

Jurnal Ilmiah

# DASI

DATA MANAJEMEN DAN TEKNOLOGI INFORMASI



STMIK AMIKOM  
YOGYAKARTA

**VOL. 16 NO. 3 SEPTEMBER 2015**  
**JURNAL ILMIAH**  
**Data Manajemen Dan Teknologi Informasi**

---

Terbit empat kali setahun pada bulan Maret, Juni, September dan Desember berisi artikel hasil penelitian dan kajian analitis kritis di dalam bidang manajemen informatika dan teknologi informatika. ISSN 1411-3201, diterbitkan pertama kali pada tahun 2000.

**KETUA PENYUNTING**

Abidarin Rosidi

**WAKIL KETUA PENYUNTING**

Heri Sismoro

**PENYUNTING PELAKSANA**

Kusrini

Emha Taufiq Luthfi

Hanif Al Fatta

Anggit Dwi Hartanto

**STAF AHLI (MITRA BESTARI)**

Jazi Eko Istiyanto (FMIPA UGM)

H. Wasito (PAU-UGM)

Supriyoko (Universitas Sarjana Wiyata)

Jano Hendarto (FMIPA-UGM)

Sri Mulyana (FMIPA-UGM)

Winoto Sukarno (AMIK "HAS" Bandung)

Rum Andri KR (AMIKOM)

Arief Setyanto (AMIKOM)

Krisnawati (AMIKOM)

Ema Utami (AMIKOM)

**ARTISTIK**

Amir Fatah Sofyan

**TATA USAHA**

Lya Renyta Ika Puteri

Murni Elfiana Dewi.

**PENANGGUNG JAWAB :**

Ketua STMIK AMIKOM Yogyakarta, Prof. Dr. M. Suyanto, M.M.

**ALAMAT PENYUNTING & TATA USAHA**

STMIK AMIKOM Yogyakarta, Jl. Ring Road Utara Condong Catur Yogyakarta, Telp. (0274) 884201 Fax. (0274) 884208, Email : jurnal@amikom.ac.id

**BERLANGGANAN**

Langganan dapat dilakukan dengan pemesanan untuk minimal 4 edisi (1 tahun) pulau jawa Rp. 50.000 x 4 = Rp. 200.000,00 untuk luar jawa ditambah ongkos kirim.

## DAFTAR ISI

HALAMAN JUDUL.....	i
KATA PENGANTAR .....	ii
DAFTAR ISI.....	iii
Perlindungan Data Terhadap Serangan Menggunakan Metoda Tebakan Pada Sistem Operasi Linux.....	1-8
Akhmad Dahlan (Teknik Informatika STMIK AMIKOM Yogyakarta)	
Perlindungan Data Terhadap Serangan Menggunakan Metoda Tebakan Pada Sistem Operasi Linux.....	9-17
Ali Mustopa (Teknik Informatika STMIK AMIKOM Yogyakarta)	
Integrasi Sistem Informasi Laboratorium Dengan Menggunakan Pendekatan <i>Service Oriented Architecture (Soa)</i> .....	18-26
Andika Agus Slameto (Teknik Informatika STMIK AMIKOM Yogyakarta)	
Analisis dan Implementasi Algoritma Kriptografi Kunci Publik Rsa dan Luc Untuk Penyandian Data.....	27-36
Bayu Setiaji (Teknik Informatika STMIK AMIKOM Yogyakarta)	
Kajian Infrastruktur Sistem Informasi Berbasis Sistem Multimedia.....	37-45
Dina Maulina (Teknik Informatika STMIK AMIKOM Yogyakarta)	
Pemanfaatan Konsep Ontology Dalam Interaksi Sistem <i>Collaborative Learning</i> .....	46-52
Emigawaty (Teknik Informatika STMIK AMIKOM Yogyakarta)	
Penerapan Algoritma <i>Learning Vector Quantization</i> Untuk Prediksi Nilai Akademis Menggunakan Instrumen Ams ( <i>Academic Motivation Scale</i> ).....	53-58
Hartatik (Teknik Informatika STMIK AMIKOM Yogyakarta)	
Perancangan Sistem Audio On Demand Berbasis Jaringan Tcp/Ip di STMIK AMIKOM Yogyakarta.....	59-67
Hastari Utama (Teknik Informatika STMIK AMIKOM Yogyakarta)	
Analisis Perbandingan Aplikasi Web Berdasarkan <i>Quality Factors</i> dan <i>Object Oriented Design Metrics</i> .....	68-78
Jamal <sup>1)</sup> , Ema Utami <sup>2)</sup> , Armadyah Amborowati <sup>3)</sup> ( <sup>1,2</sup> )Magister Teknik Informatika, <sup>3</sup> Teknik Informatika STMIK AMIKOM Yogyakarta)	
Evaluasi Sumber Daya Teknologi Informasi di SMK Negeri 3 Magelang.....	79-86
Maria Harpeni Eko Meladewi <sup>1)</sup> , Abidarin Rosidi <sup>2)</sup> , Hanif Al Fatta <sup>3)</sup> ( <sup>1, 2, 3</sup> )Magister Teknik Informatika STMIK AMIKOM Yogyakarta)	

Uji Performa Implementasi Software-Based Openflow Switch Berbasis Openwrt Pada Infrastruktur Software-Defined Network.....	87-95
Rikie Kartadie <sup>1)</sup> , Barka Satya <sup>2)</sup> ( <sup>1)</sup> Teknik Informatika, <sup>2)</sup> Manajemen Informatika STMIK AMIKOM Yogyakarta)	
Analisis Keakuratan Metode Ahp dan Metode Saw Terhadap Sistem Pendukung Keputusan Penerimaan Beasiswa .....	96-100
Saifulloh <sup>1)</sup> , Noordin Asnawi <sup>2)</sup> ( <sup>1,2)</sup> Teknik Informatika STT Dharma Iswara Madiun)	
Perbandingan Kinerja Algoritma Nbc, Svm, C 4.5 Dan Nearest Neighbor : Kasus Prediksi Status Resiko Pembiayaan Di Bank Syariah.....	101-106
Sumarni Adi (Teknik Informatika STMIK AMIKOM Yogyakarta)	

## ANALISIS PERBANDINGAN APLIKASI WEB BERDASARKAN QUALITY FACTORS DAN OBJECT ORIENTED DESIGN METRICS

Jamal<sup>1)</sup>, Ema Utami<sup>2)</sup>, Armadyah Amborowati<sup>3)</sup>

<sup>1,2)</sup>Magister Teknik Informatika STMIK AMIKOM Yogyakarta

<sup>3)</sup>Teknik Informatika STMIK AMIKOM Yogyakarta

Jl. Ring Road Utara, Condong Catur, Sleman, Yogyakarta Telp: (0274) 884201

Email : <sup>1)</sup>[jmlbpb@yahoo.com](mailto:jmlbpb@yahoo.com), <sup>2)</sup>[ema.u@amikom.ac.id](mailto:ema.u@amikom.ac.id), <sup>3)</sup>[armadyah.a@amikom.ac.id](mailto:armadyah.a@amikom.ac.id)

### Abstract

*This study was made with the purpose of comparing the quality of the software five web applications based on open source CMS e-commerce, where most good quality design model, so as to provide recommendations to web developers, businesses and lay people in building an e-commerce website. This study uses a web application prestashop, magento, woocommerce, oscommerce and openchart. Measurement of software quality metrics using traditional parameters and CK Metrics Suite. To measure quality parameters depend software using PHP tools. Traditional quantitative assessment metrics, CK Metrics Suite and software quality factors to get the best quality web applications using a combination of methods Analytical Hierarchy Process (AHP) and methods Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). Determination of the quality of software is based on two main stages, namely the first stage by using AHP. AHP is used to find the weight of traditional metrics parameters, CK Metrics Suite, software quality factors. The second stage uses TOPSIS method. TOPSIS used to search for and ranking the final value. The end result of this study indicate that web applications prestashop that has the best software quality compared with Woocommerce, OsCommerce Magento and Opencart*

**Keywords:** *traditional metrics, CK Metrics Suite, quality factors, AHP, TOPSIS*

### Pendahuluan

Tujuan dari pembuatan perangkat lunak adalah untuk menciptakan perangkat lunak yang berkualitas. Tujuan tersebut dapat dicapai dengan melakukan penilaian terhadap kualitas perangkat lunak dengan melibatkan banyak faktor. Berbagai teknik dan pola implementasi *object oriented design* memiliki sejumlah faktor-faktor yang masih berlaku umum sesuai dengan kaidah perancangan berorientasi objek. Faktor-faktor ini yang seharusnya dapat dikuantifikasi untuk menghasilkan seperangkat parameter sebagai alat ukur (*metrics*) kualitas *software*.

