Vol. 5 No. 3, 2024, pp. 238-248 **Jurnal Pepadun**
©2024 Ilmu Komputer Unila Publishing Network all rights reserved

Implementasi Pengukuran *Object Oriented Metrics* (Studi Kasus Aplikasi Movie DB)

[1]Hartsa Hanifah, [*2]Anie Rose Irawati, dan [3]Yohana Tri Utami

[1,2,3]Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung,
Jl. Prof. Dr. Ir. Sumantri Brojonegoro No. 1, Gedung Meneng, Rajabasa, Bandar Lampung, Indonesia
e-mail: [1]hartsa.hanifah1029@fmipa.unila.ac.id, [*2]**anie.roseirawati@fmipa.unila.ac.id**,
[3]yohana.utami@fmipa.unila.ac.id

Abstract *—The development of high-quality software is a primary goal in software engineering, with Object-Oriented Programming (OOP) being a widely adopted approach. OOP promotes modularity, reusability, and maintainability, which are essential for creating scalable and maintainable systems. Research in object-oriented software metrics has grown in importance, with Chidamber and Kemerer proposing six key metrics to evaluate OOP designs: Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Objects (CBO), Response for a Class (RFC), and Lack of Cohesion in Methods (LCOM).In this study, the source code of the Java classes in the Movie DB Application is analyzed using these metrics, calculated with the CK-Metrics Suite. The values of the Chidamber and Kemerer metrics are then compared with established benchmarks to assess the quality of the object-oriented principles implemented in the software.The results indicate that the Movie DB Application demonstrates strong adherence to key OOP principles, including maintainability, usability, reusability, understandability, modifiability, and testability. These findings suggest that the application's object-oriented design is of high quality, supporting long-term software success. By using these metrics, developers can gain insights into the software's strengths and identify potential areas for improvement, ultimately enhancing software quality and ensuring its adaptability to future changes.*

Keywords**:** *Chidamber & Kemerer Metrics; CK Metrics Suite; Object Oriented Metrics; Quality Factor.*

1. PENDAHULUAN

Perkembangan teknologi perangkat lunak yang semakin terstruktur, khususnya menuju perangkat lunak berorientasi objek (Object-Oriented Programming, OOP), telah memengaruhi kemajuan penelitian di bidang *software metrics*. Sebelumnya, perangkat lunak yang dikembangkan dengan pendekatan prosedural sering kali diukur menggunakan metrik untuk mengidentifikasi tingkat kompleksitas program. Tujuan utama dalam pembuatan perangkat lunak adalah menciptakan perangkat lunak yang berkualitas tinggi. Kualitas perangkat lunak ini tidak hanya berpengaruh terhadap performa teknis, tetapi juga memengaruhi kinerja perusahaan dalam menjalankan proses bisnisnya. Perangkat lunak yang berkualitas akan meningkatkan efisiensi operasional perusahaan, sehingga dapat menghemat sumber daya dan mempercepat pencapaian tujuan bisnis [1].

Saat ini, banyak pengembang perangkat lunak lebih memilih menggunakan konsep OOP karena memudahkan dalam membangun perangkat lunak yang lebih modular dan fleksibel. Dalam OOP, perangkat lunak dibangun berdasarkan objek yang terdiri dari data dan sekumpulan operasi (metode) yang diterapkan pada data tersebut [2]. Untuk menilai kualitas desain perangkat lunak berorientasi objek, Chidamber dan Kemerer mengusulkan enam metrik utama yang disebut CK Metrics Suite. Metrik-metrik ini terdiri dari *Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Objects (CBO), Response for a Class (RFC),* dan *Lack of Cohesion in Methods (LCOM)* [3]. Metrik-metrik ini digunakan untuk mengukur kualitas perangkat lunak pada level kelas dan sangat berguna dalam evaluasi desain perangkat lunak berbasis objek [4].

Penelitian ini bertujuan untuk mengimplementasikan pengukuran menggunakan metrik-metrik tersebut untuk mengevaluasi kualitas perangkat lunak berbasis objek pada studi kasus Aplikasi Movie DB. Aplikasi Movie DB merupakan aplikasi berbasis Java yang menampilkan informasi film-film terkini yang sedang tayang serta

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

film-film yang akan datang. Aplikasi ini menggunakan konsep desain berorientasi objek yang memungkinkan untuk pengelolaan data secara lebih terstruktur. Dengan melakukan perhitungan metrik CK pada kode program aplikasi ini, penelitian ini menghasilkan nilai kuantitatif yang mencerminkan kualitas desain perangkat lunak Aplikasi Movie DB. Hasil pengukuran ini diharapkan dapat memberikan wawasan yang berguna bagi pengembang Aplikasi Movie DB untuk meningkatkan desain dan kualitas kode program mereka. Secara lebih luas, penelitian ini menunjukkan bagaimana penerapan *software metrics* dapat digunakan sebagai alat bantu untuk mengevaluasi dan memperbaiki kualitas perangkat lunak berorientasi objek, yang pada gilirannya dapat mendukung pengembangan perangkat lunak yang lebih efisien dan mudah dipelihara.

