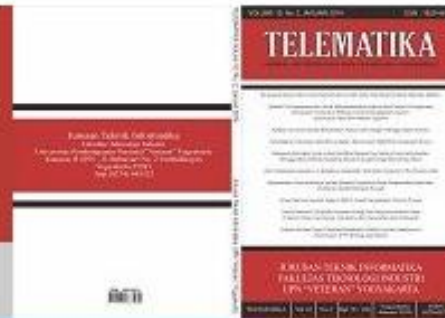


TELEMATIKA

JURNAL INFORMATIKA DAN TEKNOLOGI INFORMASI



[HOME](#) | [ABOUT](#) | [LOGIN](#) | [REGISTER](#) | [SEARCH](#) | [CURRENT](#) | [ARCHIVES](#) | [ANNOUNCEMENTS](#) | [AIMS AND SCOPE](#) | [AUTHOR GUIDELINES & SUBMIS](#)

Home > [Vol 14, No 2 \(2017\)](#)

Telematika

Telematika diterbitkan sejak Juli 2004 oleh Fakultas Teknologi Industri Jurusan Teknik Informatika UPN "Veteran" Yogyakarta sebagai media untuk menyalurkan pemahaman tentang informasi yang berkembang saat ini, baik berupa hasil penelitian lapangan atau laboratorium maupun studi pustaka. Jurnal Telematika ini terbit dua kali dalam satu tahun yaitu di bulan Januari dan Juli.

Redaksi menerima sumbangan naskah dari dosen, peneliti, mahasiswa maupun praktisi dengan ketentuan penulisan seperti tercantum pada halaman dalam sampul belakang pada jurnal ini



Announcements

No announcements have been published.

[More Announcements...](#)

[Home](#) > [About the Journal](#) > [Editorial Team](#)

Editorial Team

Ketua Penyunting

Heru Cahya Rustamaji, Department of Informatics Engineering Faculty of Industrial Engineering University Pembangunan Nasional, Indonesia

Dewan Penyunting

Hidayatulah Himawan, Department of Informatics Engineering Faculty of Industrial Engineering University Pembangunan Nasional, Indonesia

Bambang Yuwono, Department of Informatics Engineering Faculty of Industrial Engineering University Pembangunan Nasional, Indonesia

Dr. Awang Hendrianto Pratomo, Department of Informatics Engineering Faculty of Industrial Engineering University Pembangunan Nasional "Veteran" Yogyakarta, Indonesia

S.Kom., M.Eng. Oliver Samuel Simanjuntak, Jurusan Teknik Informatika, Fakultas Teknologi Industri, UPN "Veteran" Yogyakarta, Indonesia

Dessyanto Boedi Prasetyo, Department of Informatics Engineering Faculty of Industrial Engineering University Pembangunan Nasional, Indonesia

Dr. Herlina Jayadianti, Department of Informatics Engineering Faculty of Industrial Engineering University Pembangunan Nasional



Telematika by <http://jurnal.upnyk.ac.id/index.php/telematika/index/> is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#) (CC BY-NC-SA 4.0) **00049463**


Status Kunjungan Jurnal Telematika

Table of Contents

General


STRATEGI CUSTOMER RELATIONSHIP MANAGEMENT UNTUK MENINGKATKAN LOYALITAS PELANGGAN DAN VOLUME PENJUALAN MENGGUNAKAN TEKNIK CLUSTERING K-MEANS

75-82 

 *Arfi Joyendri*


PENGEMBANGAN DETEKSI PENYUSUPAN MENGGUNAKAN MULTIAGENT



 *Bambang Sugiantoro*


SISTEM INFORMASI PEMERINTAHAN DESA MENGGUNAKAN FRAMEWORK ZACHMAN STUDI KASUS DESA TRIWIDADI, PAJANGAN, BANTUL



 *Aulia Pasca Sahida, Heru Cahya Rustamaji, Hafsa Hafsa*

PENGIDENTIFIKASIAN EXTRACT CLASS REFACTORING UNTUK MENINGKATKAN NILAI COHESION CLASS: SYSTEMATIC LITERATURE REVIEW




 *Mohammad Farid Naufal*

Robotics

PENERAPAN PREDIKSI BOLA PADA STRATEGI PENYERANGAN ROBOT SEPAKBOLA UPN VETERAN YOGYAKARTA




 *Awang Hendrianto Pratomo, Balza Ahmad, Tejo Pramono*

Artificial Intelligence

MODEL PREDIKSI RENTET WAKTU NEURAL NETWORK BERBASIS PARTICLE SWARM OPTIMIZATION UNTUK PREDIKSI HARGA SAHAM



 *Andri Pramuntadi*

PENENTUAN JUMLAH UKURAN PAKAIAN OPTIMAL SEBAGAI RANCANGAN SISTEM UKURAN PAKAIAN ANAK LAKI-LAKI DI INDONESIA DENGAN ANALISIS KESEIMBANGAN DAN FUZZY C MEANS BERBASIS ARTIFICIAL BEE COLONY