Beberapa penelitian sebelumnya telah menggunakan alat ukur kualitas *software* yang diwujudkan dalam beberapa parameter seperti *traditional metrics* [12,17,18], *CK Metrics Suite* [4,5,13]. Alat ukur kualitas *software* ini disebut *Object Oreinted Design (OOD) Metrics*.

Dalam penerapan *OOD Metrics*, faktor-faktor kualitas *software* memiliki parameter yang beragam, masih sulit untuk menentukan aplikasi web mana yang lebih baik, sehingga diperlukan sebuah metode untuk menggabungkan keseluruhan nilai tersebut menjadi nilai yang mempresentasikan kualitas *software* disebabkan oleh banyak faktor diantaranya domain aplikasi yang berbeda. Oleh karena, perbandingan suatu aplikasi seharusnya dilakukan dengan sejumlah aplikasi dalam domain yang sama.

Penerapan *traditional metrics, CK Metrics Suite* dan kombinasi metode AHP-TOPSIS yang diusulkan

telah diujikancobakan pada lima *open source CMS e-commerce* aplikasi web (Prestashop, Magento, Woocommerce, Oscommerce dan Opencart) dengan menggunakan *Scripting Side PHP*. Pengukuran kualitas *software* pada penelitian ini, menggunakan parameter *traditional metrics, CK Metrics Suite* dan *tools PHP Depend*. *Traditional metrics, CK metrics Suite* dipilih karena menghasil pemetaan antara *traditional metrics, CK Metrics Suite* dan faktor-faktor kualitas *software*[18]. *PHP Depend* dipilih karena sebagian besar parameter yang dimilikinya sama dengan parameter yang ada pada *traditional metrics* dan *CK Metrics Suite*.

Metrik-metrik disain berorientasi objek atau *Object Oreinted Design Metrics (OOD Metrics)* digunakan untuk mengukur kualitas internal perangkat lunak (*software*). Linda H. Rosenberg mengusulkan sembilan metrik berorieantasi objek. Metrik ini termasuk enam metrik berorientasi objek dan tiga metrik tradisional [3,17,18,21]. Parameter *traditional metrics* adalah *Cyclomatic Complexity (CC)*, *Lines Of Code (LOC)*, *Comment Percentage (CP)* dan parameter *CK Metrics Suite* adalah *Weighted Method Per Class (WMC)*, *Response For Class (RFC)*, *Lack of Cohesion Of Methods (LCOM)*, *Coupling Between Objects (CBO)*, *Depth of Inheritance Tree (DIT)*, *Number Of Children (NOC)*. *CK Metrics Suite* yaitu WMC, RFC, DIT, NOC dan CBO cocok digunakan untuk mengevaluasi kualitas *software* dan mengukur kualitas *software* berorientasi *object* pada level

class[2,20]. *Response for Class* (RFC) adalah *Number of Remote Methods* (NRM) + *Number Local Methods* (NLM). Metrik ini dikembangkan oleh Wei Li dan Sallie Henry atau dikenal dengan nama *Li & Henry Metrics* [10,12], sehingga dalam penelitian ini hanya menggunakan empat metrik dari *CK Metrics Suite* yaitu WMC, DIT, NOC dan CBO.

Secara umum parameter *traditional metrics* dan *CK Metrics Suite* memiliki kecenderungan berbanding terbalik dengan faktor-faktor kualitas *software*. Semakin kecil nilai parameter *traditional metrics* dan *CK Metrics Suite* maka semakin baik kualitas *software* tersebut [17]. Kecuali untuk parameter *Comment Percentage* (CP) semakin besar semakin baik, namun persentase komentar sekitar 30% yang paling efektif karena dapat membantu developer [17].

Pengukuran kualitas *software* dapat dilakukan dengan menggunakan salah satu dari berbagai *software quality model* yang ada. Model ISO 9126 merupakan standar internasional yang ada saat ini, model ini bersifat umum[14]. Model ini memiliki faktor-faktor kualitas *software* sebagai berikut : *functionality, reliability, usability, efficiency, maintainability* dan *portability* [14,15]. Model McCall memiliki banyak komponen penilaian. Model ini sesuai digunakan jika permasalahan utama adalah penilaian secara menyeluruh dan mendalam [14]. Model ini memiliki faktor-faktor kualitas *software* sebagai berikut ini : *correctness, reliability, efficiency, integrity, usability, maintainability, flexibility, testability, portability, Reusability* dan *Interoperability* [11,14,15]. Olsina dan rekan kerjanya, beberapa tahun yang lalu mengembangkan sebuah “Pohon penilaian kualitas” yang dapat digunakan untuk mengidentifikasi sejumlah atribut-atribut teknis yaitu : *usability, functionality, reliability, efficiency* dan *maintainability* yang selanjutnya akan digunakan sebagai panduan demi tercapainya sasaran yang berkaitan dengan kriteria-kriteria yang dapat ditetapkan pada aplikasi web yang berkualitas tinggi [15,16]. Masing-masing parameter dalam metrik memiliki kontribusi yang berbeda-beda pada faktor-faktor kualitas *software*. Tabel 1 memperlihatkan hasil pemetaan *metrics* dan faktor-faktor kualitas *software* [18].

**Tabel 1. Hubungan Software Quality Factors dan Object Oriented Design Metrics**

Faktor-faktor Kualitas Software	Parameter Metrik / Parameter Object Oriented Design Metrics	
	Traditional Metrics	CK Metrics Suite
<i>efficiency</i>	-	LCOM,CBO,DIT,NOC
<i>complexity</i>	CC	-
<i>understandability</i>	CP,LOC	WMC, RFC,DIT
<i>reusability</i>	CP,LOC	WMC,LCOM,CBO,DIT, NOC
<i>maintainability/testability</i>	CP,LOC	WMC,RFC,DIT,NOC

AHP adalah sebuah metode perbandingan berpasangan terhadap beberapa objek yang akan dievaluasi. AHP pertama kali dipublikasikan oleh T.L. Saaty dalam bukunya berjudul “ *The Analytic Hierarchy Process* “ tahun 1990 [19]. Tujuan dari AHP adalah membantu manusia dalam mengorganisir pemikiran dan penilaian untuk membuat keputusan yang lebih efektif. Salah satu kelebihan dari AHP adalah berdasar pada matriks perbandingan berpasangan dan melakukan analisis cek konsistensi. Perbandingan berpasangan merupakan bagian yang terpenting dari metode AHP, dimana pemberian perbandingan berpasangan harus dilakukan oleh yang ahli dibidangnya [22].

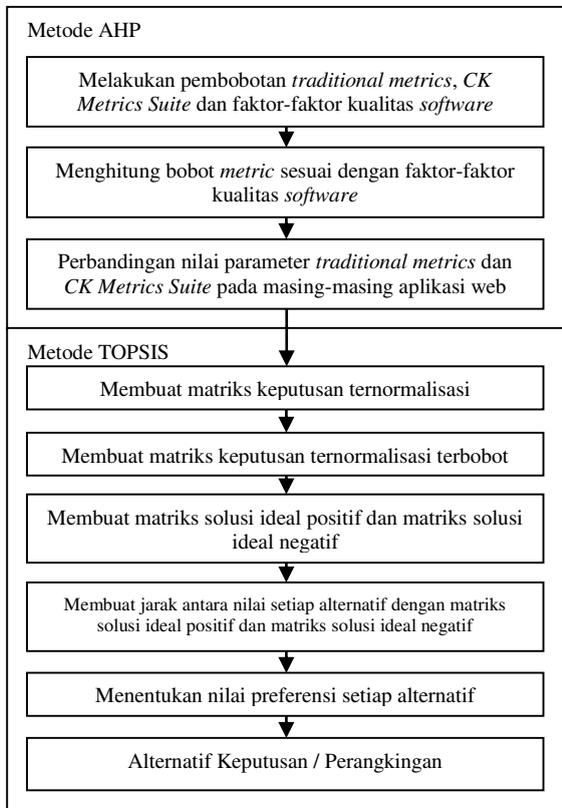
TOPSIS adalah salah satu pengambilan keputusan multikriteria, pertama kali diperkenalkan oleh Yoon dan Hwang (1981). TOPSIS pada prinsipnya bahwa alternatif yang dipilih harus mempunyai jarak yang terdekat dari solusi ideal positif dan terjauh dari ideal negatif. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis. Hal ini disebabkan karena konsepnya yang sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan mengukur kinerja relatif dari alternatif-alternatif keputusan.