2. METODOLOGI PENELITIAN

Metode pengerjaan penelitian ini terdiri dari 4 tahap. Masing-masing tahap memiliki proses yang disesuaikan dengan tujuan tahap. Penelitian ini diawali dengan studi literatur, pengumpulan data, kemudian melakukan implementasi perhitungan *metrics*, dan menganalisis hasil perhitungan *metrics*.

2.1. Studi Literatur

Studi literatur dilakukan dengan mempelajari jurnal, buku-buku referensi, dan situs yang terkait dengan *metrics* perangkat lunak berorientasi objek. *Metrics* yang digunakan dalam penelitian ini adalah *object-oriented metrics* yaitu enam buah *metrics* yang diajukan oleh Chidamber dan Kemerer (1994). Metrik Chidamber dan Kemerer adalah salah satu metrik yang digunakan untuk mengukur kualitas disain sebuah perangkat lunak berdasarkan enam metrik dengan melihat perspektif desain berorientasi objek [5].

2.2. Pengumpulan Data

Pengumpulan data dilakukan setelah Aplikasi Movie DB selesai dikembangkan. Data yang dikumpulkan adalah semua yang terkait dengan pengukuran *object-oriented metrics.* Dalam proses pengukuran kualitas dengan enam *object-oriented metrics* pada Aplikasi Movie DB diperlukan *source code* lengkap. Aplikasi tersebut dikembangkan dengan bahasa pemrograman Java, terdiri dari 12 *classes*. Perhitungan *metrics* dilakukan secara manual membutuhkan semua *file* *.java (*java file*).

Metrik Chidamber dan Kemerer mempunyai enam metrik yaitu *Weighted Method per Class* (WMC), *Depth of Inheritance Tree* (DIT), *Number OF Children* (NOC), *Coupling Between Object Classes* (CBO), *Response for a Class* (RFC), dan *Lack of Cohesion Method* (LCOM). SATC (*Software Assurance Technology Center*) telah menghasilkan pemetaan Metrik Chidamber dan Kemerer pada kode program perangkat lunak dengan parameter-parameter Metrik Chidamber dan Kemerer yang mempunyai objektivitas dari pengukuran kualitas perangkat lunak [1]. Pemetaan Metrik Chidamber dan Kemerer ditunjukkan pada Tabel 1.

Tabel 1. Pemetaan Metrik Chidamber dan Kemerer dengan kode program.

Source	***Metrics***	***Object Oriented Construct***
Object oriented (New)	*Weighted Methods per Class* (WMC)	*Class/Method*
Object oriented (New)	*Depth of Inheritance Tree* (DIT)	*Inheritance*
Object oriented (New)	*Number of Children* (NOC)	*Inheritance*
Object oriented (New)	*Coupling Between Object Classes* (CBO)	*Coupling*
Object oriented (New)	*Response For a Class* (RFC)	*Class/Message*
Object oriented (New)	*Lack of Cohesion in Method* (LCOM)	*Class/Cohesion*

Nilai metrik WMC, DIT, NOC, CBO, RFC berdasarkan penelitian sebelumnya [1] dan nilai metrik LCOM [6] telah menghasilkan kategori dari Metrik Chidamber dan Kemerer yang ditunjukkan pada Tabel 2.

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

Tabel 2. Nilai Metrik Chidamber dan Kemerer.

Metrik	Baik	Sedang	Buruk
WMC	$1 \leq x < 50$	$50 \leq x \leq 150$	$150 < x$
DIT	$x < 5$	$5 \leq x < 8$	$8 \leq x$
NOC	$x < 4$	$4 \leq x < 8$	$8 \leq x$
CBO	$x < 14$	$14 \leq x \leq 150$	$150 < x$
RFC	$x < 100$	$100 \leq x \leq 1000$	$1000 < x$
LCOM	$0 \leq x \leq 1$		