 *Novenda Kartika Putrianto*

PENGIDENTIFIKASIAN EXTRACT CLASS REFACTORING UNTUK MENINGKATKAN NILAI COHESION CLASS: SYSTEMATIC LITERATURE REVIEW

Mohammad Farid Naufal

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Surabaya
Jl. Raya Kali Rungkut, Surabaya, Jawa Timur 60293
e-mail : faridnaufal@staff.ubaya.ac.id

Abstract

Context: Cohesion is one factor that is taken into account in assessing the quality of a software that uses the basic Object Oriented Programming (OOP). The larger value of cohesion make a class more independent and more easy to do maintenance when software evolve. In OOP development the class is expected to have high cohesion values.

Objective: This paper do a systematic literature review related to one of the refactoring techniques namely extract class refactoring. It is a way to increase the value of cohesion of a class. This paper also do a systematic literature review related to class cohesion metrics for assess the level of class complexity.

Method: This paper do a systematic literature review on two journal databases relating to extract class refactoring and class cohesion metrics from 2010 to 2017.

Results: The literature on extract class refactoring and class cohesion metrics that already exist currently classified and made a comparison of each method.

Conclusion: From the systematic literature review there are still issues related to how many optimal number of classes to be formed from the extract class refactoring.

Keywords : cohesion, extract class, refactoring, object oriented programming.

Abstrak

Context: Cohesion merupakan faktor yang sangat diperhitungkan dalam menilai tingkat kualitas sebuah software yang menggunakan dasar Object Oriented Programming (OOP) dalam pengembangannya. Semakin besar nilai cohesion maka class tersebut semakin independen sehingga semakin mudah dilakukan maintenance pada saat software berevolusi. Dalam pengembangan OOP diharapkan memiliki nilai cohesion yang tinggi.

Objective: Paper ini menggunakan studi literatur sistematis terkait pada salah satu teknik refactoring yaitu extract class yang merupakan suatu cara untuk meningkatkan nilai cohesion dari sebuah class dan class cohesion metric untuk menilai tingkat kompleksitas class.

Method: Dalam paper ini akan dilakukan studi literatur secara sistematis pada dua database jurnal yang berkaitan dengan extract class dan class cohesion metric hingga tujuh tahun terakhir.

Results: Literatur tentang extract class refactoring dan class cohesion metric yang telah ada saat ini diklasifikasikan dan dilakukan perbandingan dari masing-masing metode.

Conclusion: Dari review studi literatur masih terdapat issue terkait berapa jumlah optimal class yang harus dibentuk dari extract class refactoring.

Kata Kunci : cohesion, extract class, refactoring, object oriented programming.

1. Pendahuluan

Refactoring merupakan sebuah studi yang membahas tentang restrukturisasi sebuah code. Refactoring berguna untuk pengembangan software dalam konteks maintenance software yang selalu berkembang. Dalam melakukan refactoring tentu juga mempertimbangkan perhitungan cohesion metric berdasarkan informasi yang ada dalam source code (Lucia, Oliveto and R. Vorraro 2008). Salah satu jenis refactoring yang ada adalah Extract Class. Extract Class diterapkan pada sebuah class yang terdiri dari banyak method dan atribut yang membuat class tersebut menjadi tidak terstruktur dan sulit untuk dipahami. Dengan menggunakan Extract Class refactoring akan dibuat sebuah class baru yang akan menampung beberapa method dan atribut

dari class lama tersebut sehingga struktur class lebih terstruktur dan lebih mudah dipahami dan diharapkan nilai cohesion semakin tinggi.