## Metodologi

Hasil penelitian ini diharapkan dapat menghasilkan sebuah metode yang dapat membantu para pengguna menentukan *software* mana yang terbaik dari beberapa *software* yang dievaluasi. Pada studi kasus ini peneliti menggunakan lima aplikasi web *Open Source CMS e-commerce* yaitu Prestashop 1.6.0.4, Magento 1.7.0.2, Woocommerce 2.3.7, Oscommerce 2.3.4 dan Opencart 1.5.6. Kelima aplikasi tersebut dipilih karena sudah cukup dikenal oleh banyak pengguna dan diminati oleh banyak developer web yang terdeteksi oleh website [wappalyzer.com/categories/ecommerce](http://wappalyzer.com/categories/ecommerce). Pengukuran parameter *traditional metrics* dan *CK Metrics Suite* dilakukan dengan menggunakan *PHP depend*.

Untuk mempermudah penjelasan mengenai metode perbandingan kualitas *software*. Penulis membagi metode ke dua tahap utama yaitu tahapan pertama dengan menggunakan metode AHP dengan proses pembobotan *traditional metrics*, pembobotan *CK Metrics Suite* dan pembobotan faktor-faktor kualitas *software*. Tahapan kedua dengan menggunakan metode TOPSIS dengan proses membuat matriks keputusan yang ternormalisasi, membuat matriks keputusan yang ternormalisasi terbobot, menentukan matriks solusi ideal positif dan matriks solusi ideal negatif, menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif, menentukan nilai preferensi untuk setiap alternatif dan alternatif keputusan. Tahapan riset dijelaskan pada gambar 1. Hasil akhir dari penelitian ini diharapkan nilai

kuantitatif dari masing-masing aplikasi web yang didapatkan menjadi representasi peringkat kualitas aplikasi web tersebut.



Gambar 1. Tahapan utama metode penelitian

Pada tahapan pertama menggunakan metode AHP untuk mencari nilai bobot [1,7], pembobotan parameter *traditional metrics* dan *CK Metrics Suite* dan faktor-faktor kualitas *software* dengan tujuan untuk mendapatkan bobot masing-masing parameter tersebut. Hasil dari pembobotan parameter-parameter tersebut di cek konsistensinya, jika nilai *Consistency Ratio* < 0.1, maka hasil pembobotan dapat digunakan / diterima.

Perbandingan parameter *traditional metrics* dan *CK Metrics Suite* berpedoman pada pada skala saathy. Dalam membandingkan parameter harus mempertimbangkan objectivitas dari pengukuran. Pemetaan faktor-faktor kualitas *software* dan parameter metrik pada tabel 2 Pemetaan pada tabel tersebut yang dijadikan sebagai acuan untuk menentukan nilai perbandingan berpasangan variabel kriteria pada parameter-parameter *traditional metrics*.

Tabel 2. Pemetaan Faktor-faktor Kualitas Software dengan Parameter Metrics

Software Quality Factors	Parameter Metrics						
	Traditional Metrics			CK Metrics Suite			
	C C	LO C	C P	WM C	CB O	DI T	NO C
Efficiency					√	√	√
Complexity	√						
Understandability		√	√	√		√	
Reusability		√	√	√	√	√	√
Maintainability/testability		√	√	√		√	√

Tabel 3. Hasil Perhitungan Pembobotan Parameter Traditional Metrics

	CC	LOC	CP	Perkalian per baris (Mi)	n Akar Pangkat (Wi)	Eigen Vector (Wi)/Bobot
CC	1	1/3	1/5	0,0667	0,4055	0,1140
LOC	3	1	1/1	3,0000	1,4422	0,4054
CP	5	1	1	5,0000	1,7100	0,4806

Berdasarkan pada tabel 3. matriks perbandingan berpasangan *traditional metrics*. Maka untuk mencari nilai perkalian matriks pada masing-masing baris menggunakan persamaan (1) [6,23].

$$M_i = \prod_{j=1}^n b_{ij}, i = 1,2,\dots,n \tag{1}$$

$$M_1 = 1 \times (1/3) \times (1/5) = 0,0667$$

Untuk menghitung  $M_2, \dots, M_3$  langkahnya sama dengan  $M_1$ . Dari hasil perkalian matriks perbaris di atas, maka untuk menghitung n akar pangkat  $M_i$  menggunakan persamaan (2) [6,23].

$$\bar{w}_i = \sqrt[n]{M_i}, i = 1,2,\dots,n \tag{2}$$

$$\bar{w}_1 = \sqrt[3]{0,0667} = 0,4055$$

Untuk menghitung  $\bar{w}_2, \dots, \bar{w}_3$  langkahnya sama dengan  $\bar{w}_1$ .

Untuk melakukan normalisasi matriks, menggunakan persamaan (3) [6,23].

$$\bar{w}_i = \bar{w}_i / \sum_{j=1}^n \bar{w}_j, i = 1,2,\dots,n \tag{3}$$

$$W_1 = 0,4055 + 1,4422 + 1,7100 = 3,5577$$

$$W_1 = \frac{0,4055}{3,5577} = 0,1140$$

Untuk menghitung  $W_2, \dots, W_3$  langkahnya sama dengan mencari nilai  $W_1$ .

Untuk mencari nilai  $\lambda_{maks}$ , menggunakan persamaan (4) [6,23].

$$\lambda_{maks} = \sum_{i=1}^n \frac{n-1}{\sum_{j=1}^n \bar{w}_j}, i = 1,2,\dots,n \tag{4}$$

$$\lambda_{maks} = ((1 + 3 + 5) \times 0,1140) + ((0,3333 + 1 + 1) \times 0,4054) +$$

$$((0,2 + 1 + 1) \times 0,4806) = 3,0291$$

Untuk mencari nilai *Consistency Index* (CI), menggunakan persamaan (5) [6,19,23].

$$CI = \frac{\lambda_{maks} - n}{n - 1} \tag{5}$$

$$CI = \frac{3,0291 - 3}{3 - 1} = 0,0145$$

Untuk mencari nilai *Random Index* (RI) berdasarkan pada nilai *Random Index* (RI) pada tabel 4., Karena n = 3, maka nilai *Random Index* (RI) = 0,58

**Tabel 4.** Nilai *Random Index* (RI) [6,23]

n	1	2	3	4	5	6	7	8	9
RI	0,00	0,00	0,58	0,90	1,12	1,24	1,32	1,44	1,45

Untuk mencari nilai *Consistency Ratio* (CR), menggunakan persamaan (6) [6,19,23].

$$CR = \frac{CI}{RI} \tag{6}$$

$$CR = \frac{0,0145}{0,58} = 0,0251$$

Sesuai dengan konsep AHP, jika nilai *Consistency Ratio* (CR) < 0.1, nilai perbandingan di atas didapatkan nilai CR=0.0251, lebih kecil dari 0.1 maka pembobotan parameter di atas dapat diterima.

Langkah selanjutnya adalah menghitung pembobotan parameter *CK Metrics Suite*. Untuk menghitung faktor pembobotan *CK Metrics Suite* langkah-langkahnya sama dengan menghitung pembobotan parameter *traditional metrics*.

**Tabel 5.** Hasil Perhitungan Pembobotan Parameter *CK Metrics Suite*

	W M C	D I T	N O C	C B O	Perkalian per baris (Mi)	n Akar Pangkat ( Wi )	Eigen Vector (Wi)/ Bobot
WMC	1	1	1	3	3,0000	1,3161	0,2818
DIT	1/1	1	3	5	15,0000	1,9680	0,4214
NOC	1/1	1/3	1	3	1,0000	1,0000	0,2141
CBO	1/3	1/5	1/3	1	0,0222	0,3861	0,0827

Setelah dihasilkan *Eigen Vector* / bobot, pada masing-masing parameter *CK Metrics Suite*, maka dilakukan cek *consistency*. Pada kasus ini nilai *Lamda Maks/ λmaks* = 4,1410, Jumlah parameter 4, maka diperoleh *Consistency Index* (CI).

$$CI = \frac{4,1410 - 4}{4 - 1} = 0,0470 \tag{7}$$

Untuk mencari nilai *Random Index* (RI) berdasarkan pada nilai *Random Index* (RI) pada tabel 4., Karena n = 4, maka nilai *Random Index* (RI) = 0,90.