2.3. Implementasi Perhitungan *Metrics*

Perhitungan *metrics* dilakukan secara manual dengan melihat *source code class* (*.java) dari data yang diperoleh. Setiap *class* akan diukur nilai metriknya dan dihitung rata-ratanya (*average*). Patokan nilai yang dapat dijadikan parameter dalam pengukuran di tahap ini terdapat pada pembahasan pada Tabel 2. Metrik Chidamber dan Kemerer mempunyai enam metrik, yaitu:

a. *Weighted Methods per Class* (WMC)
   WMC menghitung jumlah *method* dalam setiap *class* [3]. Nilai WMC yang tinggi akan mengakibatkan program rentan terhadap *bug* dan terlalu kompleks untuk dipahami. Semakin rendah nilai akan semakin bagus. Semakin tinggi nilai *methods* akan semakin sulit pengujiannya [13].
b. *Depth of Inheritance Tree* (DIT)
   DIT menghitung jumlah tingkatan dari kelas *node* ke *root* dari *inheritance hierarchy tree* [12]. Semakin besar nilai DIT maka akan semakin kompleks kode program tersebut. Kode program yang kompleks berisiko lebih tinggi dalam pengembangan serta pemakaiannya [13].
c. *Number of Children* (NOC)
   NOC menghitung jumlah *subclass* yang diturunkan langsung dari suatu *class*. Sebuah *class* dengan NOC yang besar akan sulit dimodifikasi dan membutuhkan pengujian yang lebih karena berpengaruh pada perubahan *children* [3].
d. *Coupling Between Objects Classes* (CBO), menghitung banyaknya kolaborasi untuk sebuah kelas atau jumlah hubungan kelas dengan kelas yang lain [13]. Nilai CBO yang tinggi akan mengindikasikan *coupling* yang berlebihan sehingga merugikan desain modular dan membatasi *reuse*, menyulitkan *testing* dan modifikasi [1].
e. *Response For a Class* (RFC), menghitung jumlah semua *method* yang dipanggil sebagai respon terhadap *object* luar dari sebuah *class* [3]. Sehingga semakin tinggi nilai RFC maka, akan semakin banyak *method* yang digunakan untuk merespon *object* dari luar dan semakin kompleks sehingga akan meningkatkan waktu untuk *testing* [14].
f. *Lack of Cohesion in Method* (LCOM), menghitung hasil yang diperoleh dari memperkirakan jumlah pasangan metode di kelas yang tidak memiliki kesamaan atribut [3]. LCOM ini sudah mengalami beberapa penyempurnaan dan sekarang yang dipakai adalah LCOM5 [6]. LCOM5 dengan jumlah metode yang mengakses setiap set atribut atau data, khususnya hanya variabel instan. Untuk LCOM5 yang memiliki nilai 0 dianggap kohesi sempurna.

Persamaan 5 berikut menunjukkan metrik LCOM5 [15]:

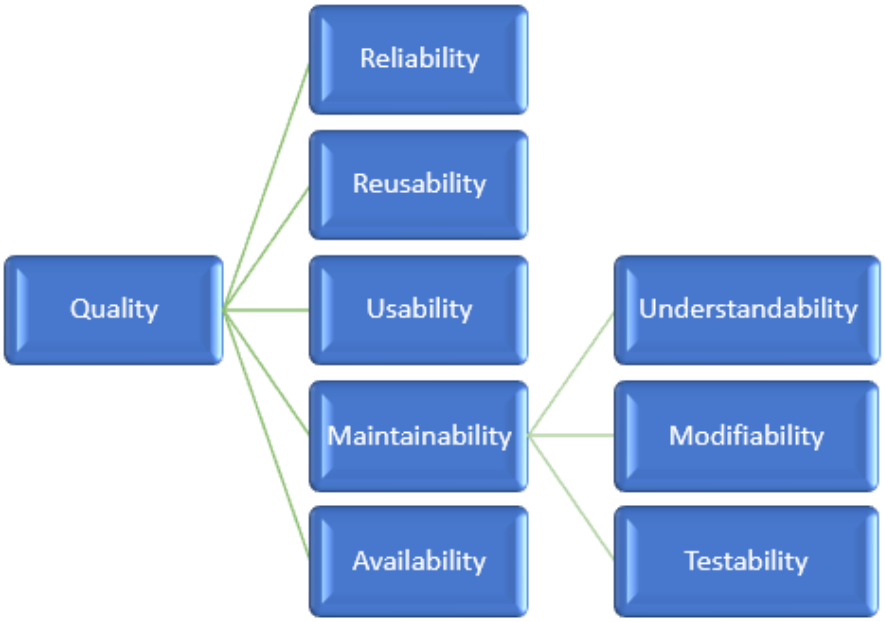
$$LCOM5 = \frac{\left(\frac{1}{a} x\ Mu\right) - m}{1 - m} \quad (1)$$

Keterangan:
a = banyak *variable* dalam satu *class.*
Mu = jumlah pemanggilan atribut oleh seluruh *method* dalam satu *class.*
m = jumlah *method* dalam *class.*

