

Fuzzy Model for Predicting Reusability Factor of a Software

Model Fuzzy untuk Memprediksi Faktor *Reusability* Sebuah Perangkat Lunak

Endang Suryawati*

Pusat Penelitian Informatika
Lembaga Ilmu Pengetahuan Indonesia
Komplek LIPI, Jalan Cisitu No 21/154D Bandung 40135
Jawa Barat, Indonesia

Abstract

This paper describes the use of object oriented model to predict the quality of a software based on the factors of reusability. This model was constructed using Mamdani Fuzzy Inference with a Centroid technique for defuzzification process. Five parameters of CK Metrics that relate to reusability factor is used as an input of the model. The simulation results show that the output (reusability) is well associated with the established rules.

Key Words: object oriented metrics, reusability, Mamdani fuzzy inference, defuzzification, CK metrics

Abstrak

Tulisan ini menjelaskan tentang penggunaan model metrik berbasis obyek untuk memprediksi kualitas sebuah perangkat lunak berdasarkan faktor *reusability*. Model dibangun dengan menggunakan Inferensi Fuzzy Mamdani dengan teknik Centroid untuk proses defuzzifikasi. Lima parameter CK Metrics yang terkait dengan faktor *reusability*, digunakan sebagai *input* model. Hasil simulasi memperlihatkan *output* (*reusability*) yang bersesuaian dengan aturan-aturan yang dibentuk.

Kata kunci: metrik berbasis obyek, *reusability*, mesin penalaran *fuzzy* Mamdani, defuzzifikasi, metrik CK

1. PENDAHULUAN

Dalam beberapa dekade terakhir ini, teknologi berbasis obyek menjadi sangat populer dan diminati di lingkungan pengembangan perangkat lunak. Hal ini terkait dengan konsep *encapsulation* dan *inheritance* yang memudahkan dalam pemeliharaan dan pengembangan sistem selanjutnya. Akan tetapi, kemudahan ini tentunya tidak terlepas dari upaya-upaya untuk menghasilkan perangkat lunak yang berkualitas. Kualitas perangkat lunak sangat ditentukan oleh kualitas disain yang dilakukan pada fase awal dari pengembangan perangkat lunak [1]. Fase disain hanya mengambil sedikit waktu pengerjaan dibanding total keseluruhan fase

dalam proses pengembangan perangkat lunak [2]. Dalam kondisi disain yang tidak optimum, fase yang kecil ini dapat menimbulkan upaya besar untuk perbaikan kesalahan disain pada sebagian besar fase selanjutnya [2].

Penggunaan *Object-Oriented Metrics* (OO Metrics) sebagai media pengukur kualitas disain pada fase awal pengembangan, memberikan dampak positif bagi pengembang dalam menentukan kualitas disain [1, 3]. Disain yang optimum, yang dapat meminimalkan kompleksitas perangkat lunak [4, 5] tentunya akan menghasilkan produk akhir yang berkualitas.

Sejumlah OO Metrics telah banyak diusulkan dan dikembangkan oleh sejumlah peneliti. Beberapa di antaranya CK metrics, MOOD metrics dan QMOOD metrics [3, 4]. Telah dilakukan perbandingan beberapa OO metrics seperti CK metrics, MOOSE metrics, QMOOD metrics, GQM metrics dan EMOOSE metrics [5]. Dari sekian banyak metrics yang telah ada dan berkembang terus, CK metrics [6] yang paling banyak diusung oleh banyak peneliti di ranah OO metrics [3, 7–9]. Sebagian

*Corresponding Author. Tel: +6222-2504711

Email: endang@informatika.lipi.go.id

Received: 2 Apr 2013; revised: 14 Apr 2013; accepted: 23 May 2013

Published online: 28 May 2013

Corresponding editor: Tigor Nauli
(tigor@informatika.lipi.go.id)

© 2013 INKOM 2013/13-NO221

ada yang menggunakan CK *metrics* secara utuh atau memodifikasinya dengan beberapa parameter *metrics* tambahan untuk melengkapi kebutuhan disainnya. Secara resmi, tidak ada penetapan *metrics* standar, hanya bagaimana para peneliti mencari parameter-parameter *metrics* yang tepat dan memodifikasinya untuk mengukur properti disain OO [10].

Properti disain yang diukur adalah *methods*, *claseses*, *cohesion*, *coupling*, *modularity*, *encapsulation* dan *inheritance*. Setiap *metrics* mengukur properti disain perangkat lunak dan merefleksikannya dengan faktor kualitas perangkat lunak, seperti *complexity*, *reusability* dan lain-lain [10]. Ada beberapa faktor kualitas perangkat lunak yang dapat dijadikan referensi sesuai kebutuhan, seperti *reusability*, *perfomance*, *extensibility* dan *manageability* [7]. ISO 9126-1 mengeluarkan *functionality*, *reliability*, *usability*, *efficienty*, *maintainability* dan *portability* sebagai model kualitas perangkat lunak. Ada pula yang mengusung enam faktor kualitas perangkat lunak yaitu *efficiency*, *complexity*, *understandability*, *reusability*, *maintainability* dan *testability* [11].