Untuk mencari nilai *Consistency Ratio* (CR).

$$CR = \frac{CI}{RI} = \frac{0,0470}{0,90} = 0,0522 \tag{8}$$

Sesuai dengan konsep AHP, jika nilai *Consistency Ratio* (CR) < 0.1, nilai perbandingan di atas didapatkan nilai CR=0.0522, lebih kecil dari 0.1 maka pembobotan parameter di atas dapat diterima.

Perbandingan parameter-parameter Faktor-faktor Kualitas Software berpedoman pada pada skala saathy. Dalam membandingkan parameter harus mempertimbangkan objectivitas dari pengukuran. Pemetaan perbandingan software quality model pada tabel 6. Pemetaan pada tabel tersebut yang dijadikan sebagai acuan untuk menentukan nilai perbandingan berpasangan variabel kriteria pada parameter-parameter faktor-faktor kualitas *software*, maka pemberian nilai skala dapat disesuaikan dengan nilai objectivitas pengukuran kualitas software. Berikut ini adalah tabel pemetaan perbandingan *software quality model*:

**Tabel 6.** Pemetaan Perbandingan Software Quality Models

Software Quality Factors	ISO-9126	McCall	Olsina
efficiency	√	√	√
complexity			
understandability	√		
reusability		√	
maintainability/testability	√	√	√

Dalam pemberian nilai perbandingan dengan memperhatikan penggunaan model kualitas *software*. Dari tabel 6 di atas dapat diurutkan dari faktor-faktor kualitas *software* yang paling banyak digunakan oleh *software quality models* adalah *maintainability / testability*, *efficiency*, *understandability*, *reusability* dan *complexity*.

Langkah selanjutnya adalah menghitung pembobotan parameter Faktor-faktor kualitas *software*. Untuk menghitung faktor-faktor kualitas *software* langkah-langkahnya sama dengan menghitung pembobotan parameter *CK Metrics Suite*.

**Tabel 5.** Hasil Perhitungan Pembobotan Parameter Faktor-faktor Kualitas Software

	maintainability / testability	efficiency	understandability	reusability	complexity	Perkalian per baris (Mi)	n Akar Pangkat ( W1 )	Eigen Vector (Wi)/ Bobot
maintainability / testability	1	2	5	5	7	350,0000	3,2271	0,4434
efficiency	1/2	1	5	5	7	87,5000	2,4457	0,3361
understandability	1/5	1/5	1	1	3	0,1200	0,6544	0,0899
reusability	1/5	1/5	1/1	1	3	0,1200	0,6544	0,0899
complexity	1/7	1/7	1/3	1/3	1	0,0023	0,2959	0,0407

Setelah dihasilkan *Eigen Vector* / bobot, pada masing-masing parameter Faktor-faktor Kualitas *Software*, perlu dilakukan cek *consistency*. Pada kasus ini nilai *Lamda Maks/ λmaks* = 5,1483, Jumlah parameter 5, maka diperoleh *Consistency Index* (CI).

$$CI = \frac{5,1683 - 5}{5 - 1} = 0,0421 \tag{9}$$

Untuk mencari nilai *Random Index* (RI) berdasarkan pada nilai *Random Index* (RI) pada tabel 4., Karena  $n = 5$ , maka nilai *Random Index* (RI) = 1,12.

Untuk mencari nilai *Consistency Ratio* (CR).

$$CR = \frac{CI}{RI} = \frac{0,0421}{1,1200} = 0,0376 \quad (10)$$

Sesuai dengan konsep AHP, jika nilai *Consistency Ratio* (CR) < 0.1, nilai perbandingan di atas didapatkan nilai CR=0.0376, lebih kecil dari 0.1 maka pembobotan parameter di atas dapat diterima.

Masing-masing parameter *traditional metrics* dan *CK Metrics Suite* memiliki kontribusi yang berbeda dengan faktor-faktor kualitas *software*, sehingga bobot metrik dapat dihitung pada masing-masing faktor-faktor kualitas *software* sesuai dengan tabel 6 dibawah ini :

**Tabel 6. Perhitungan Bobot Metrik (BM) pada Faktor-faktor Kualitas Software**

Faktor-faktor Kualitas Software	Bobot Parameter Traditional Metrics dan CK Metrics Suite
Maintainability / Testability	$\frac{(BM)_{CP} + (BM)_{LOC} + (BM)_{CC} + (BM)_{WMC} + (BM)_{DIT} + (BM)_{NOC}}{6}$
Efficiency	$\frac{(BM)_{CBO} + (BM)_{DIT} + (BM)_{NOC}}{3}$
Understandability	$\frac{(BM)_{CP} + (BM)_{LOC} + (BM)_{WMC} + (BM)_{DIT}}{4}$
Reusability	$\frac{(BM)_{CP} + (BM)_{LOC} + (BM)_{WMC} + (BM)_{CBO} + (BM)_{DIT} + (BM)_{NOC}}{6}$
Complexity	$(BM)_{CC}$

Untuk lebih jelasnya penulis akan memasukkan nilai bobot dari tabel 3 dan tabel 5 dengan menggunakan persamaan pada tabel 6 sehingga hasilnya dapat dilihat pada tabel 7 dibawah ini :

**Tabel 7. Hasil Perhitungan Bobot Metrik (BM) pada Faktor-faktor Kualitas Software**

Faktor-faktor Kualitas Software	Nilai Bobot Parameter Traditional Metrics dan CK Metrics Suite	Bobot Metrik (BM)
Maintainability / Testability	$(0,1140 + 0,4050 + 0,4806 + 0,2818 + 0,4214 + 0,2141) / 6$	0,3195
Efficiency	$(0,0827 + 0,4214 + 0,2141) / 3$	0,2394
Understandability	$(0,4806 + 0,4050 + 0,2818 + 0,4214) / 4$	0,3972
Reusability	$(0,4806 + 0,4050 + 0,2818 + 0,0827 + 0,4214 + 0,2141) / 6$	0,3143
Complexity	0,1140	0,1140

Nilai akhir yang dipakai untuk mencari nilai akhir dengan menggunakan metode TOPSIS untuk menentukan kualitas software dihasilkan persamaan pada tabel 8 berikut ini :

**Tabel 8. Perhitungan Bobot Akhir pada Faktor-faktor Kualitas Software**

Faktor-faktor Kualitas Software	Bobot Kualitas (BK) x Bobot Metrik (BM)
Maintainability / Testability	$(BK)_{Maintainability/Testability} \times (BM)_{Maintainability/Testability}$
Efficiency	$(BK)_{Efficiency} \times (BM)_{Efficiency}$
Understandability	$(BK)_{Understandability} \times (BM)_{Understandability}$
Reusability	$(BK)_{Reusability} \times (BM)_{Reusability}$
Complexity	$(BK)_{Complexity} \times (BM)_{Complexity}$

Untuk lebih jelasnya penulis akan memasukkan nilai dari tabel 5 dan tabel 7 dengan menggunakan persamaan yang ada pada tabel 8 sehingga hasilnya dapat dilihat pada tabel 9 dibawah ini :

**Tabel 9. Hasil Perhitungan Bobot Akhir pada Faktor-faktor Kualitas Software**

Faktor-faktor Kualitas Software	Bobot Kualitas (BK) x Bobot Metrik (BM)	Bobot Akhir
Maintainability / Testability	$(0,4434 \times 0,3195)$	0,1417
Efficiency	$(0,3361 \times 0,2394)$	0,0805
Understandability	$(0,0899 \times 0,3972)$	0,0357
Reusability	$(0,0899 \times 0,3143)$	0,0283
Complexity	$(0,0407 \times 0,1140)$	0,0046

### Hasil dan Pembahasan

Dengan menggunakan PHP Depend, hasil pengukuran nilai *traditional metrics* dan *CK Metrics Suite* dari kelima aplikasi web *Open Source CMS e-commerce* ditampilkan dalam tabel 10 dibawah ini :

**Tabel 10. Hasil Pengukuran Parameter Traditional Metrics dan CK Metrics Suite pada Masing-masing Aplikasi Web**

Nama Metrik	Parameter Metrik	Prestashop	Magento	Woocommerce	Oscommerce	Opencart
Traditional Metrics	CC	54,3472	12,6594	29,6294	23,3063	28,1069
	LOC	377,7358	110,3943	247,5175	156,3100	214,0554
	CP	43,4410	29,4527	53,3636	14,3506	3,0238
CK Metrics Suite	WMC	65,1419	13,9196	34,5035	26,5646	31,5723
	DIT	1,7009	1,8467	1,0699	0,4391	2,0337
	NOC	0,0000	0,0000	0,2203	0,0849	0,0000
	CBO	1,7686	0,8863	0,9930	0,7417	0,2693