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

2.4. Analisis Hasil Perhitungan *Metrics*

Analisis pengukuran dilakukan dengan cara membandingkan nilai enam metrik Chidamber dan Kemerer dengan nilai standar *metrics* yang dianjurkan Chidamber dan Kemerer [3]. Dari hasil perhitungan, dicari nilai minimal, maksimal, dan rata-rata secara keseluruhan dari masing-masing *metrics*. Dari hasil keenam *metrics* dapat dianalisis seberapa baik Aplikasi Movie DB secara keseluruhan berdasarkan faktor-faktor kualitas yang terkait dengan masing-masing *metrics.* Tinggi rendahnya faktor-faktor kualitas pada masing-masing *metrics* dipengaruhi oleh tinggi rendahnya nilai *metrics.* Kualitas perangkat lunak yang baik dari sudut pandang pengembang yaitu pengembang akan melihat produk perangkat lunak dari dalam perangkat lunak itu sendiri. Pengembang yang menggunakan pemikiran berorientasi objek memiliki tujuan pada terpenuhinya karakteristik tertentu. Terpenuhinya karakteristik-karakteristik tersebut merupakan kualitas dari sudut pandang pengembang [7]. Karakteristik tersebut berupa faktor kualitas. Faktor-faktor tersebut dapat dilihat pada Gambar 1.



Gambar 1. Faktor-faktor kualitas perangkat lunak [7].

Definisi faktor-faktor tersebut sebagai berikut:

a. *Reliability*
   Reliability adalah tingkat ketepatan fungsi-fungsi perangkat lunak tanpa adanya kesalahan [8].
b. *Reusability*
   Reusability adalah tingkat kemampuan perangkat lunak atau bagian perangkat lunak untuk dapat digunakan ulang di lain tempat sebagai komponen yang *reusable* [8].
c. *Usability*
   Usability adalah tingkat usaha yang dibutuhkan untuk mempelajari, mengoperasikan, dan menggunakannya untuk tujuan yang diinginkan [8].
d. *Maintainability*
   Maintainability adalah tingkat usaha untuk mengetahui letak kesalahan dan memperbaikinya. Faktor ini dapat dipecah menjadi tiga subfaktor, masing-masing *understandability*, *modifiability*, dan *testability* [7].
e. *Understandability*
   Understandability terkait dengan peningkatan kompleksitas psikologis [9].
f. *Modifiability*

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

Menurut IEEE Std. 610.12 diacu dalam [10], *modifiability* dari segi *flexibility* adalah tingkat kemudahan sebuah sistem atau komponen agar dapat dimodifikasi untuk penggunaan dalam suatu aplikasi atau lingkungan.

g. *Testability*
Testability adalah tingkat usaha yang dibutuhkan untuk menguji perangkat lunak dalam proses *quality assurance* [8].
h. *Availability*
Menurut IEEE Std. 610.12 diacu dalam [10], *availability* adalah tingkat kemudahan suatu sistem atau komponen agar dapat dioperasikan dan diakses ketika ingin digunakan.

Semua faktor kualitas tersebut dapat diukur secara kuantitatif menggunakan *metrics.* Kualitas perangkat lunak yang digunakan berdasarkan faktor-faktor kualitas yang terkait dengan enam metrik tersebut memiliki masing-masing fungsi [11], sebagai berikut:

a. *Weighted Methods per Class* (WMC) mempengaruhi terhadap faktor-faktor kualitas yaitu *maintainability* (*testability*), *usability*, *reusability*.
b. *Depth of Inheritance Tree* (DIT) berpengaruh pada *reusability*, *understandability*, *modifiability*, dan *testability*.
c. *Number of Children* (NOC) berpengaruh pada *reusability* dan *testability*.
d. *Coupling Between Object* (CBO) berpengaruh dengan *reusability* dan *maintainability*.
e. *Response for A Class* (RFC) terkait dengan *maintainability*, *understandability*, *modifiability*, dan testability.
f. *Lack of Cohesion in Method* (LCOM) berpengaruh pada *understandability*, *modifiability*, dan *testability*.

Berdasarkan faktor-faktor kualitas yang terkait dengan keenam *metrics* tersebut menentukan tinggi rendahnya nilai *metrics* yang mempengaruhi pada tinggi rendahnya faktor-faktor kualitas yang terkait pada masing-masing *metrics*. Sebagai contoh, nilai WMC yang tinggi menyebabkan menurunnya tingkat *maintainability* (*testability*), *usability*, dan *reusability* [11].

