

PENGUKURAN KUALITAS UML DENGAN CK METRIC, DERAJAT KOHESI DAN COUPLING

Arwin Halim¹, Alex Xandra Albert Sim², Andam Sari Purnama Sulthan³

^{1,2,3}Teknik Informatika STMIK Mikroskil

Jl. Thamrin No. 112, 124, 140, Medan 20212, Indonesia

arwin@mikroskil.ac.id¹, alex.sim@mikroskil.ac.id², andamspsulthan@gmail.com³

Abstrak – kualitas merupakan salah satu aspek penting dalam pengembangan perangkat lunak. Pengukuran kualitas secara dini dapat meningkatkan keandalan sistem dan mengurangi biaya perbaikan aplikasi. Penelitian ini mengembangkan alat ukur kualitas pada tahapan analisis dan desain Unified Modeling Language (UML). Nilai metrik kualitas diperoleh dari nilai CK-Metric, derajat kohesi dan coupling. Nilai metrik digunakan untuk menentukan kualitas dari perangkat lunak melalui *fuzzy system*. Data penelitian diperoleh dari perangkat lunak open-source yang menggunakan bahasa pemrograman Java, yaitu Lucene.Net-1.9.RC1 dan Lucene.Net-1.4.3. Hasil dari penelitian menunjukkan alat ukur yang dibangun mampu menghitung nilai metrik kualitas dan menunjukkan indikasi kualitas dari *class-class* pada perangkat lunak beroorientasi objek.

Kata kunci – CK-Metrics, coupling, kohesi, kualitas perangkat lunak.

I. PENDAHULUAN

Program berorientasi objek sangat populer digunakan para programmer dalam membuat program rumit atau skala besar yang bersifat fleksibel. Konsep berorientasi objek populer karena memungkinkan programmer untuk memperbaiki dan modifikasi kode program dengan mudah. Namun, perbaikan dan modifikasi pada program tersebut sering dilakukan pada akhir implementasi. Perubahan di akhir pengembangan dapat meningkatkan biaya dan waktu dari proyek. Hal ini disebabkan oleh pengukuran kualitas yang dilakukan setelah tahapan implementasi. Hal tersebut dapat dihindari apabila pengukuran kualitas dilakukan pada tahapan awal, seperti pada tahap perancangan Unified Modeling Language (UML).

Perangkat lunak berorientasi objek dapat dimodelkan dengan menggunakan UML. UML memiliki notasi yang standar dan berfungsi sebagai cetak biru dari aplikasi yang dibangun. Cetak biru perangkat lunak yang diperoleh pada saat tahapan pemodelan mengandung informasi rinci tentang perangkat lunak secara keseluruhan. UML biasa digambarkan dalam bentuk diagram sesuai dengan kegunaannya, seperti *class* diagram, *sequence* diagram, *activity* diagram dan lain-lain. Pengukuran kualitas dapat dilakukan pada tahapan desain melalui UML, sehingga perbaikan perangkat lunak dapat dilakukan secara mudah [1]. Evaluasi desain UML

telah dilakukan oleh Penulis [2]. Penelitian tersebut mengusulkan pendekatan untuk mengevaluasi, mendeteksi, dan memperbaiki desain dari UML *Class* Diagram untuk meningkatkan kualitas dari perangkat lunak yang dibangun. Penulis juga menguji kualitas kohesi [3] dan kompleksitas [4] pada desain UML.