Kualitas yang diinginkan dari hasil disain adalah sebuah perangkat lunak yang sesuai dengan spesifikasinya dan memiliki efek *reuse* pada komponen-komponennya [7]. *Reuse* memungkinkan komponen dapat digunakan kembali, sekalipun oleh aplikasi yang berbeda tanpa harus melakukan pengembangan dari awal [12]. Ini menunjukkan perangkat lunak memiliki *reusability* tinggi yang berdampak pada peningkatan produktivitas [13].

Beberapa peneliti mengusulkan model kualitas perangkat lunak untuk memprediksi nilai *reusability* yang dimilikinya, melalui beberapa pendekatan. Beberapa pendekatan tersebut adalah model *reusability* melalui pendekatan *Neural Network* [12], model *reusability* melalui pendekatan *Clustering* [13] dan model *reusability* dengan pendekatan *fuzzy logic* [14]. Semua model tersebut menggunakan parameter Ck *Metrics* sebagai input.

Penentuan kualitas perangkat lunak melalui beberapa pendekatan tersebut akan menghasilkan pengelompokan dalam pemeringkatan kualitas perangkat lunak. Pemeringkatan tersebut terbagi atas tiga kelompok, yaitu kelompok bernilai *low*, keompok bernilai *medium* dan kelompok bernilai *high*. Sebagai contoh, nilai kualitas yang diperoleh adalah berupa 60% *low* atau 40% *medium*, 70% *high* atau 30% *medium* dan seterusnya.

Tulisan ini akan mengusung model *reusability* dengan pendekatan *fuzzy logic*. Meskipun pendekatan ini telah dilakukan [14, 15], akan tetapi ada sedikit perbedaan dalam menentukan nilai *threshold* yang dirujuk untuk setiap parameter CK

Metrics. Diketahui bahwa seluruh parameter CK *Metrics* dimasukkan dalam himpunan keanggotaan input dari sistem inferensi *fuzzy* Mamdani yang akan digunakan. Nilai *threshold* yang dirujuk akan mempengaruhi batasan himpunan keanggotaan input dan output.

2. CK METRICS THRESHOLDS DAN FAKTOR KUALITAS PERANGKAT LUNAK

CK *Metrics*, smemiliki 6 parameter pengukur properti disain, yaitu WMC (*Weight Methods per Class*), RFC (*Response sets For a Class*), LCOM (*Lack of Cohesion of Methods*), CBO (*Coupling Between Classes*), DIT (*Dept of Inheritance*) dan NOC (*Number Of Children*). Ada beberapa referensi yang digunakan pada peneliti dalam memilih nilai *threshold* CK *Metrics*. Tabel I menunjukkan nilai *threshold* yang digunakan [16]. Sementara [17] menetapkan nilai *threshold* seperti diperlihatkan oleh Tabel II.

Tabel I. *Threshold* CK Versi [16]

CK Metrics	Threshold
WMC	0 - 15
RFC	0 - 35
LCOM	0 - 1
CBO	0 - 8
DIT	0 - 6
NOC	0 - 6

Sumber [16]

Tabel II. *Threshold* CK Versi NASA Rosenberg

CK Metrics	Threshold	
	Optimum	Maksimum
WMC	20	100
RFC	[50,100]	222
LCOM	[0,1]	-
CBO	5	24
DIT	3	6
NOC	-	5

Tabel III. *Threshold* CK Versi RefactorIT

CK Metrics	Threshold	
	Min-Maks	Optimum
WMC	0 - 50	-
RFC	0 - 50	-
LCOM	[0,1]	-
CBO	-	-
DIT	0-5	[2,3]
NOC	0-10	-

Dalam penetapan nilai *threshold* CK *Metrics*, penulis merujuk kepada *threshold* gabungan versi NASA Rosenberg dan versi RefactorIT. Versi RefactorIT digunakan penulis untuk memangkas nilai *threshold* Rosenberg yang dirasa terlalu tinggi untuk diterapkan pada aplikasi sederhana. Misalnya, penetapan nilai *threshold* WMC yang terlalu tinggi, dapat memungkinkan pengembang untuk terjebak dalam ketidakpedulian terhadap efek dari peningkatan *complexity* yang mempengaruhi penurunan *reusability*. Konfigurasi nilai *threshold* yang digunakan penulis dapat dilihat pada Tabel IV. Sementara itu, konfigurasi nilai *threshold* yang digunakan oleh Mago. J. Et al. [15], diperlihatkan pada Tabel V.