1.1. Class Cohesion

Cohesion pada sebuah class merupakan salah satu atribut yang penting dalam pengembangan software object oriented. Cohesion dihitung sebelum tahap desain untuk memprediksi kualitas sebuah software (Al Dallal 2011). Dengan melakukan penilaian terhadap cohesion pada sebuah class saat pengembangan sebuah software object oriented maka akan didapatkan desain software yang lebih mudah dipahami dan mudah untuk dilakukan maintenance ketika software berevolusi (Dallal and Briand, An object-oriented high-level design-based class cohesion metric 2010). Dalam paradigma object oriented, cohesion dari sebuah class menunjukkan sejauh mana member dalam sebuah class yaitu atribut dan method saling terhubung (Kaur 2011). Cohesion dari sebuah class didefinisikan pada level class (Yadav and Khan 2011). Sebuah class didefinisikan sebagai kombinasi antara atribut dan method. Atribut dan methods dalam sebuah class saling terhubung yang berarti sebuah atribut digunakan atau dipanggil oleh methods. Cohesion dari sebuah class berarti banyaknya method yang saling berkaitan dalam sebuah class (Birkmeier and Q. 2010). Sebuah class yang memiliki cohesion rendah terdiri dari elemen yang tidak saling terhubung atau melakukan terlalu banyak eksekusi (Ahmed, Abubakar and AlGhamdi 2011). Semakin besar nilai cohesion sebuah class menghasilkan desain class yang lebih mudah dimengerti, mudah diubah, dan mudah untuk dilakukan maintain (Halim and Mursanto 2011).

1.2. Extract Class

Permasalahan rendahnya nilai cohesion dapat diatasi dengan beberapa teknik refactoring. Dari semua teknik refactoring tersebut bertujuan untuk melakukan extract terhadap beberapa data dan fungsionalitas sebuah class ke dalam class yang baru. Move method, move atribut, dan extract class merupakan beberapa teknik untuk mengatasi hal tersebut. Namun extract class merupakan solusi yang paling baik untuk meningkatkan nilai cohesion sebuah class (Fokaefs, et al. 2011).

Extract Class Refactoring bertujuan untuk merestruktur sebuah code class dengan mendistribusikan beberapa fungsionalitasnya ke class baru sehingga mengurangi kompleksitas dan meningkatkan cohesion class tersebut (Bavota, De Lucia, et al. 2012). Dengan melakukan Extract Class Refactoring secara manual mungkin akan sangat sulit dikarenakan kompleksitas dari beberapa class.

2. Metode

Metode Systematic Literature Review (SLR) yang digunakan dalam paper ini adalah yang telah dideskripsikan oleh Kitchenman dan Charters yang terdiri dari beberapa langkah yaitu membentuk research question, melakukan search process, study selection, quality assesment, dan data extraction process.

2.1. Research Question

Tujuan utama dari paper ini adalah untuk mengidentifikasi dan mengklasifikasi study literature yang berkaitan dengan extract class dan penghitungan cohesion class sehingga penting untuk mendefinisikan research question untuk menyeleksi literature mana yang terkait dengan hal tersebut. Research question yang dibentuk dalam paper ini adalah:

RQ 1: Apa saja metode extract class yang ada hingga saat ini?

RQ 1.1: Apa perbandingan dari setiap metode identifikasi extract class yang ada saat ini?

RQ 2: Apa saja metode penghitungan cohesion class yang ada saat ini?

RQ 2.1: Apa perbandingan dari setiap metode penghitungan cohesion class yang ada saat ini?

RQ 3: Apakah benar metode extract class yang ada saat ini memperbaiki nilai cohesion?

RQ 4: Apakah keterbatasan dari metode identifikasi extract class saat ini masih menjadi issue?

2.2. Proses Pencarian

Pada bagian ini akan dilakukan proses pencarian studi literature yang terkait dengan tujuan dari paper ini. Proses pencarian dilakukan pada dua database jurnal yaitu:

- Science Direct (www.sciencedirect.com)
- IEEE eXplore (www.ieeeexplore.ieee.org)

Proses pencarian yang dilakukan adalah berdasarkan pencarian string judul, keyword, dan abstract pada jurnal dan conference yang diterbitkan antara tahun 2010 hingga 2014. Keyword yang digunakan terdiri dari:

- "Class Cohesion" AND "Metrics"
- "Class Cohesion" AND "Measurement"
- "Extract Class" AND "Refactoring"
- "Extract Class" AND "Cohesion"

Tabel 2.1. Hasil Query String

Strings	IEEE	Sciencedirect	Konten	Tahun
"Class Cohesion" AND "Metrics"	28	14	Journal and Conference	2010-2017
"Class Cohesion" AND "Measurement"	32	3		
"Extract Class" AND "Refactoring"	10	7		
"Extract Class" AND "Cohesion"	7	4		
Total	75	28		
	103			

2.3. Ekstraksi Data

Pada bagian ini akan dilakukan penyeleksian semua studi literatur yang telah didapat di search process sebelumnya dengan menentukan kriteria inklusi dan eksklusi. Kriteria inklusi dan eksklusi digunakan untuk menyeleksi literatur mana yang termasuk dan tidak termasuk dalam topik extract class atau cohesion measure.