Berbagai metrik kualitas telah diusulkan untuk mengukur kualitas dari perangkat lunak. Metrik kualitas dapat diukur pada tahapan desain dan kode program. Nilai metrik kualitas digunakan untuk tujuan strategis manajer proyek dan tim perangkat lunak untuk memodifikasi atau memperbaiki untuk meningkatkan kualitas produk perangkat lunak. Beberapa contoh metrik misalnya Source Lines of Code (SLOC), metrik yang digunakan untuk memperkirakan besar suatu aplikasi dengan menghitung jumlah baris dari source code program. Salah satu metrik pengukuran kualitas perangkat lunak pada kode program adalah Chidamber dan Kemerer Metrics Suite atau lebih dikenal sebagai CK Metrik [5]. CK Metrik terdiri dari enam nilai metrik yaitu Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number Of Children (NOC), Coupling Between Object classes (CBO), Response For a Class (RFC), Lack of Cohesion in Methods (LCOM). Selain itu, Saxena dan Kumar [6] mengusulkan metrik derajat kohesi dan derajat coupling yang mengukur kualitas dari desain UML. Nilai kohesi dan coupling menunjukkan keterkaitan antar *class* dan digunakan untuk mengontrol kompleksitas dari desain. Semakin rendah nilai coupling dan semakin tinggi nilai kohesi menunjukkan desain perangkat lunak yang baik. Nilai dari metrik-metrik tersebut yang akan digunakan untuk menentukan kualitas perangkat lunak secara keseluruhan. Mursanto dan Halim [7] telah mengusulkan model pengambilan keputusan berdasarkan multi kriteria untuk memilih perangkat lunak. Pada penelitian ini, kualitas untuk setiap *class* dari perangkat lunak berorientasi objek dapat diukur dengan CK-Metrik, derajat kohesi dan derajat coupling yang diukur pada tahapan desain melalui diagram UML. Kualitas secara keseluruhan dapat ditentukan berdasarkan *fuzzy system* yang telah diusulkan oleh Mago dan Kaur [8].

II. METODE PENELITIAN

A. CK- Metrik

CK–Metrik merupakan paket metrik kualitas yang digunakan untuk mengukur kualitas perangkat lunak berorientasi objek [5]. Metrik kualitas pada CK-Metrik antara lain:

1. Weighted Methods per Class (WMC): metrik yang menjumlahkan kompleksitas dari setiap metode per class. Jika sebuah class C1 memiliki sejumlah method M1, M2, ..., Mn maka nilai WMC dihitung dengan Persamaan 1. Semakin tinggi nilai WMC, semakin buruk kualitas dari aplikasi.

$$WMC = \sum_{i=1}^n c_i \quad (1)$$

2. Depth of Inheritance Tree (DIT): metrik yang menghitung jumlah kedalaman hirarki class. Semakin tinggi nilai DIT berarti semakin besar potensi penggunaan kembali metode yang diwarisinya.
3. Number of Children (NOC): metrik yang menunjukkan jumlah dari class yang diwariskan secara langsung dalam sebuah hirarki class. Nilai NOC berbanding lurus dengan besar upaya dan waktu untuk testing.
4. Coupling Between Object: metrik yang menghitung jumlah class yang berpasangan. Sebuah class dikatakan berpasangan dengan class lainnya, jika sebuah class menggunakan metode atau instance variable pada class lainnya. Semakin tinggi nilai CBO, semakin buruk kualitas dari aplikasi.
5. Response for a Class (RFC): metrik yang menghitung respon sebuah pesan dari class. Semakin besar nilai RFC, semakin buruk kualitas aplikasi. RFC dapat dihitung dengan Persamaan 2 dan 3.

$$RFC = |RS| \quad (2)$$

$$RS = \{M\} \cup_{all i} \{R_i\} \quad (3)$$

Dimana

{M}: himpunan seluruh metode dalam class
{R_i}: himpunan metode yang dipanggil oleh metode i.

6. Lack of Cohesion in Method (LCOM). LCOM dapat dihitung dengan Persamaan 4.

$$LCOM = |P| - |Q|, \text{ jika } |P| > |Q|$$

$$LCOM = 0, \text{ jika } |P| \leq |Q| \quad (4)$$

B. Derajat Coupling

Derajat coupling (DC) [6] dihitung sebagai perbandingan dari jumlah pesan yang di terima terhadap sejumlah pesan yang dilalui.

$$\text{Degree of Coupling (DC)} = \frac{MRC}{MPC} \quad (5)$$

Dimana:

MRC = jumlah pesan yang diterima class dari class lain

MPC = jumlah pesan yang lewat diantara object pada class.