3. HASIL DAN PEMBAHASAN

3.1. Analisis Perhitungan Metrik WMC

WMC dihitung berdasarkan banyaknya jumlah *method* yang diimplementasikan dalam suatu *class* [6]. Cara menghitungnya adalah dengan memindai setiap baris pada *code* yang memiliki *keywords* untuk *method,* yaitu '*void*' [13]. Dari hasil perhitungan WMC pada Tabel 3, diperoleh nilai minimal WMC adalah 2, nilai maksimal WMC adalah 18, dan nilai rata-rata WMC adalah 6,5. Berdasarkan referensi pada Tabel 2, menunjukkan bahwa keseluruhan nilai metrik WMC pada Aplikasi Movie DB memiliki nilai WMC yang baik karena berada di bawah batas ambang, sehingga tidak ada *class* yang perlu dipecah menjadi beberapa *class*. Maka dapat disimpulkan pengukuran dengan metrik WMC pada Aplikasi Movie DB menunjukkan tingkat *maintainability* (*testability*), *usability*, dan *reusability* yang baik. Hasil perhitungan nilai metrik WMC pada Aplikasi Movie DB dapat dilihat pada Tabel 3.

3.2. Analisis Perhitungan Metrik DIT

DIT diukur berdasarkan jumlah dari induk *class*-nya. Cara menghitung DIT adalah dengan menampung data semua nama *class* yang ada pada *file input*-an, kemudian memeriksa apakah *class* tersebut terdapat *inheritance* dengan mencari *keywords* '*extends*' [13]. Setiap *class* merupakan turunan dari *class Object* sehingga DIT setiap *class* pada pemrograman Java minimal 1. *Class* Object sendiri memiliki DIT = 0 [16].

Dari hasil perhitungan pada Tabel 4, diperoleh nilai minimal DIT adalah 1, nilai maksimal DIT adalah 4, dan nilai rata-rata DIT adalah 1,92. Berdasarkan referensi pada Tabel 2, dengan nilai DIT < 5 berada dalam kategori baik. Semua *class* pada Aplikasi Movie DB memiliki nilai DIT di bawah 5, sehingga masing-masing kelas tersebut tidak perlu mengalami penggabungan dengan kelas lain. Hal ini menunjukkan bahwa keseluruhan

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

nilai metrik DIT pada Aplikasi Movie DB memiliki nilai DIT yang baik. Maka dapat disimpulkan pengukuran dengan metrik DIT pada Aplikasi Movie DB menunjukkan tingkat *reusability*, *understandability*, *modifiability*, dan *testability* yang baik. Hasil perhitungan nilai metrik DIT pada Aplikasi Movie DB ditunjukkan pada Tabel 4.

Tabel 3. Nilai Metrik WMC Aplikasi Movie DB.

No	Nama *Class*	Nilai WMC
1	CariFragment	7
2	DatangFragment	7
3	FavoritFragment	6
4	TayangFragment	7
5	Movie	5
6	DatabaseContract	4
7	DatabaseHelper	2
8	MovieHelper	8
9	MovieModel	18
10	Filem	6
11	DetailActivity	2
12	MainActivity	6
Total		**78**
Rata-rata		**6,5**

Tabel 4. Nilai Metrik DIT Aplikasi Movie DB.

No	Nama *Class*	Nilai DIT
1	CariFragment	2
2	DatangFragment	2
3	FavoritFragment	2
4	TayangFragment	2
5	Movie	4
6	DatabaseContract	1
7	DatabaseHelper	2
8	MovieHelper	1
9	MovieModel	1
10	Filem	2
11	DetailActivity	2
12	MainActivity	2
Total		**23**
Rata-rata		**1,92**

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

3.3. Analisis Perhitungan Metrik NOC

NOC diukur berdasarkan jumlah *subclass* yang diturunkan langsung dari suatu *class*. Setiap kelas akan diperiksa apakah merupakan anak dari kelas lain atau bukan dengan menggunakan *keyword* '*extends*'. Apabila kelas tersebut merupakan kelas anak, maka cari kelas induknya pada daftar yang telah ditampung, kemudian tambahkan nilai 1 untuk setiap anak kelas yang memiliki induk. Apabila ditemukan anak kelas lain yang memiliki induk yang sama maka nilainya juga akan ditambahkan [13].