Tabel 2.2. Kriteria Inklusi dan Eksklusi

Inklusi	Ekskulusi
<ol style="list-style-type: none"> 1. Fokus utama studi literatur adalah membahas tentang metode/algorithm atau tools yang digunakan untuk mengidentifikasi extract class atau penghitungan nilai cohesion class. 2. Metode penghitungan nilai cohesion adalah yang dilakukan pada level class. 3. Studi Literatur berupa jurnal atau paper conference yang dipublikasikan pada rentang tahun 2010 hingga 2014 4. Literatur menggunakan bahasa Inggris. 	<ol style="list-style-type: none"> 1. Fokus utama bukan pada penjelasan metode extract class atau penghitungan nilai cohesion class. 2. Tidak menjelaskan tentang dataset atau studi kasus yang digunakan dalam percobaan metode.

Gambar 2.1 menunjukkan langkah-langkah yang dilakukan untuk melakukan eksklusi studi literatur di berbagai level. Studi literatur dilakukan eksklusi berdasarkan Tabel 2.1.



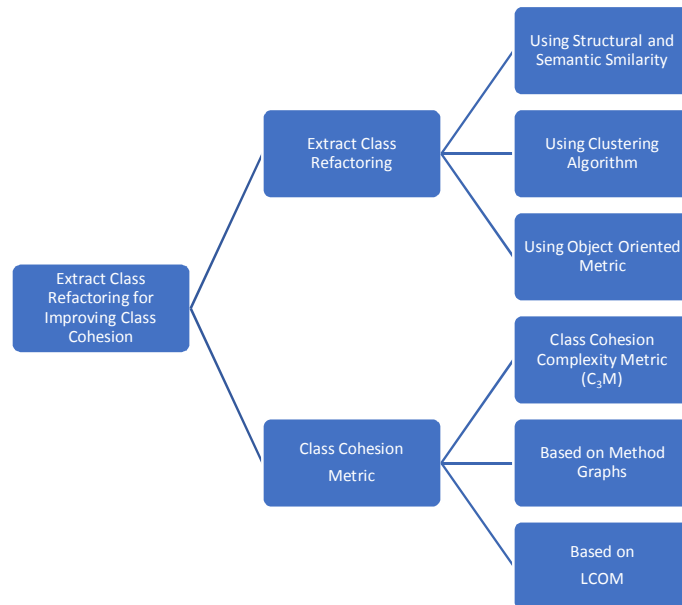
Gambar 2.1. Langkah-langkah eksklusi

Tabel 2.3 Hasil Eksklusi

ID	Judul	Sumber	Tipe	Tahun
S1	Identifying Extract Class refactoring opportunities using structural and semantic cohesion measure	ScienceDirect	Journal	2011
S2	Playing with refactoring: Identifying extract class opportunities through game theory	IEEE	Conference	2010
S3	Identification and application of Extract Class refactorings in object-oriented system	ScienceDirect	Journal	2012
S4	JDeodorant: identification and application of extract class refactorings	IEEE	Conference	2013
S5	Suggesting Extract Class Refactoring Opportunities by Measuring Strength of Method Interactions	IEEE	Conference	2013
S6	A New Cohesion Metric and Restructuring Technique for Object Oriented Paradigm	IEEE	Conference	2013
S7	Class cohesion complexity metric (C3M)	IEEE	Conference	2011
S8	Utilization of Method Graphs to Measure Cohesion in Object Oriented Software	IEEE	Conference	2013
S9	Transitive-based object-oriented lack-of-cohesion metric	ScienceDirect	Conference	2011
S10	Measuring Class Cohesion Based on Iterative Process Using External Class Relationships	IEEE	Conference	2014

2.4. Data Mapping

Studi literatur yang dibahas dalam paper ini mencakup dua poin utama yaitu tentang penghitungan nilai class cohesion (Class Cohesion Metric and Measurement) dan pengidentifikasian extract class refactoring. Dari dua poin utama tersebut akan dilakukan pemetaan berdasarkan metode yang dibahas di dalamnya. Untuk poin extract class refactoring terdapat perluasan pembahasan yaitu extract subclass refactoring yang juga akan dilakukan pemetaan metodenya. Gambar 2.2 menunjukkan diagram pemetaan studi literatur yang dilakukan dalam paper ini.