C. Derajat Cohesion

Derajat kohesi (DHC) [6] menghitung kekuatan fungsi atribut asosiasi dalam class, dimana hal itu menunjukkan seberapa kuat sebuah method bergantung pada sebuah class.

$$\text{Degree of Cohesion (DCH)} = \frac{NAU}{TNA} \quad (6)$$

Dimana:

NAU = jumlah atribut yang digunakan pada class

TNA = jumlah atribut pada class

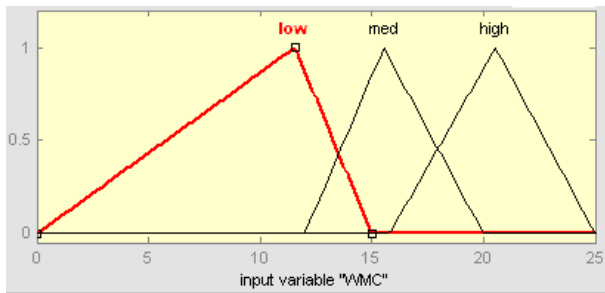
D. Pengaruh Fitur Metrik dengan Kualitas Perangkat Lunak

Metrik perangkat lunak memungkinkan pengembang untuk mengukur dan memprediksi proses dan sumber daya pada proyek pengembangan perangkat lunak. Metrik tersebut akan mempengaruhi atribut kualitas seperti reliability, maintainability, efficiency, portability, usability dan reusability. Hubungan dari CK-Metrik [8], Derajat Coupling dan Derajat Cohesion dengan kualitas perangkat lunak terlihat pada Tabel 1.

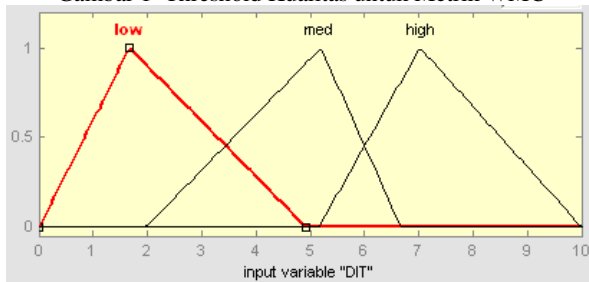
Tabel 1 Pengaruh CK Metrik dan Faktor Kualitas

CK Metrik	Kualitas	Fitur OO
WMC	Maintainability, Understandability, Usability, Reusability	Class/ Method
RFC	Understandability, Usability, Testability	Class / Method
LCOM, DHC	Reusability, Efficiency	Class / Method
CBO, DC	Reusability, Efficiency	Coupling
DIT	Reusability, Efficiency, Understandability, Testability	Inheritance
NOC	Reusability, Efficiency, Understandability	Inheritance

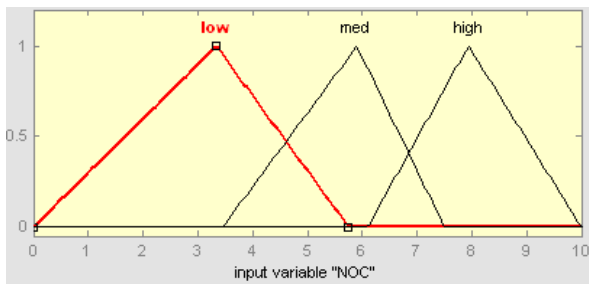
Berdasarkan pengaruh kualitas dan nilai metrik, kita dapat menentukan nilai threshold-nya. Threshold pada CK Metrik telah diusulkan oleh Mago dan Kaur [8] dengan menggunakan fuzzy system. Threshold tersebut ditunjukkan pada Gambar 1 sampai Gambar 6.



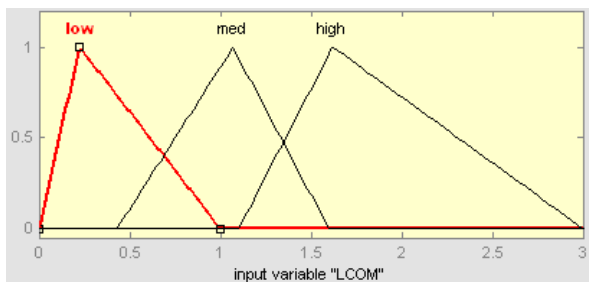
Gambar 1 Threshold Kualitas untuk Metrik WMC