Berdasarkan referensi pada Tabel 2, nilai NOC < 4 berada dalam kategori baik. Hasil perhitungan pada Tabel 5, menjelaskan bahwa NOC pada Aplikasi Movie DB seragam, yaitu sebesar 0. Ini berarti bahwa semua *class* pada Aplikasi Movie DB tidak memiliki *subclass*. Dengan demikian, rata-rata nilai NOC sangat rendah dibandingkan nilai batas atas yang disarankan pada Tabel 2 sehingga tidak diperlukan pemeriksaan ulang dan/atau revisi *class* dari segi ketepatan *sub-classing*. Hal ini menunjukkan bahwa keseluruhan nilai metrik NOC pada Aplikasi Movie DB memiliki nilai NOC yang baik. Maka dapat disimpulkan pengukuran dengan metrik NOC pada Aplikasi Movie DB menunjukkan tingkat *reusability* dan *testability* yang baik. Hasil perhitungan nilai metrik NOC pada Aplikasi Movie DB dapat dilihat pada Tabel 5.

Tabel 5. Nilai Metrik NOC Aplikasi Movie DB.

No	Nama *Class*	Nilai NOC
1	CariFragment	0
2	DatangFragment	0
3	FavoritFragment	0
4	TayangFragment	0
5	Movie	0
6	DatabaseContract	0
7	DatabaseHelper	0
8	MovieHelper	0
9	MovieModel	0
10	Filem	0
11	DetailActivity	0
12	MainActivity	0
Total		**0**
Rata-rata		**0**

3.4. Analisis Perhitungan Metrik CBO

Pengukuran CBO yaitu menghitung pasangan fungsi dengan kelas tipe objek. CBO adalah banyaknya kolaborasi untuk sebuah kelas atau jumlah hubungan kelas dengan kelas yang lain. Cara menghitungnya adalah pada *string* yang merupakan isi dari inputan akan di *list* terlebih dahulu seluruh *class* yang ada, kemudian dilakukan pemeriksaan apakah *class* tersebut merupakan *child class* dari kelas lain, kemudian dilakukan pula pemeriksaan apakah pada kelas tersebut ada dilakukan pemanggilan kelas lain dengan *keyword* "*new*" dan memeriksa apakah yang dipanggil merupakan data dari *class* yang ada atau bukan (kelas yang mungkin merupakan bagian dari *library* yang ada) [13].

Hasil perhitungan pada Tabel 6 diperoleh nilai minimal CBO adalah 0, nilai maksimal CBO adalah 1, dan nilai rata-rata CBO adalah 0,42. Berdasarkan referensi pada Tabel 2, nilai CBO < 14 berada dalam kategori baik. Hal ini menunjukkan bahwa keseluruhan nilai metrik CBO pada Aplikasi Movie DB memiliki nilai CBO yang

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

baik, sehingga setiap kelas tidak perlu mengalami penggabungan dengan kelas-kelas yang memiliki relasi dengan kelas tersebut. Maka dapat disimpulkan pengukuran dengan metrik CBO pada Aplikasi Movie DB menunjukkan tingkat *reusability*, *understandability*, *modifiability,* dan *testability* yang baik. Hasil perhitungan nilai metrik CBO pada Aplikasi Movie DB dapat dilihat pada Tabel 6.

Tabel 6. Nilai Metrik CBO Aplikasi Movie DB.

No	Nama *Class*	Nilai CBO
1	CariFragment	1
2	DatangFragment	1
3	FavoritFragment	1
4	TayangFragment	1
5	Movie	0
6	DatabaseContract	0
7	DatabaseHelper	0
8	MovieHelper	1
9	MovieModel	0
10	Filem	0
11	DetailActivity	0
12	MainActivity	0
Total		**5**
Rata-rata		**0,42**

Tabel 7. Nilai Metrik RFC Aplikasi Movie DB.

No	Nama *Class*	Nilai RFC
1	CariFragment	12
2	DatangFragment	11
3	FavoritFragment	10
4	TayangFragment	11
5	Movie	7
6	DatabaseContract	4
7	DatabaseHelper	2
8	MovieHelper	8
9	MovieModel	18
10	Filem	6
11	DetailActivity	3
12	MainActivity	7
Total		**99**
Rata-rata		**8,25**

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

3.5. Analisis Perhitungan Metrik RFC

Pengukuran RFC yaitu menghitung jumlah *method* lokal yang diimplementasikan ditambah *method* yang dipanggil dalam objek (*external method*) [6]. Perhitungan RFC dilakukan pada seluruh fungsi dari setiap *class* yang ada pada sistem. Hasil perhitungan pada Tabel 7 diperoleh nilai minimal CBO adalah 0, nilai maksimal CBO adalah 1, dan nilai rata-rata CBO adalah 0,42. Berdasarkan referensi pada Tabel 2, nilai CBO < 14 berada dalam kategori baik. Hal ini menunjukkan bahwa keseluruhan nilai metrik CBO pada Aplikasi Movie DB memiliki nilai CBO yang baik, sehingga *method* yang dimiliki oleh setiap kelas tidak perlu digabung dengan *method* lain. Maka dapat disimpulkan pengukuran dengan metrik CBO pada Aplikasi Movie DB menunjukkan tingkat *reusability*, *understandability*, *modifiability*, dan *testability* yang baik. Hasil perhitungan nilai metrik RFC pada Aplikasi Movie DB ditunjukkan pada Tabel 7.