Gambar2.1. Data Mapping dari studi literatur

3. Hasil dan Diskusi

3.1. Hasil Perbandingan Studi Literatur Extract Class Refactoring

Perbandingan metode yang telah dijelaskan pada studi literatur akan dilakukan berdasarkan beberapa data item, yaitu: (1) mekanisme atau metode, (2) dataset, (3) akurasi.

Table 3.1. Perbandingan metode dan mekanisme extract class refactoring

Literatur	Penjelasan
Structural and semantic similarity (Bavota, Lucia and Oliveto 2011)(Bavota, Oliveto, et al. 2010)	<ol style="list-style-type: none"> 1. Membentuk sebuah graph dari sebuah class di mana edge dari graph tersebut merupakan nilai distance dari antara elemen yang ada dalam sebuah class. Nilai distance dihitung berdasarkan nilai Structural Similarity between Method (SSM), Call-based Dependence between Methods (CDM), dan Conceptual Similarity between Methods (CSM). 2. Melakukan class extraction menggunakan algoritma MaxFlow-Mincut (Bavota, Lucia and Oliveto 2011) untuk melakukan pemisahan atau splitting pada graph yang telah terbentuk berdasarkan treshold yang ditentukan dan menggunakan teori Nash Equilibrium (Bavota, Oliveto, et al. 2010) untuk membentuk class baru.
Clustering (Fokaefsa, et al. 2012)(Fokaefs, et al. 2011)	<ol style="list-style-type: none"> 1. Melakukan clustering dengan menggunakan algoritma hierarchical agglomerative terhadap setiap elemen sebuah class seperti atribut dan methodnya berdasarkan nilai distance dari setiap elemen tersebut. Nilai distance dihitung menggunakan rumus Jaccard Distance. 2. Melakukan ranking terhadap class-class yang teridentifikasi untuk dilakukan extract class berdasarkan perhitungan entity placement value. 3. Melakukan class extraction terhadap class berdasarkan treshold ranking yang ditentukan.

Object Oriented Metric (Pappalardo and Tramontana 2013)(Kaya and Fawcett 2012)	<ol style="list-style-type: none"> 1. Metode (Pappalardo and Tramontana 2013) menggunakan Object Oriented Metric untuk menghitung keterkaitan setiap method guna mengidentifikasi Extract Class Refactoring. Object Oriented metric akan menentukan method yang dapat dipisahkan dalam sebuah class dengan menghitung interaksi method tersebut dengan method ataupun atribut yang lain. 2. Metode (Kaya and Fawcett 2012) menggunakan program slicing dan graph teory dalam mendefinisikan class cohesion metric. Program slicing mendefinisikan data member pada line ke-n (statement) dan mencari statement lain yang mempengaruhi data member tersebut pada source code program. Selanjutnya akan dibentuk sebuah Data Slicing Graph (DSG) berdasarkan relasi data member dan dilakukan pemisahan pada data member yang saling tidak memiliki relasi.
--	---

Tabel 3.2. Perbandingan dataset

Literature	Dataset
Structural dan semantic similarity (Bavota, Lucia and Oliveto 2011)(Bavota, Oliveto, et al. 2010)	Dataset yang digunakan adalah software open source yang cukup dikenal yaitu: <ol style="list-style-type: none"> 1. ArgoUMLv0.16 sebuah tool pemodelan UML yang memiliki fitur reverse engineering dan code generation. Terdiri dari 1071 class 2. Eclipsev3.2 sebuah Integrated Development Environment (IDE) multi language. Terdiri dari 23462 class 3. JhotDrawv6.0 b1 sebuah tool yang digunakan untuk membuat java GUI. Terdiri dari 275 class
Clustering (Fokaefsa, et al. 2012)(Fokaefs, et al. 2011)	Dataset yang digunakan terdiri dari tiga projek yaitu: <ol style="list-style-type: none"> 1. CLRServerPack sebuah software PDF annotating client yang berfungsi untuk mengelola data storage pada database. Terdiri dari 33 class dan 242 method. 2. TPSim sebuah framework yang digunakan untuk melakukan simulasi service aware software. Terdiri dari 161 class dan 1000 method. 3. TAPorWithCoverflow sebuah environment untuk analisa text berbasis web dengan integrasi web service. Terdiri dari 103 class dan 312 method.
Object Oriented Metric (Pappalardo and Tramontana 2013)	<ol style="list-style-type: none"> 1. JhotDraw sebuah tool yang digunakan untuk membuat java GUI. Terdiri dari 600 class dan +- 500 method.