Jumlah *class* pada masing-masing aplikasi web setelah dilakukan perhitungan dengan menggunakan *PHP Depend* dapat dilihat pada tabel 11 berikut ini:

**Tabel 11. Jumlah Class pada Masing-masing Aplikasi Web**

No.	Aplikasi Web	Jumlah Class
1	Prestashop 1.6.0.4	458
2	Magento 1.7.0.2	3086
3	Woocommerce 2.3.7	286
4	Oscommerce 2.3.4	271
5	Opencart 1.5.6	505

Setelah dilakukan pengukuran parameter *traditional metrics* dan *CK Metrics Suite* dilakukan perbandingan antara aplikasi web dengan menggunakan salah satu parameter *traditional metrics* atau *CK Metrics Suite*. Namun seperti yang dijelaskan sebelumnya karena nilai parameter berbanding terbalik dengan faktor-faktor kualitas software. Maka perbandingan tersebut perlu dilakukan *inverse* [8]. Sehingga apabila Aplikasi web-1 CC =a, Aplikasi web-2 CC =b dan Aplikasi web-3 CC =c, maka perbandingannya adalah sebagai berikut :

**Tabel 12. Perbandingan nilai Parameter metrik pada Masing-Masing Aplikasi Web**

CC	Aplikasi Web-1	Aplikasi Web-2	Aplikasi Web-3
Aplikasi Web-1	1	1/(a/b)	1/(a/c)
Aplikasi Web-2	1/(1/(a/b))	1	1/(b/c)
Aplikasi Web-3	1/(1/(a/c))	1/(1/(b/c))	1

Pada aplikasi web-1 dibandingkan dengan aplikasi web-2 adalah a/b karena faktor-faktor kualitas *software* berbanding terbalik dengan dengan nilai parameter sehingga harus di *inverse* menjadi 1/(a/b). Sedangkan untuk perbandingan aplikasi web-2 dengan aplikasi web-1 adalah kebalikan dari 1/(a/b) menjadi 1/(1/a/b) pada nilai-nilai yang lain memiliki kaidah yang sama. Pada tabel 13 perbandingan CC pada masing-masing aplikasi web, nilai-nilai berdasarkan dari tabel 10.

**Tabel 13. Perbandingan Parameter CC pada Masing-masing Aplikasi Web**

CC	Prestashop	Magento	Woocommerce	Oscommerce	Opencart
Prestashop	1	1/(54,3472 /12,6594)	1/(54,3472 /29,6294)	1/(54,3472 /23,3063)	1/(54,3472 /28,1069)
Magento	1/(1/(54,3472 /12,6594))	1	1/(12,6594 /29,6294)	1/(12,6594 /23,3063)	1/(12,6594 /28,1069)
Woo commerce	1/(1/(54,3472 /29,6294))	1/(1/(12,6594 /29,6294))	1	1/(29,6294 /23,3063)	1/(29,6294 /28,1069)
Os commerce	1/(1/(54,3472 /23,3063))	1/(1/(12,6594 /23,3063))	1/(1/(29,6294 /23,3063))	1	1/(23,3063 /28,1069)
Opencart	1/(1/(54,3472 /28,1069))	1/(1/(12,6594 /28,1069))	1/(1/(29,6294 /28,1069))	1/(1/(23,3063 /28,1069))	1

Pada tabel 14 adalah nilai-nilai perbandingan antara masing-masing aplikasi web terhadap parameter CC, nilainya berdasarkan dari tabel 13.

**Tabel 14. Hasil Perbandingan CC pada Masing-masing Aplikasi Web**

CC	Prestashop	Magento	Woocommerce	Oscommerce	Opencart
Prestashop	1	0,2329	0,5452	0,4288	0,5172
Magento	4,2930	1	2,3405	1,8410	2,2202
Woo commerce	1,8342	0,4273	1	0,7866	1,206
Os commerce	2,3319	0,5432	1,2713	1	1,206
Opencart	1,9336	0,4504	0,8292	0,8292	1

Berdasarkan pada tabel 14. matriks perbandingan parameter CC pada masing-masing aplikasi web. Maka untuk mencari nilai perkalian matriks pada masing-masing baris menggunakan persamaan (11).

$$M_i = \prod_{j=1}^n b_{ij}, i = 1,2,\dots,n \tag{11}$$

$$M_1 = 1 \times 0,2329 \times 0,5452 \times 0,4288 \times 0,5172 = 0,0282$$

Untuk menghitung  $M_2, \dots, M_5$  langkahnya sama dengan  $M_1$ . Dari hasil perkalian matriks perbaris di atas, maka untuk menghitung n akar pangkat  $M_i$  menggunakan persamaan (12).

$$\bar{w}_i = \sqrt[n]{M_i}, i = 1,2,\dots,n \tag{12}$$

$$\bar{w}_1 = \sqrt[5]{0,0282} = 0,4897$$

Untuk menghitung  $\bar{w}_2, \dots, \bar{w}_5$  langkahnya sama dengan  $\bar{w}_1$ .

Untuk melakukan normalisasi matriks, menggunakan persamaan (13).

$$\bar{w}_i = \bar{w}_i / \sum_{j=1}^n \bar{w}_j, i = 1,2,\dots,n \tag{13}$$

$$W_1 = 0,4897 + 2,1024 + 0,9424 + 1,1419 + 0,9025 = 5,5790$$

$$w_1 = \frac{0,4897}{5,5790} = 0,0878$$

Untuk menghitung  $W_2, \dots, W_5$  langkahnya sama dengan mencari nilai  $W_j$ . Hasil secara keseluruhan dapat dilihat pada tabel 15 berikut ini :

**Tabel 15. Hasil Evaluasi Perbandingan Nilai Parameter CC pada Masing-Masing Aplikasi Web**

CC	Perkalian per Baris (Mi)	n akar pangkat ( $\frac{n}{w_1}$ )	Eigen Vector (Wi)/ Faktor Evaluasi
Prestashop	0,0282	0,4897	0,0878
Magento	41,0709	2,1024	0,3768
Woocommerce	0,7434	0,9424	0,1689
Oscommerce	1,9419	1,1419	0,2047
Opencart	0,5988	0,9025	0,1618

Pada tabel 15 di atas dibandingkan 5 aplikasi web. Pada faktor evaluasi CC aplikasi web Magento mempunyai nilai tertinggi. Hal ini menandakan bahwa aplikasi web Magento mempunyai kompleksitas *algoritma* pada *method* yang tinggi. Kemudian urutan kedua sampai ke lima yaitu Oscommerce, wocommerce, Opencart, Prestashop.

Langkah yang sama pada proses perbandingan CC, dilakukan untuk proses perbandingan parameter LOC pada masing-masing aplikasi web, sehingga didapatkan hasil evaluasi perbandingan nilai parameter LOC pada masing-masing aplikasi web.

**Tabel 16. Hasil Evaluasi Perbandingan Nilai Parameter LOC pada Masing-Masing Aplikasi Web**

LOC	Faktor Evaluasi
Prestashop	0,0988
Magento	0,3381
Woocommerce	0,1653
Oscommerce	0,2388
Opencart	0,1590

Pada tabel 16 di atas dibandingkan 5 aplikasi web . Pada faktor evaluasi LOC aplikasi web Magento mempunyai nilai tertinggi. Hal ini menandakan bahwa aplikasi web Magento mempunyai size yang tinggi. Size / jumlah baris yang terlalu tinggi akan berakibat bertambah besarnya jumlah file. Kemudian urutan kedua sampai ke lima yaitu Oscommerce, Woocommerce, Opencart, Prestashop.

Langkah yang sama pada proses perbandingan CC, dilakukan untuk proses perbandingan parameter CP pada masing-masing aplikasi web, tetapi sesuai dengan penjelasan sebelumnya bahwa nilai parameter CP semakin besar semakin baik, sehingga tidak dilakukan *invers*. Hasil evaluasi perbandingan nilai parameter CP pada masing-masing aplikasi web dapat dilihat pada tabel 17 di bawah ini:

**Tabel 17. Hasil Evaluasi Perbandingan Nilai Parameter CP pada Masing-Masing Aplikasi Web**

CP	Faktor Evaluasi
Prestashop	0,3286
Magento	0,2228
Woocommerce	0,3104
Oscommerce	0,1085
Opencart	0,0297

Pada tabel 17 di atas dibandingkan 5 aplikasi web. Hasil evaluasi parameter CP pada masing-masing aplikasi web Prestashop mempunyai nilai tertinggi. Hal ini menandakan bahwa aplikasi web Prestashop mempunyai komentar paling baik. Kemudian urutan kedua sampai ke lima yaitu Woocommerce, Magento, Oscommerce dan Opencart.