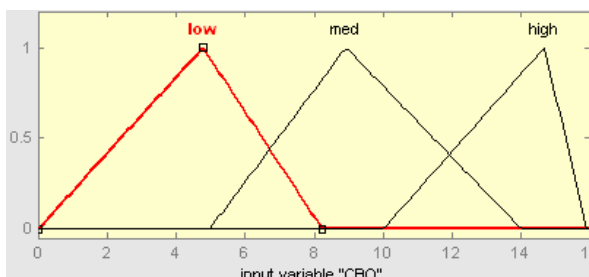
Gambar 2 Threshold Kualitas untuk Metrik DIT



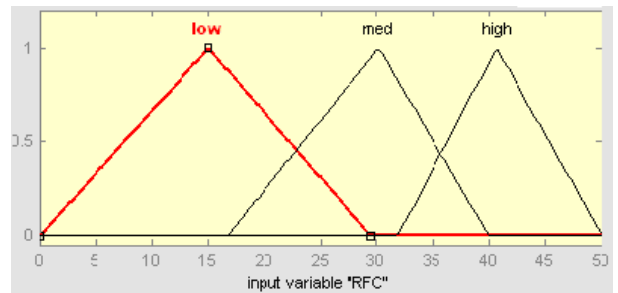
Gambar 3 Threshold Kualitas untuk Metrik NOC



Gambar 4 Threshold Kualitas untuk Metrik LCOM



Gambar 5 Threshold Kualitas untuk Metrik CBO



Gambar 6 Threshold Kualitas untuk Metrik RFC

Nilai threshold yang direkomendasikan pada CK-Metrik [9] ditunjukkan pada Tabel 2.

Tabel 2 Nilai Threshold Kualitas untuk CK Metrik

Metrics	Nilai Threshold
WMC	<i>Low</i> , < 11
DIT	<i>Low</i> , < 4
NOC	<i>Low</i> , < 3
CBO	<i>Low</i> , < 3
RFC	<i>Low</i> , < 12
LCOM	<i>Low</i> , < 4

Pada penelitian ini, nilai kualitas akan dibagi menjadi high dan low. Semua nilai dalam kategori medium akan dikonversi menjadi low. Nilai metrik yang direkomendasikan untuk mendapatkan perangkat lunak yang berkualitas dilihat pada Tabel 3.

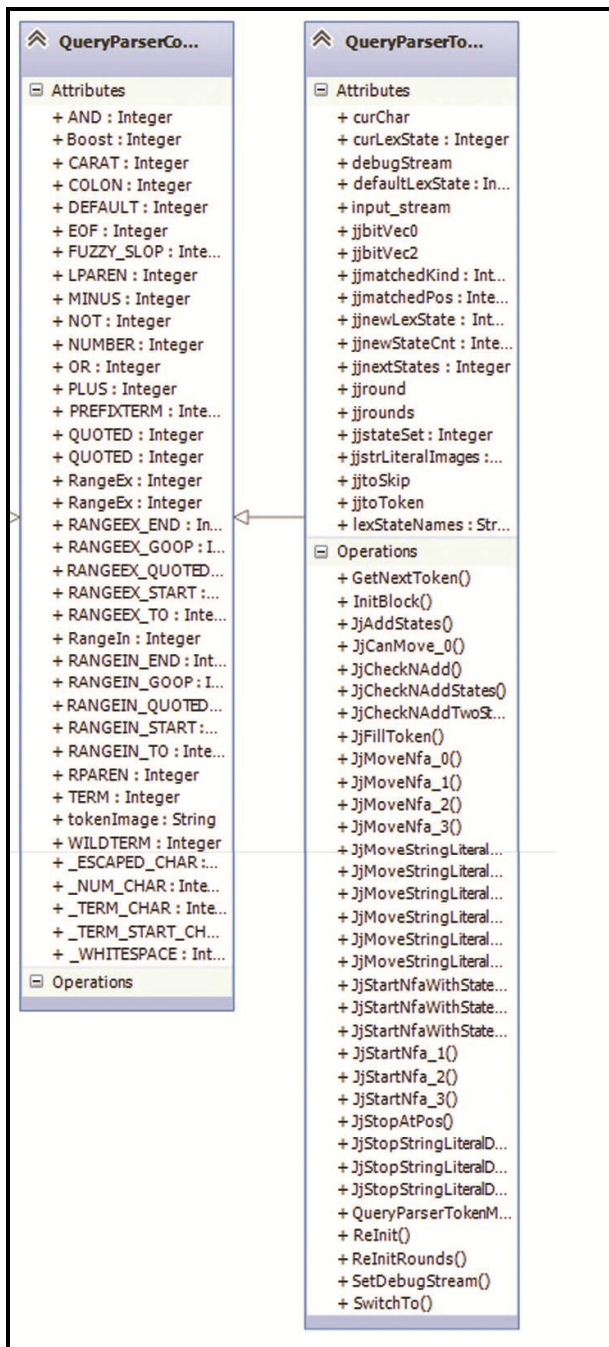
Tabel 3 Nilai Kualitas yang Direkomendasikan untuk CK Metrik