Tabel 3.3. Perbandingan akurasi

Literatur	Akurasi
Structural and Semantic similarity (Bavota, Lucia and Oliveto 2011)(Bavota, Oliveto, et al. 2010)	Akurasi dihitung menggunakan rumus <i>F-Measure</i> . Yaitu: $F\text{-measure}_{c_i} = 2 \times \frac{precision_{c_i} \times recall_{c_i}}{precision_{c_i} + recall_{c_i}}$ Akurasi yang dihasilkan dari setiap dataset adalah: <ol style="list-style-type: none"> 1. ArgoUML, 0.803 2. Eclipse, 0.740 3. JhotDraw, 0.791
Clustering (Bavota, Lucia and Oliveto 2011)(Bavota, Oliveto, et al. 2010)	Akurasi dihitung menggunakan rumus precision dan recall. Akurasi yang dihasilkan dari setiap dataset adalah: <ol style="list-style-type: none"> 1. CLRServerPack, Prec = 77.1 %, rec = 87.5 % 2. TPMSim, Prec = 53.8 %, Rec = 84.6 % 3. CoverFlow, Prec = 70 %, Rec = 75 %

Object Oriented Metric (Pappalardo and Tramontana 2013)	Metode ini menggunakan nilai LCOM sebagai hasil penelitiannya. Dari 33 class yang dianalisa menunjukkan bahwa 4 class nilai LCOM5 meningkat sedangkan 3 lainnya menurun dan lainnya tetap.
---	--

3.2. Hasil Perbandingan Studi Literatur Class Cohesion Metric

Pada bagian ini akan dilakukan review dan perbandingan dari setiap studi literatur yang berkaitan dengan penghitungan cohesion class.

Tabel 3.4. Perbandingan perhitungan nilai cohesion

Literatur	Penjelasan
Class Cohesion Complexity Metric [5]	<ol style="list-style-type: none"> 1. Metode ini menggunakan relasi antara sebuah method baik secara langsung (direct) maupun tidak langsung (indirect) dengan method atau atribut lain dalam sebuah class sebagai dasar pengitungan nilai cohesion dan kompleksitas sebuah class. 2. Low Complexity Cohesion Metric (LC_3) dan High Complexity Cohesion Metric dihitung dengan cara: $LC_3 = DUMD / NDIC$ $HC_3 = (DUMD U IDUM) / NDIC$ Apabila $LC_3 < HC_3$ maka class tersebut memiliki nilai cohesion rendah atau kompleksitas tinggi.
Method Graph [16]	<ol style="list-style-type: none"> 1. Metode ini melakukan analisa berbasis graph untuk menghitung Lack of Cohesion Metric (LCOM). 2. Menggunakan Abstract Syntax Tree berbasis Java Parser untuk mengekstrak empat graph terpisah yang bergantung pada perbedaan pemanggilan method yang muncul pada software object oriented. 3. Menggunakan graph clustering untuk mengelompokkan method yang berhubungan. Setiap cluster pada sebuah graph menunjukkan hubungan kuat.
Based LCOM [17]	<ol style="list-style-type: none"> 1. Metode ini merupakan perbaikan dari LCOM dengan mempertimbangkan hubungan transitive antara method dengan method atau atribut lain (Dallal, Transitive-based object-oriented lack-of-cohesion metric 2011) dan relasi eksternal dengan class lain (Lee and Kim 2014)

3.3. Analisa Kelebihan dan Kekurangan Metode

Pada bagian ini akan dilakukan penilaian oleh penulis tentang kelebihan dan kekurangan dari setiap metode. Analisa dilakukan pada metode extract class refactoring dan class cohesion metric.

Tabel 3.5. Kelebihan dan kekurangan metode Extract Class Refactoring

Metode	Kelebihan	Kekurangan
Structural dan semantic similarity (Bavota, Lucia and Oliveto 2011)(Bavota, Oliveto, et al. 2010)	Dengan menggunakan kombinasi penghitungan SSM, CDM, dan CSM penghitungan tingkat relasi antara elemen class menjadi akurat.	Algoritma MaxFlow MinCut yang digunakan hanya bisa memisahkan class menjadi dua bagian.

