mudah untuk dilakukan *maintain* sehingga dapat mengurangi *cost* yang dikeluarkan.

#### DAFTAR PUSTAKA

- Amiroh. 2012. *Kupas Tuntas membangun elearning dengan LMS Moodle versi 2*, Sidoarjo, Genta Group Production.
- Bachtiar, A. M., Dharmayanti, D. & Sabariah, M. K. 2013. Analisis Kualitas Perangkat Lunak Terhadap Sistem Informasi UNIKOM. *Majalah Ilmiah UNIKOM*.
- Bickford, A. 2013. *12 Common Complaints about LMS*. <http://connectthinking.com.au/12-common-complaints-about-learning-management-systems-lms/> . Diakses : 9 Mei 2016, jam 19.00.
- Emanuel, A. W. R., WARDOYO, R. & ISTIYANTO, J. E. 2011. Modularity Index Metrics for Java-Based Open Source Software Projects. *International Journal of Advanced Computer Science and Applications*, 2.
- Galín, D. 2004. *Software Quality Assurance from theory to implementation*, England, Pearson Education Limited.
- Heron, M., L Hanson, V. & Ricketts, I. 2013. Open Source and Accessibility : Anvantages and Limitations. *Journal of Interaction Science*.
- Paliwal, N., Shrivastava, V. & Tiwari, K. 2014. An Approach to Find Reusability of Software Using Obejct Oriented Metrics. *International Journal of Innovative Research in Science Engineering and Technology*, 3.
- Pappas, C. 2015. *The Top 8 Open Source Learning Management Systems*. <http://elearningindustry.com/top-open-source-learning-management-systems>. Diakses : 9 Mei 2016, jam 16.45.
- ProjectCodeMeter.*ProjectCodeMeter*. [http://www.projectcodemeter.com/cost\\_estimation/index.html](http://www.projectcodemeter.com/cost_estimation/index.html). Diakses : 9 Mei 2016, jam 17.49.
- Tuteja, M. & Dubey, G. 2012. A Research Study on importance of Testing and Quality Assurance in Software Development Life Cycle ( SDLC) Models. *International Journal of Soft Computing and Engineering(IJSCE)*, 2.