Langkah yang sama pada proses perbandingan CC, dilakukan untuk proses perbandingan parameter WMC pada masing-masing aplikasi web, sehingga didapatkan hasil evaluasi perbandingan nilai parameter WMC pada masing-masing aplikasi web.

**Tabel 18. Hasil Evaluasi Perbandingan Nilai Parameter WMC pada Masing-Masing Aplikasi Web**

WMC	Faktor Evaluasi
Prestashop	0,0828
Magento	0,3874
Woocommerce	0,1647
Oscommerce	0,2030
Opencart	0,1621

Pada tabel 18 di atas dibandingkan 5 aplikasi web . Pada faktor evaluasi WMC Magento mempunyai nilai tertinggi. Hal tersebut menunjukkan bahwa nilai WMC tinggi mempunyai kecenderungan kegagalan *software*. Kemudian disusul dengan urutan kedua sampai kelima yaitu Oscommerce, Wocommerce, Opencart, Prestashop.

Langkah yang sama pada proses perbandingan CC, dilakukan untuk proses perbandingan parameter DIT pada masing-masing aplikasi web, sehingga didapatkan hasil evaluasi perbandingan nilai parameter DIT pada masing-masing aplikasi web pada tabel 19 dibawah ini.

**Tabel 19. Hasil Evaluasi Perbandingan Nilai Parameter DIT pada Masing-Masing Aplikasi Web**

DIT	Faktor Evaluasi
Prestashop	0,1191
Magento	0,1097
Woocommerce	0,2263
Oscommerce	0,4615
Opencart	0,0834

Pada tabel 19 di atas dibandingkan 5 aplikasi web. Pada faktor evaluasi DIT aplikasi web Oscommerce mempunyai nilai tertinggi. Hal ini menandakan bahwa kedalaman hirarki aplikasi Oscommerce paling besar, sehingga semakin tinggi kompleksitasnya. Kemudian urutan kedua sampai ke lima yaitu Woocommerce, Prestashop, Magento, Opencart.

Langkah yang sama pada proses perbandingan CC, dilakukan untuk proses perbandingan parameter CBO pada masing-masing aplikasi web, sehingga didapatkan hasil evaluasi perbandingan nilai parameter CBO pada masing-masing aplikasi web pada tabel dibawah ini.

**Tabel 20. Hasil Evaluasi Perbandingan Nilai Parameter CBO pada Masing-Masing Aplikasi Web**

CBO	Faktor Evaluasi
Prestashop	0,0743
Magento	0,1482
Woocommerce	0,1402
Oscommerce	0,1771
Opencart	0,4601

Pada tabel 20 di atas dibandingkan 5 aplikasi web. Pada faktor evaluasi CBO aplikasi web Opencart mempunyai nilai tertinggi. Hal ini menandakan bahwa Opencart banyak memiliki banyak *class* yang saling ketergantungan antara satu *class* dengan *class* lainnya, sehingga menunjukkan kurang baik untuk sebuah aplikasi. Kemudian urutan kedua sampai ke lima yaitu Oscommerce, Magento, Woocommerce dan Prestashop.

Pada penelitian ini menggunakan kombinasi antara metode AHP dan TOPSIS. Proses pembobotan parameter *traditional metrics*, proses pembobotan parameter *CK Metrics Suite* dan proses pembobotan faktor-faktor kualitas *software* menggunakan metode AHP. Proses selanjutnya yaitu mencari nilai akhir alternatif keputusan /perangkingan menggunakan metode TOPSIS [1,7]. Data awal dari hasil pengukuran parameter *traditional metrics* dan *CK Metrics Suite* pada Masing-masing Aplikasi Web.

54,3472	12,6594	29,6294	23,3063	28,1069
377,7358	110,3943	247,5175	156,3100	214,0554
43,4410	29,4527	53,3636	14,3506	3,0238
65,1419	13,9196	34,5035	26,5646	31,5723
1,7009	1,8467	1,0699	0,4391	2,0337
0,0000	0,0000	0,2203	0,0849	0,0000
1,7686	0,8863	0,9930	0,7417	0,2693

TOPSIS membutuhkan rating kinerja untuk setiap alternatif pada setiap kriteria yang ternormalisasi sesuai persamaan (14) [9].

$$rij = \frac{Xij}{\sqrt{\sum_{i=0}^m Xij}}, i = 1,2,\dots,m; j = 1,2,\dots,n \quad (14)$$

Perhitungan untuk mendapat total  $\sum XiI^2$  adalah :

$$\sum_{i=1}^7 Xij = ((54,3472^2) + (377,7358^2) + (43,4410^2) + (65,1419)^2 + (1,7009^2) + (0,0000^2) + (1,7686^2) = 151.775,5611$$

Cara yang sama untuk mendapatkan total  $\sum Xi2^2, \dots, \sum Xi5^2$ .

$$rij = \frac{54,3472}{\sqrt{151.744,5611}} = 0,1395$$

Cara yang sama dilakukan untuk mendapatkan kinerja rating untuk :  $r_{21} \dots r_{51}$  ;  $r_{12} \dots r_{52}$  ; ... ;  $r_{15} \dots r_{55}$

Hasil keseluruhan matriks keputusan yang ternormalisasi dapat dilihat pada tabel 21.

**Tabel 21. Matriks Keputusan yang Ternormalisasi**

0,1395	0,1093	0,1152	0,1449	0,1288
0,9696	0,9532	0,9621	0,9715	0,9809
0,1115	0,2543	0,2074	0,0892	0,0139
0,1672	0,1202	0,1341	0,1651	0,1447
0,0044	0,0159	0,0042	0,0027	0,0093
0,0000	0,0000	0,0009	0,0005	0,0000
0,0045	0,0077	0,0039	0,0046	0,0012

Sebagaimana yang telah dijelaskan sebelumnya bahwa masing-masing parameter *traditional metrics* dan *CK Metrics Suite* memiliki kontribusi yang berbeda-beda terhadap faktor-faktor kualitas *software*, sehingga pada masing-masing faktor-faktor kualitas *software* dihitung sesuai dengan kontribusinya masing-masing dengan menggunakan persamaan lihat tabel 6. Berikut ini menghitung faktor-faktor kualitas *software* pada aplikasi web Prestashop :

$$\begin{aligned} \text{Maintainability/} &= ((BM)_{CP} + (BM)_{LOC} + (BM)_{CC} + (BM)_{WMC} + \\ \text{testability} & (BM)_{DIT} + (BM)_{NOC})/6 \\ &= (0,1115 + 0,9696 + 0,1395 + 0,1672 + \\ & 0,0044 + 0,0000)/6 \\ &= \mathbf{0,2320} \\ \text{Efficiency} &= ((BM)_{CBO} + (BM)_{DIT} + (BM)_{NOC})/3 \\ &= (0,0045 + 0,0044 + 0,0000)/3 \\ &= \mathbf{0,0030} \\ \text{Understandability} &= ((BM)_{CP} + (BM)_{LOC} + (BM)_{WMC} + (BM)_{DIT})/4 \\ &= (0,1115 + 0,9696 + 0,1672 + 0,0044)/4 \\ &= \mathbf{0,3132} \\ \text{Reusability} &= ((BM)_{CP} + (BM)_{LOC} + (BM)_{WMC} + (BM)_{CBO} + \\ & (BM)_{DIT} + (BM)_{NOC})/6 \\ &= (0,1115 + 0,9696 + 0,1672 + 0,0045 + \\ & 0,0044 + 0,0000)/6 \\ &= \mathbf{0,2095} \\ \text{Complexity} &= (BM)_{CC} \\ &= \mathbf{0,1395} \end{aligned}$$

Hasil selengkapnya dapat dilihat pada tabel 22 pada kolom Prestashop. Cara yang sama yang dilakukan untuk menghitung faktor-faktor kualitas *software* aplikasi web lainnya yaitu megento, Woocommerce, Oscommerce dan Opencart.

**Tabel 22. Bobot Faktor-faktor Kualitas Software pada Masing-masing Aplikasi Web**

	Prestashop	Magento	Woocommerce	Oscommerce	Opencart
Maintainability / Testability	0,2320	0,2422	0,2373	0,2290	0,2129
Efficiency	0,0030	0,0079	0,0030	0,0026	0,0035
Understandability	0,3132	0,3359	0,3270	0,3071	0,2872
Reusability	0,2095	0,2252	0,2188	0,2056	0,1917
Complexity	0,1395	0,1093	0,1152	0,1449	0,1288

Membuat matriks keputusan yang ternormalisasi terbobot, dengan bobot  $W = (W_1, W_2, \dots, W_n)$  adalah bobot parameter kriteria (eigen vector)

yang didapatkan pada proses perhitungan metode AHP, dimana  $W = (0,1417; 0,0805; 0,0357; 0,0283; 0,0046)$ .