3.6. Analisis Perhitungan Metrik LCOM

Pengukuran LCOM dilakukan dengan menghitung jumlah koneksi pada *method* [6]. LCOM digunakan mengukur derajat kemiripan *method* oleh variabel *input* data atau atribut dalam *class* [12]. Rumus LCOM5 yang digunakan pada penelitian ini dapat dilihat pada persamaan (1). Dari hasil perhitungan pada Tabel 8, diperoleh nilai minimal LCOM5 adalah 0,118, nilai maksimal LCOM5 adalah 1, dan nilai rata-rata LCOM5 adalah 0,68. Berdasarkan referensi pada Tabel 2, nilai ambang standar LCOM yaitu 0 sampai 1. Hal ini menunjukkan bahwa Aplikasi Movie DB memiliki nilai LCOM yang baik, sehingga kelas-kelas tersebut tidak perlu dipecah menjadi beberapa kelas. Maka dapat disimpulkan pengukuran dengan metrik DIT pada Aplikasi Movie DB menunjukkan tingkat *understandability*, *modifiability*, dan *testability* yang baik. Hasil perhitungan nilai metrik LCOM5 pada Aplikasi Movie DB dapat dilihat pada Tabel 8.

Tabel 8. Nilai Metrik LCOM5 Aplikasi Movie DB.

No	Nama *Class*	Nilai LCOM5
1	CariFragment	0,833
2	DatangFragment	0,722
3	FavoritFragment	0,914
4	TayangFragment	0,722
5	Movie	0,125
6	DatabaseContract	1
7	DatabaseHelper	1
8	MovieHelper	0,524
9	MovieModel	0,745
10	Filem	0,5
11	DetailActivity	0,118
12	MainActivity	1
	Total	**8,204**
	Rata-rata	**0,68**

3.7. Analisis Hasil Perhitungan Metrik

Aplikasi Movie DB dikatakan mempunyai tingkat *maintainability* yang baik ketika terdapat komponen yang sesuai dengan kebutuhan dan proses yang ada pada aplikasi, tingkat *understandability* yang baik ketika kode program dapat dengan mudah dipelajari, digunakan *user* dengan mudah dan menghasilkan *output* sesuai dengan aplikasi dibuat. Untuk tingkat *modifiability* yang baik ketika sebuah sistem atau komponen dapat

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

dimodifikasi untuk penggunaan sesuai dengan tujuan aplikasi dibuat, dan dikatakan mempunyai tingkat *testability* yang baik ketika aplikasi mudah diuji sesuai dengan fungsinya. Aplikasi Movie DB juga mempunyai tingkat *usability* yang baik ketika produk perangkat lunak dengan mudah dipelajari dan digunakan sesuai dengan tujuan aplikasi dibuat, dan aplikasi dikatakan mempunyai tingkat *reusability* yang baik ketika program/bagian dari program dapat dipakai ulang dalam aplikasi dan dalam perangkat lain. Pengaruh *CK Metrics* terhadap faktor-faktor kualitas berdasarkan sudut pandang masing-masing *metrics* dapat dilihat pada Tabel 9.

Tabel 9. Ringkasan pengaruh *CK Metrics* terhadap faktor-faktor kualitas berdasarkan sudut pandang setiap *metrics.*

	CK Metrics					
Faktor Kualitas	**WMC (6,5)**	**DIT (1,92)**	**NOC (0)**	**CBO (0,42)**	**RFC (8,25)**	**LCOM5 (0,68)**
Usability	✓					
Reusability	✓	✓	✓	✓		
Understandability		✓		✓	✓	✓
Modifiability		✓		✓	✓	✓
Testability	✓	✓	✓	✓	✓	✓

Berdasarkan perhitungan Metrik Chidamber dan Kemerer pada Aplikasi Movie DB, diperoleh:

1. Nilai metrik WMC yaitu 6,5 menunjukkan bahwa metrik WMC memiliki *maintainability* (*testability*), *usability*, dan *reusability* yang baik berdasarkan jumlah *method* setiap *class*.
2. Nilai metrik DIT yaitu 1,92 menunjukkan bahwa metrik DIT memiliki *reusability*, *understandability*, *modifiability*, dan *testability* yang baik berdasarkan kedalaman sebuah *class*.
3. Nilai metrik NOC yaitu 0 menunjukkan bahwa metrik NOC memiliki *reusability* dan *testability* yang baik berdasarkan jumlah *subclass* (*child class*) setiap *class*.
4. Nilai metrik CBO yaitu 0,42 menunjukkan bahwa metrik CBO memiliki *reusability*, *understandability*, *modifiability*, dan *testability* yang baik berdasarkan jumlah *coupling* pada setiap *class*.
5. Nilai metrik RFC yaitu 8,25 menunjukkan bahwa metrik RFC memiliki *understandability*, *modifiability*, dan *testability* yang baik berdasarkan jumlah *message* setiap *class*.
6. Nilai metrik LCOM5 yaitu 0,68 menunjukkan bahwa metrik LCOM memiliki *understandability*, *modifiability*, dan *testability* yang baik berdasarkan jumlah koneksi pada *method*.

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan bahwa pengukuran *object oriented metrics* dengan enam metrik Chidamber dan Kemerer, 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), dan *Lack of Cohesion in Methods* (LCOM) yang dapat digunakan untuk mengetahui kualitas perangkat lunak berorientasi objek dari beberapa faktor kualitas pada Aplikasi Movie DB memiliki nilai metrik yang baik. Aplikasi Movie DB sudah memiliki komponen kualitas *object-oriented* yang baik dari segi *usability*, *reusability*, *understandability*, *modifiability*, dan *testability*.

DAFTAR PUSTAKA

[1] A. Ayubi, "Quality Measurement of Object Oriented Code Using Chidamber and Kemerer Metric in The Perspective of Maintainability, Efficiency, Understandability, and Replaceability (Case Studies Software Accounting XYZ)," Final Project, Institut Teknologi Sepuluh November, Surabaya, Indonesia, Juli, 2015.

[2] J. Simarmata, *Rekayasa Perangkat Lunak*, Yogyakarta: Andi, 2010.

©2024 Ilmu Komputer Unila Publishing Network all rights reserved

[3] Shyam R. Chidamber & Chris F. Kemerer, "A Metrics Suite for Object Oriented Design," *IEEE Transactions on Software Engineering*, vol. 20, 1994.

[4] A. K. Sharma, A. Kalia, & H. Singh, "Metrics Identification for Measuring Object Oriented Software Quality," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, 2012.

[5] W. R. D. Septian, "Analisis Perbandingan Framework PHP Berdasarkan MOOSE CK dan Properti Kualitas Disain Menggunakan Metode Analytic Hierarchy Process (AHP)," Bachelor Thesis, UIN Syarif Hidayatullah, Jakarta, Indonesia, 2010.

[6] A. Mitra, "Analisis Sistem Informasi Data Nilai Siswa Berbasis PHP di SMK YPKK 1 Sleman," Bachelor Thesis, Universitas Negeri Yogyakarta, Yogyakarta, Indonesia, 2013.

[7] A. El-Ahmadi, "Software Quality Metrics for Object Oriented Systems," Bachelor Thesis, Denmark Technical University, Denmark, 2006.

[8] W. S. Jawadekar, *Software Engineering: Principles and Practice*, New Delhi: Tata McGraw-Hill, 2004.

[9] L. H. Rosenberg, *Software Quality Metrics for Object Oriented System Environments*, Greenbelt Maryland: NASA Goddard Space Flight Center, 1995.

[10] M. R. Barbacci, *Software Quality Atributes*: *Modifiability and Usability*, Pittsburgh: Carnegie Mellon University, 2004.

[11] K. Letelay & S. N. Azhari, "Evaluasi Kualitas Perangkat Lunak dengan Metrics Berorientasi Objek", *Seminar Nasional Informatika 2012 (semnas IF 2012)*, vol. 1, 2012.

[12] A. Kusjani & B. Bednar, "Penggunaan Program CKJM untuk Analisis Paket Remote Method Invocation", *TEKNOMATIKA*, vol. 8, 2015.

[13] W. M. Marpaung, "Pembuatan Kakas Bantu Pengukuran Kualitas Perangkat Lunak pada Kode Pemrograman Java," Undergraduate Thesis, Institut Teknologi Sepuluh November, Surabaya, Indonesia, 2015.

[14] A. Chhikara & R. S. Chhillar, "Analyzing the Complexity of Java Programs using Object-Oriented Software Metrics", *IJCSI International Journal of Computer Science Issues*, vol. 9, 2012.

[15] B. Henderson-Sellers, *Software Metrics*. New Jersey: Prentice-Hall, 1996.

[16] Rhamdani, "Evaluasi Kualitas Perangkat Lunak Berorientasi Objek", Undergraduate Thesis, Institut Pertanian Bogor, Bogor, Indonesia, 2008.