Clustering (Fokaefsa, et al. 2012)(Fokaefs, et al. 2011)	Menggunakan penghitungan Entity Placement Metric dan meranking setiap class berdasarkan dampak yang dihasilkan terhadap desain software sehingga memberikan opsi kepada user terhadap class mana yang terlebih dahulu dilakukan refactoring.	Algoritma Hierarchical Clustering membutuhkan input user secara manual terkait banyaknya class yang akan terbentuk dari extract class refactoring.
Object Oriented Metric(Pappalardo and Tramontana 2013)(Kaya and Fawcett 2012)	Dengan menggunakan penilaian object oriented metric yang telah terdefiniskan maka tingkat relasi antara elemen menjadi akurat.	Pemisahan class dilakukan manual oleh user berdasarkan nilai relasi dari object oriented metric.

Tabel 3.6 Kelebihan dan kekurangan class cohesion metric

Metode	Kelebihan	Kekurangan
Class Cohesion Complexity Metric (Yadav and Khan 2011)	Memiliki kompleksitas rendah dalam melakukan penghitungan LC_3 dan HC_3	hanya memberikan penilaian kompleksitas berdasarkan nilai LC_3 dan HC_3 apabila $LC_3 > HC_3$ kelas tersebut kompleks dan sebaliknya sehingga tidak terdapat nilai pasti tentang tingkat kompleksitas class tersebut.
Method Graph (Aral and Ovatman 2013)	Mengurangi kompleksitas dari perhitungan LCOM dengan membentuk graph dan cluster antara elemen class yang terdiri dari method dan atribut	Walaupun mengurangi kompleksitas dari LCOM namun metode ini memiliki akurasi di bawah LCOM
Based LCOM (Dallal, Transitive-based object-oriented lack-of-cohesion metric 2011)(Lee and Kim 2014)	Merupakan perbaikan dari metode LCOM dengan mempertimbangkan relasi transitive antara method dan atribut (Dallal, Transitive-based object-oriented lack-of-cohesion metric 2011) dan relasi eksternal dengan class lain (Lee and Kim 2014).	Belum didukung oleh tools.

4. Kesimpulan

Dalam paper ini telah dilakukan review literatur metode pengidentifikasian extract class refactoring dan penghitungan class cohesion metric yang dipublikasikan antara tahun 2010 hingga tahun 2014 dan berasal dari dua database jurnal yaitu sciendirect dan IEEE. Dari studi literatur yang dibahas terdapat tiga metode utama pada extract class refactoring yaitu menggunakan structural-semantic similarity, clustering, dan object oriented metric. Sedangkan pada penghitungan class cohesion terdapat tiga metode utama dalam penghitungan class cohesion yaitu menggunakan class cohesion complexity metric, method graphs, dan transitive based LCOM. Dari review yang telah dilakukan penulis terdapat perkembangan yang signifikan dari penelitian yang telah dilakukan dalam melakukan extract class refactoring dan sangat berguna untuk membentuk sebuah software yang maintainable dan understandable. Selain itu juga terdapat perkembangan yang signifikan dari penelitian yang dilakukan dalam penghitungan class cohesion yang berguna untuk menilai tingkat kualitas dari sebuah software berdasarkan class-class yang ada di dalamnya.

Setelah dilakukan review extract class refactoring terdapat beberapa issue yang masih menjadi permasalahan saat ini. Pada metode structural-semantic similarity setelah graph terbentuk akan dilakukan pemisahan node pada graph tersebut menggunakan algoritma MaxFlow Mincut yang hanya bisa memisahkan graph menjadi dua buah cluster sehingga metode ini hanya bisa

memisahkan sebuah class menjadi dua. Selain itu pada metode Clustering membutuhkan input dari user berapa banyak cluster yang akan terbentuk sehingga metode ini tidak dapat secara otomatis melakukan pemisahan class dengan jumlah yang paling optimal. Pada metode extract class menggunakan object oriented metric hanya memberikan penilaian terhadap relasi setiap elemen dalam sebuah class sehingga tetap membutuhkan peran user untuk memisahkan class berdasarkan penilaian object oriented tersebut. Setelah dilakukan review class cohesion metric juga terdapat beberapa issue yang masih menjadi permasalahan saat ini. Metode Class Complexity Cohesion Metric (C_3M) hanya memberikan penilaian kompleksitas berdasarkan nilai LC_3 dan HC_3 apabila $LC_3 > HC_3$ kelas tersebut kompleks dan sebaliknya sehingga tidak terdapat nilai pasti tentang tingkat kompleksitas class tersebut.