Hasil keputusan yang ternormalisasi terbobot  $Y_{11}$  adalah sebagai berikut ini ( dapat dilihat pada tabel 3.43, pada bagian yang diarsir) :

$$Y = w_1 \times r_{11} = 0,1417 \times 0,2320 = 0,0329$$

Cara yang sama dilakukan untuk mendapatkan nilai :  $Y_{21} \dots Y_{51}; Y_{12} \dots Y_{52}; \dots; Y_{15} \dots Y_{55}$

Hasil keseluruhan matriks keputusan yang ternormalisasi terbobot dapat dilihat pada tabel 23 dibawah ini :

**Tabel 23. Matriks Keputusan yang Ternormalisasi Terbobot**

0,0329	0,0343	0,0336	0,0324	0,0302
0,0002	0,0006	0,0002	0,0002	0,0003
0,0112	0,0120	0,0117	0,0110	0,0103
0,0059	0,0064	0,0062	0,0058	0,0054
0,0006	0,0005	0,0005	0,0007	0,0006

Menentukan matriks solusi ideal positif dan matriks solusi negatif. Solusi ideal positif dinotasikan dengan  $A^+$  dan solusi ideal negatif dinotasikan dengan  $A^-$ . Untuk mencari solusi ideal positif adalah sebagai berikut :

$$Y_1^+ = \text{Max}(0,0329; 0,0002; 0,0112; 0,0059; 0,0006) = 0,0329$$

Cara yang sama dilakukan untuk mendapatkan nilai  $Y_2^+, \dots, Y_5^+$ , sehingga matriks solusi ideal positif adalah sebagai berikut :

$$A^+ = [ 0,0329 ; 0,0343 ; 0,0336 ; 0,0324 ; 0,0302 ]$$

Untuk mencari solusi ideal negatif adalah sebagai berikut :

$$Y_1^- = \text{Max}(0,0329; 0,0002; 0,0112; 0,0059; 0,0006) = 0,0002$$

Cara yang sama dilakukan untuk mendapatkan nilai  $Y_2^-, \dots, Y_5^-$ , sehingga matriks solusi ideal negatif adalah sebagai berikut :

$$A^- = [ 0,0002 ; 0,0005 ; 0,0002 ; 0,0002 ; 0,0003 ]$$

Menghitung jarak antara alternatif dengan solusi ideal positif, dengan menggunakan persamaan (15) berikut ini [9]:

$$D_i^+ = \sqrt{\sum_{j=1}^n (Y_j^+ - Y_{ij})^2}, i = 1,2,\dots,m \quad (15)$$

$$D_1^+ = \sqrt{(0,0329 - 0,0329)^2 + (0,0343 - 0,0343)^2 + (0,0336 - 0,0336)^2 + (0,0324 - 0,0324)^2 + (0,0302 - 0,0302)^2}$$

$$D_1^+ = 0,0000$$

Cara yang sama dilakukan untuk mendapatkan nilai  $D_3^+, \dots, D_5^+$ . Hasil keseluruhan dapat dilihat pada

tabel 24. Menghitung jarak antara alternatif dengan solusi ideal negatif, dengan menggunakan persamaan (16) berikut ini [9]:

$$D_i^- = \sqrt{\sum_{j=1}^n (Y_{ij}^- - Y_j^-)^2}, i = 1,2,\dots,m \quad (16)$$

$$D_1^- = \sqrt{(0,0329 - 0,0002)^2 + (0,0343 - 0,0005)^2 + (0,0336 - 0,0002)^2 + (0,0324 - 0,002)^2 + (0,0302 - 0,0003)^2}$$

$$D_1^- = 0,07247$$

Cara yang sama dilakukan untuk mendapatkan nilai  $D_3^-, \dots, D_5^-$ . Hasil keseluruhan dapat dilihat pada tabel 24. Untuk menentukan nilai preferensi untuk setiap alternatif ( $V_i$ ), menggunakan persamaan (17) adalah sebagai berikut [9]:

$$V_i = \frac{D_i^-}{D_i^+ + D_i^-}, i = 1,2,\dots,m \quad (17)$$

$$V_1 = \frac{0,07247}{0,07247 + 0,0000} = 1,00000$$

Cara yang sama dilakukan untuk mendapatkan nilai  $V_2 \dots V_5$ . Hasil keseluruhan dapat dilihat pada tabel 24.

**Tabel 24. Hasil Preferensi dengan Metode TOPSIS**

Nama Aplikasi Web	$D^+$	$D^-$	$V_i$
Prestashop	0,00000	0,07247	1,00000
Magento	0,07242	0,00013	0,00173
Woocommerce	0,04803	0,02445	0,33732
Oscommerce	0,05984	0,01263	0,17429
Opencart	0,07182	0,00075	0,01037

Pada tabel 24 adalah hasil preferensi dengan menggunakan metode TOPSIS dari masing-masing aplikasi web.

**Tabel 25. Hasil Perangkingan**

Nama Aplikasi Web	$V_i$	Rangking
Prestashop	1,00000	1
Woocommerce	0,33732	2
Oscommerce	0,17429	3
Opencart	0,01037	4
Magento	0,00173	5

Dari tabel 25 hasil perangkingan, maka dapat disimpulkan bahwa sesuai dengan konsep TOPSIS bahwa semakin tinggi nilai  $V_i$ , maka semakin tinggi / baik preferensi untuk alternatif tersebut. Dari aspek *object oriented desain model* dengan menggunakan *traditional metrics, CK Metrics Suite* dan faktor-faktor kualitas desain *software*, aplikasi web Prestashop dengan nilai 1,00000 yang paling baik kualitasnya dari aplikasi web lainnya. Sedangkan yang paling renda adalah aplikasi web Magento

dengan nilai 0,00173. Disusul dengan urutan kedua, ketiga dan keempat adalah aplikasi web Woocommerce dengan nilai 0,33732, Oscommerce dengan nilai 0,17429 dan Opencart dengan nilai 0,01037.

Berdasarkan hasil analisis dan pengukuran kualitas *software* aplikasi web (Prestashop, Magento, Woocommerce, Oscommerce dan Opencart), maka diperoleh beberapa rekomendasi untuk 3 stakeholder. Adapun rekomendasi tersebut adalah sebagai berikut ini :

1. Hasil evaluasi parameter CBO pada masing-masing aplikasi web, Opencart mempunyai nilai tertinggi. Hal ini menandakan bahwa Opencart memiliki banyak *class* yang saling ketergantungan antara satu *class* dengan *class* lainnya, sehingga menunjukkan kurang baik untuk sebuah aplikasi. Hal ini juga menyebabkan ketidakkonsisten tingkat *interdependency* antara modul pada sebuah aplikasi.
2. Hasil evaluasi parameter WMC pada masing-masing aplikasi web. Magento mempunyai nilai tertinggi. Hal tersebut menunjukkan bahwa nilai WMC tinggi mempunyai kecenderungan kegagalan *software*.
3. Hasil evaluasi parameter DIT pada masing-masing aplikasi web. Oscommerce memiliki nilai tertinggi. Hal ini menandakan bahwa kedalaman hirarki aplikasi Oscommerce paling besar, sehingga tingkat kompleksitasnya tinggi.
4. Hasil evaluasi parameter LOC pada masing-masing aplikasi web. Prestashop mempunyai nilai tertinggi. Hal ini menandakan bahwa aplikasi web Prestashop mempunyai size yang tinggi. Jumlah baris/size yang tinggi akan mengakibatkan bertambahnya jumlah file.
5. Hasil evaluasi parameter DIT dan LOC. Woocommerce memiliki menempati urutan kedua. Hal ini menandakan bahwa tingkat kompleksitas dan size agak tinggi perlu dievaluasi kembali.
6. Direkomendasikan kepada pelaku bisnis untuk menggunakan Prestashop, karena paling baik kualitasnya diantara aplikasi web e-commerce (Magento, Woocommerce, Oscommerce dan Opencart).
7. Untuk orang Awam jika tujuannya untuk pembelajaran programming disarankan menggunakan Prestashop dan Woocommerce, karena memiliki komentar yang lebih lengkap. Jika tujuannya untuk bisnis direkomendasikan menggunakan Prestashop, karena kualitasnya paling baik.