CK Metric	Recommended
WMC	<i>Low</i>
DIT	<i>Low</i>
NOC	<i>High</i>
CBO	<i>Low</i>
RFC	<i>Low</i>
LCOM	<i>Low</i>

III. HASIL DAN PEMBAHASAN

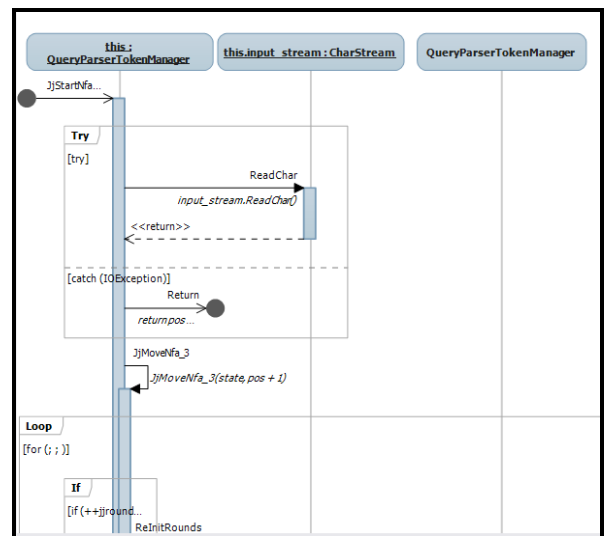
A. Data Penelitian

Data penelitian yang digunakan adalah perangkat lunak Lucene.Net-1.9.RC1 dan Lucene.Net-1.4.3. Lucene.Net adalah port dari Lucene search engine library, yang ditulis dalam C# dan ditargetkan pada pengguna NET. Lucene search engine library melakukan proses pencarian berdasarkan indeks terbalik. Lucene.Net-1.9.RC1 memiliki 7 package utama yang berisi sejumlah *class* di dalam nya yaitu, Analysis memiliki 32 *class*, Document memiliki 5 *class*, Index memiliki 40 *class*, QueryParser memiliki 9 *class*, Search memiliki 83 *class*, Store memiliki 14 *class*, dan Util memiliki 7 *class*. Gambar 7 menunjukkan contoh UML *Class* diagram pada *class* Lucene.Net-1.4.3.



Gambar 7 Contoh UML *Class* Diagram Lucene.Net-1.4.3

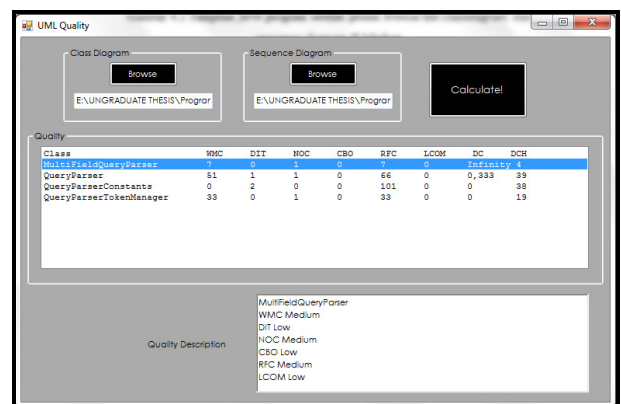
Gambar 8 menunjukkan sampel UML sequence diagram `QueryParserTokenManager_JjStartNfa_3` pada `Lucene.Net-1.4.3`.