DAFTAR PUSTAKA

- Ahmed, Moataz A., Adam Abubakar, and Jarallah S. AlGhamdi. 2011. "A study on the uncertainty inherent in class cohesion measurements." *Journal of Systems Architecture* 57 (4): 474-484.
- Al Dallal, J. 2011. "Measuring the Discriminative Power of Object-Oriented Class Cohesion Metrics." *Software Engineering, IEEE Transactions* 37 (6): 788-804.
- Aral, A., and T. Ovatman. 2013. "Utilization of Method Graphs to Measure Cohesion in Object Oriented Software." *Computer Software and Applications Conference Workshops* 505-510.
- Bavota, G., A. De Lucia, A. Marcus, R. Oliveto, and F Palomba. 2012. "Supporting Extract Class Refactoring in Eclipse: The ARIES Project." *Software Engineering (ICSE)* 1419-1422.
- Bavota, G., R. Oliveto, A. De Lucia, G. Antoniol, and Y. Guéhéneuc. 2010. "Playing with refactoring: Identifying extract class opportunities through game theory." *Software Maintenance (ICSM), 2010 IEEE International Conference* 1-5.
- Bavota, Gabriele, Andrea De Lucia, and Rocco Oliveto. 2011. "Identifying Extract Class refactoring opportunities using structural and semantic cohesion measure." *Journal of Systems and Software* 84 (3): 397-414.
- Birkmeier, and Dominik Q. 2010. "On the State of the Art of Coupling and Cohesion." *Conference*.
- Dallal, Jehad Al. 2012. "Constructing models for predicting extract subclass refactoring opportunities." *Information and Software Technology* 54 (10): 1125-1141.
- Dallal, Jehad Al. 2011. "Transitive-based object-oriented lack-of-cohesion metric." *Procedia Computer Science* 3: 1581-1587.
- Dallal, Jehad Al, and Lionel C. Briand. 2010. "An object-oriented high-level design-based class cohesion metric." *Information and Software Technology* 52 (12): 1346-1361.
- Fokaefs, M., N. Tsantalis, E. Stroulia, and A Chatzigeorgiou. 2011. "JDeodorant: identification and application of extract class refactorings." *Software Engineering (ICSE)* 1037-1039.
- Fokaefsa, Marios, Nikolaos Tsantalisa, Eleni Strouliiaa, and Alexander Chatzigeorgioub. 2012. "Identification and application of Extract Class refactorings in object-oriented system." *Journal of System and Software* 85 (10): 2241-2260.
- Halim, A., and P. Mursanto. 2011. "Refactoring rules effect of class cohesion on high-level design." *Information Technology and Electrical Engineering (ICITEE)* 197-202.
- Ibrahim, S.M., S.A. Salem, M.A. Ismail, and M. Eladawy. 2012. "Novel sensitive object-oriented cohesion metric." *Computer Theory and Applications (ICCTA), 2012 22nd International Conference* 154-159.
- Joshi, P., and R.K. Joshi. 2010. "Quality Analysis of Object Oriented Cohesion Metrics." *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference* 319-324.
- Kaur, K. Singh, H. 2011. "Towards a Valid Metric for Class Cohesion at Design Level." *Emerging Applications of Information Technology* 351-354.
- Kaya, Mehmet, and James W Fawcett. 2012. "A New Cohesion Metric and Restructuring Technique for Object Oriented Paradigm." *Computer Software and Applications Conference Workshops (COMPSACW)* 296-301.
- Lee, Jun-Ha, and Dae-Kyoo Kim. 2014. "Measuring Class Cohesion Based on Iterative Process Using External Class Relationships." *Quality Software (QSIC) 14th International Conference* 321,326.
- Lucia, De, A. Oliveto, and L. R. Vorraro. 2008. "Using Structural and Semantic Metrics to Improve Class Cohesion." *Software Maintenance* 27-36.

- Pappalardo, G., and E. Tramontana. 2013. "Suggesting Extract Class Refactoring Opportunities by Measuring Strength of Method Interactions." *Software Engineering Conference 2*: 105-110.
- Tsantalis, N., and A. Chatzigeorgiou. 2009. "Identification of Move Method Refactoring Opportunities." *Software Engineering* 35 (3): 347-367.
- Yadav, A., and R.A. Khan. 2011. "Class cohesion complexity metric (C3M).," *Computer and Communication Technology (ICCCT)* 363,366.