Bagi yang sudah pernah menggunakan CMS sebelumnya misalnya CMS Wordpress disarankan dapat menggunakan Woocommerce karena Woocommerce adalah pengembangan dari CMS Wordpress.

## Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, maka dapat disimpulkan bahwa kualitas aplikasi web Prestashop yang paling baik kualitasnya dari aplikasi web lainnya karena memiliki nilai preferensi yang paling tinggi yaitu 1,00000. Sesuai dengan konsep TOPSIS bahwa semakin tinggi nilai preferensi, maka semakin tinggi / baik alternatif (aplikasi) tersebut. Sedangkan yang paling rendah tingkat kualitasnya adalah aplikasi web Magento dengan nilai 0,00173.

Hasil evaluasi parameter WMC pada tabel 18 aplikasi web Magento mempunyai nilai tertinggi. Hal tersebut menunjukkan bahwa nilai WMC tinggi mempunyai kecenderungan kegagalan *software*. Kemudian disusul dengan urutan kedua sampai kelima yaitu Oscommerce, Woocommerce, Opencart, Prestashop.

Pada tabel 19 dibandingkan 5 aplikasi web . Hasil evaluasinya adalah bahwa nilai parameter DIT pada aplikasi web Oscommerce mempunyai nilai tertinggi. Hal ini menandakan bahwa kedalaman hirarki aplikasi Oscommerce paling besar, sehingga semakin tinggi kompleksitasnya. Kemudian urutan kedua sampai ke lima yaitu Woocommerce, Prestashop, Magento, Opencart.

Pada tabel 20 dibandingkan 5 aplikasi web. Hasil evaluasi parameter CBO pada aplikasi web Opencart mempunyai nilai tertinggi. Hal ini menandakan bahwa Opencart banyak memiliki banyak *class* yang saling ketergantungan antara satu *class* dengan *class* lainnya, sehingga menunjukkan kurang baik untuk sebuah aplikasi.

Pada tabel 15 dibandingkan 5 aplikasi web. Pada faktor evaluasi CC aplikasi web Prestashop mempunyai nilai terkecil. Hal ini menandakan bahwa aplikasi web Prestashop mempunyai kompleksitas algoritma pada method yang kecil, Cyclomatic Complexity yang rendah baik untuk sebuah aplikasi. Kemudian urutan kedua sampai ke lima yaitu Opencart, woocommerce, Oscommerce, Magento.

Pada tabel 16 hasil evaluasi parameter LOC aplikasi web Magento mempunyai nilai tertinggi. Hal ini menandakan bahwa aplikasi web Magento mempunyai jumlah baris yang banyak. Jumlah baris yang terlalu banyak kurang bagus untuk sebuah aplikasi karena akan menyebabkan besarnya ukuran file pada sebuah aplikasi. Kemudian urutan kedua sampai ke lima yaitu Oscommerce, Woocommerce, Opencart, Prestashop.

Berdasarkan hasil evaluasi pada tabel 17 dibandingkan 5 aplikasi web . bahwa nilai parameter CP pada aplikasi web Prestashop mempunyai nilai tertinggi. Hal ini menandakan bahwa aplikasi web Prestashop memiliki komentar yang lengkap, sehingga cukup baik untuk developer web dan orang awam. Kemudian urutan kedua sampai ke lima yaitu

Woocommerce, Magento, Oscommerce dan Opencart.

Riset lanjutan perlu dikembangkan dengan mengimplementasi *object oriented design metric* seperti *MOOD metrics suite* atau penggabungan metrik yang lain misalnya *Li & Henry metric*. Menggunakan kombinasi metode yang lain selain AHP dengan TOPSIS dan melibatkan banyak faktor yang evaluasi.

## Daftar Pustaka

- [1] Al Maliki, A.; Owen, G.; Bruce, D., 2012, *Combining AHP and TOPSIS Approaches to Support Site Selection for a Lead Pollution Study*, 2nd International Conference on Environmental and Agriculture Engineering IPCBEE, 37, 1-8.
- [2] Bansal, M.; Agrawal, P., C., 2014, *Critical Analysis of Object Oriented Metrics in Software Development*, Conference Publishing Services (CPS), 978-1-4799-4910/6/14, IEEE, Juni 2014.
- [3] Chauhan, R.; Singh, R.; Saraswat, A.; Joya, A., H.; Gunjan, V., K., 2014, *Estimasi of Software Quality using Object Oriented Design Metrics*, International Journal of Innovative Research in Computer and Communication Engineering, An ISO 3297:2007 Certified Organization, vol.2, issue 1, January 2014.
- [4] Chidamber, S.R.; Kemerer, F., C., 1991, *Towards a Metrics Suite for Object Oriented Design*, In Proceeding Sixth OOPSLA Conference, pp. 197-211
- [5] Chidamber, S.R.; Kemerer, F., C., 1994, *A Metrics Suite for Object Oriented Design*, IEEE Transaction on Software Engineering, Vol.20, No.6, Juni 1994.
- [6] Chun, Z.J.; Huang, Y., Y.; Wang, Z.Q., 2009, *Topsis-AHP-Simulation Method and its Application in Operational Capability Evaluation*, IEEE Journal & Magazine, 978-1-4244-2723-9/09.
- [7] Dong, C., 2011, *Innovation Capacity Appraisal of Junior College Engineering Students Based on AHP-TOPSIS*, IEEE Journal & Magazine, 979-1-61284-486-2/11.
- [8] Hermawan, E.; Mursanto, P., 2009, *Pemeringkatan Software Aplikasi Berdasarkan Properti Kualitas Disain dan Metric For Object Oriented Software menggunakan Analytic Hierarchy Process*, Journal of Information System, Volume 5, Issues 1, April 2009.
- [9] Jiang, C.Z.; Yan, Z., 2010, *Application of TOPSIS Analysis Method Based on AHP in Bid Evaluation of Power Equipment*, IEEE, 978-1-4244-6928-4/10, April 2010.
- [10] Li, W.; Henry, S., 1993, *Maintenance Metrics for the Object Oriented Paradigm*, IEEE, 0-8186-3740-4/93, April 1993.
- [11] McCall, J.; Richards, P.; Walters, G., 1977, *Factors in Software Quality*, Three Volumes, NTIS AD-A049,015, November 1977.
- [12] McNinch, A., C., 2012, *Measuring and Quantifying Web Application Design*, Tesis, Master of Science In Computer Science, University of Montana Missoula
- [13] Mishra, A.; Dubey, K., A., 2014, *Evaluation of Reliability of Object Oriented Software System Using Fuzzy Approach*, IEEE, 978-1-4799-4236-7/14, Juli 2014.
- [14] Parwita, S.G.W.; Putri, R. A. A. L., 2014, *Komponen Penilaian Kualitas Perangkat Lunak Berdasarkan Software Quality Models*, Seminar Nasional Teknologi Informasi & Komunikasi Terapan, ISBN 979-26-0255-0, 23 Juni 2012.
- [15] Pressman, R. S., 2010, *Software Engineering A Practitioner's Approach*, Seventh Edition, McGraw Hill, New York.
- [16] Pressman, R. S.; Lowe, D., 2009, *Web Engineering A Practitioner's Approach*, McGraw Hill, New York.
- [17] Rosenberg, H. L., 1998, *Applying and Interpreting Object Oriented Metrics*, Proceedings of Software Technology Conference, Utah
- [18] Rosenberg, H.L.; Hyatt, E. L., 2001, *Software Quality Metrics for Object-Oriented Environments*, NASA Technical Report SATC, no.1, pp 11-58.
- [19] Saaty, T.L., 1990, *How to make a decision : The Analytic Hierarchy Process*, European Journal of Operational Research, 0377-2221/90, Elsevier Science Publisher B.V., North Holland.
- [20] Sharma, A. K. ; Kalia, A. ; Singh, H., 2012, *Metrics Identification for Measuring Object Oriented Software Quality*, International Journal of Soft Computing and Engineering, (IJSCE) ISSN: 2231-2307, Volume-2, Issue-5, November 2012.
- [21] Srivastava, S. ; Kumar, R., 2013, *Inderck Method Measure Software Quality Using CK-OO Suite*, ISSP, 978-1-4799-7/13, IEEE, Juli 2013.
- [22] Wu, W.; Kou, G. ; Peng, Y., 2012, *Credit Risk Evaluation by Improved MCDM Models*, 2012 Fifth International Conference on Business Intelligence and Financial Engineering, 191-195