Gambar 8 Contoh UML Sequence Diagram Lucene.Net-1.4.3

B. Tampilan Aplikasi Pengukur Kualitas

Aplikasi Pengukur Kualitas UML dibangun menggunakan bahasa pemrograman C#. Aplikasi menerima masukan berupa file *class* diagram dan sequence diagram yang dihasilkan dari Visual Studio 2013. Gambar 9 menunjukkan tampilan dari aplikasi pengukur kualitas UML.



Gambar 9 Tampilan Aplikasi Pengukur Kualitas UML

C. Pengujian

Dalam pengujian, dilakukan perhitungan kualitas untuk Aplikasi Lucene.Net-1.9.RC1 dan Lucene.Net-1.4.3 pada empat kelas yang sama. Pemilihan *class* yang digunakan dilakukan secara acak. Kelas tersebut adalah `MultiFieldQueryParser`, `QueryParser`, `QueryParserConstants` dan `QueryParserTokenManager`. Hasil Perhitungan Metrik Kualitas untuk keempat kelas tersebut pada Lucene.Net-1.9-RC1 ditampilkan pada Tabel 3.

Tabel 3 Hasil Perhitungan Metrik Kualitas Lucene.Net-1.9-RC1

Class	W M C	D I T	N O C	C B O	R F C	LC OM	D C	D C H
MultiFieldQueryParser	7	0	1	0	7	0	~	4
QueryParser	51	1	1	0	66	0	0,33	39
QueryParserConstants	0	2	0	0	101	0	0	38
QueryParserTokenManager	33	0	1	0	33	0	0	19

Hasil Perhitungan Metrik Kualitas untuk keempat kelas tersebut pada Lucene.Net-1.4.3 ditampilkan pada Tabel 4.

Tabel 4 Hasil Pengukuran Kualitas Lucene.Net-1.4.3

Class	W M C	D I T	N O C	C B O	R F C	LC OM	D C	D C H
MultiFieldQueryParser	2	0	1	0	17	0	0,33	3
QueryParser	45	1	1	0	45	0	~	36
QueryParserConstants	0	2	0	0	101	0	0	35
QueryParserTokenManager	29	0	1	0	29	0	0	19

Berdasarkan nilai metrik yang dihitung, aplikasi akan menarik kesimpulan mengenai kualitas dari masing-masing kelas dan metrik kualitasnya. Tabel 5 menunjukkan deskripsi kualitas dari empat kelas yang telah diuji di atas pada Lucene.Net-1.9-RC1.

Tabel 5 Hasil Pengukuran Kualitas Lucene.Net-1.9-RC1

Class	W M C	D I T	N O C	C B O	R F C	LC OM	D C	D C H
MultiFieldQueryParser	L	L	L	L	L	L	~	4
QueryParser	H	L	L	L	H	L	0,33	39
QueryParserConstants	L	L	L	L	H	L	0	38
QueryParserTokenManager	H	L	L	L	H	L	0	19

Hasil pengukuran kualitas untuk keempat kelas pada Lucene.Net-1.4.3 ditampilkan pada Tabel 6.

Tabel 4 Hasil Pengukuran Kualitas Lucene.Net-1.4.3

Class	W M C	D I T	N O C	C B O	R F C	LC OM	D C	D C H
MultiFieldQueryParser	L	L	L	L	L	L	0,33	3
QueryParser	H	L	L	L	H	L	~	36
QueryParserConstants	L	L	L	L	H	L	0	35
QueryParserTokenManager	H	L	L	L	H	L	0	19

Berdasarkan hasil pengukuran kualitas UML Lucene.Net-1.9-RC1 dan Lucene.Net-1.4.3, dapat ditarik kesimpulan sebagai berikut:

1. Untuk nilai WMC, *class* QueryParser dan QueryParserTokenManager memperoleh kualitas High, hal ini berpengaruh buruk untuk *Maintainability*, *Understandability*, *Usability*, *Reusability* pada *class* tersebut. Nilai yang di rekomendasikan adalah Low dan hasil ini berada pada *class* QueryParserConstants dan MultiFieldQueryParser.
2. Untuk nilai DIT, *class* QueryParser dan QueryParserConstants memperoleh kualitas Medium, hal ini berpengaruh untuk *Reusability*, *Efficiency*, *Understandability*, *Testability* pada *class* tersebut. *Class* MultiFieldQueryParser dan *class* QueryParserTokenManager memperoleh hasil Low, artinya *class* memiliki kualitas yang baik, karena nilai yang di rekomendasikan adalah Low.
3. Seluruh *Class* memperoleh hasil Low, berpengaruh pada *Reusability*, *Efficiency*, *Understandability*. Nilai yang direkomendasikan adalah High, artinya tidak ada *class* yang memiliki performa maksimal.
4. Untuk nilai CBO, seluruh *class* memiliki nilai Low, hasil ini baik untuk *Reusability*, *Efficiency* pada *class* tersebut.
5. Untuk nilai RFC, *class* MultiFieldQueryParser memperoleh kualitas Low, hal ini berpengaruh baik untuk *Understandability*, *Usability*, *Testability*. *Class* lain nya memperoleh hasil High, artinya *class* memiliki kualitas yang tidak baik, karena nilai yang di rekomendasikan adalah Low.
6. Untuk nilai LCOM, seluruh *class* memiliki nilai Low, hasil ini baik untuk *Reusability*, *Efficiency* pada *class* tersebut.

IV. KESIMPULAN

Aplikasi pengukuran kualitas yang dikembangkan telah dapat menghitung nilai kualitas secara objektif berdasarkan nilai metrik kualitas pada perangkat lunak berorientasi objek. Selain menghitung nilai metrik, aplikasi juga telah menentukan kualitas dari setiap *class*, sehingga dapat digunakan menjadi pedoman pengembang ketika maintainance aplikasi. Metrik yang digunakan CK-Metrik, Derajat Kohesi dan Derajat Coupling mampu memberikan gambaran kualitas *class* yang rinci pada setiap *class*. Hal ini ditunjukkan dari hasil pengujian dari aplikasi yang sama dengan versi yang berbeda yang telah mengalami perubahan kode program. Aplikasi masih mampu mengidentifikasi kualitas yang sama walaupun memiliki nilai metrik yang berbeda.

V. SARAN

Pada pengembangan berikutnya, dapat dicoba untuk menggabungkan metrik lain untuk menarik kesimpulan yang lebih efektif dan jelas.

VI. REFERENSI

- [1] Soliman, T.H., El-Swesy, A., Ahmed, S. H., 2010, Utilizing CK Metrics Suite to UML Models: a Case Study of Microarray MIDAS Software, The 7th International Conference on Informatics and Systems (INFOS).
- [2] Halim, A. 2012, Evaluasi dan Perbaikan Kualitas Desain Diagram Kelas, Jurnal Sistem Informasi STMIK Mikroskil, vol 13 no 2, pp. 133-140.
- [3] Halim, A. 2013, Refactoring Rules Effects of *Class* Cohesion on High Level Design, IEEE International Conference on Information Technology and Electrical Engineering
- [4] Halim, A. 2013, Predict Fault Prone *Classes* using the complexity of UML *Class* Diagram, IEEE International Conference on Computer, Control, Informatics and Its Applications
- [5] Chidamber, S.R. & Kemerer, C.F., 1994, Metric Suite for Object Oriented Design, IEEE Transactions On software Engineering, volume 20 , Halaman 476 – 493.
- [6] Saxena, V. & Kumar, S., 2012, Impact of Coupling and Cohesion in Object Oriented Technology, Journal of Software Engineering and Applications, Volume 5, halaman 671-676
- [7] Mursanto, Petrus dan Halim, Arwin. 2015, Combination of AHP and PROMETHEE for Measuring Quality of Object Oriented Software Design, International Journal of the Analytic Hierarchy Process
- [8] Mago, J. & Kaur, P., 2012, Analysis of Quality of the Design of the Object Oriented Software using Fuzzy Logic, International Conference on Recent Advances and Future Trends in Information Technology, halaman 21 - 25
- [9] Antony, J dan Dev, H., 2013, Thresholds For Reliability From The Thresholds Of Ck Metrics, VSRD International Journal of Computer Science & Information Technology, Vol. 3, Halaman 207 -216