Tabel IV. *Threshold* CK Versi Penulis

CK Metrics	Threshold
WMC	0 - 55
RFC	0 - 55
LCOM	0 1.1
CBO	0 - 25
DIT	0 9.5
NOC	0 - 10

Tabel V. *Threshol* CK Versi [15]

CK Metrics	Threshold
WMC	0 - 50
RFC	0 -100
LCOM	0 - 1
CBO	0 - 50
DIT	0 - 10
NOC	0 - 30

Tabel VI. Hubungan Paramater CK *Metrics* dengan Faktor Kualitas Perangkat Lunak

CK Metrics	Faktor Kualitas
WMC	<i>Maintainability</i> , <i>Understandability</i> , <i>Usability</i> , <i>Reusability</i>
RFC	<i>Understandability</i> , <i>Usability</i> , <i>Testability</i>
LCOM	<i>Reusability</i> , <i>Efficiency</i>
CBO	<i>Reusability</i> , <i>Efficiency</i>
DIT	<i>Reusability</i> , <i>Efficiency</i> , <i>Understandability</i> , <i>Testability</i>
NOC	<i>Reusability</i> , <i>Efficiency</i> , <i>Testtability</i>

Setiap paramater CK *Metrics* memiliki keterkaitan dengan faktor kualitas perangkat. Sebuah faktor kualitas dapat dipengaruhi oleh satu atau lebih paramater CK. Sementara itu sebuah paramater CK dapat mempengaruhi satu atau lebih faktor kualitas, seperti diperlihatkan oleh Tabel VI. Dari tabel tersebut jelas bahwa paramater-paramater CK *Metrics* yang terkait dengan faktor kualitas *reusability* adalah WMC, LCOM, CBO, DIT dan NOC.

3. PEMBUATAN MODEL *REUSABILITY* DENGAN INFERENSI FUZZY MAMDANI

Inferensi *fuzzy* adalah pembuatan formulasi proses pemetaan input ke bagian output dengan menggunakan logika *fuzzy*. Tahapannya utamanya meliputi pendefinisian fungsi keanggotaan input dan output serta pembuatan basis pengetahuan dengan menggunakan aturan (*rules*) *fuzzy if-then-else*.

3.1 Pendefinisian Fungsi Keanggotaan Input dan Output

Berdasarkan Tabel VI, maka variabel *fuzzy* yang dimodelkan adalah lima paramater CK *Metrics* sebagai lima variabel input sistem dan paramater *reusability* sebagai variabel output sistem. Setiap paramater CK *Metrics* memiliki tiga himpunan *fuzzy*, yaitu *low*, *medium* dan *high*. Begitu pula dengan paramater *reusability* memiliki tiga himpunan yaitu *low*, *medium* dan *high*. Masing-masing himpunan dipetakan terhadap derajat keanggotaan dari 0 sampai dengan 1. Tipe dari fungsi keanggotaan untuk kelima variabel input *metrics* adalah fungsi *trapezoidal* dengan sintaks seperti pada persamaan 1 [15].

$$f(x) = trapmf|x, (abcd)| \tag{1}$$

Kurva *trapezoidal* [15] tersebut adalah fungsi dari sebuah vektor, x yang tergantung pada empat paramater skalar a, b, c dan d, sebagaimana diperlihatkan oleh persamaan (2),(3),(4) dan (5). Nilai a, b, c dan d disesuaikan dengan batasan himpunan keanggotaan yang dimiliki oleh setiap paramater input CK *Metrics*.

$$f(x; abcd) = 0, x \leq a \tag{2}$$

$$f(x; abcd) = \frac{x - a}{b - a}, a \leq x \leq b \tag{3}$$

$$f(x; abcd) = \frac{d - x}{d - c}.c \leq x \leq d \tag{4}$$

$$f(x; abcd) = 0, d \leq x \leq a \quad (5)$$

Untuk melengkapi fungsi *trapezoidal* tersebut, segera ditentukan nilai-nilai paramater a, b, c dan d untuk setiap variabel input CK *Metrics*, yang diperlihatkan oleh Tabel VII, VIII, IX, X dan XI. Sementara itu, bentuk kurva keanggotaan untuk setiap variabel input CK *Metrics* diperlihatkan oleh Gambar 1, 2, 3, 4 dan 5.

Tabel VII. *Weight Methods per Class*

WMC	a	b	c	d
low	0	5	20	25
medium	20	25	40	45
high	40	45	50	55

Tabel VIII. *Lack of Cohesion of Methods*

LCOM	a	b	c	d
low	0	0.1	0.5	0.6
medium	0.5	0.6	0.8	0.9
high	0.8	0.9	1	1.1

Tabel IX. *Coupling Between Classes*

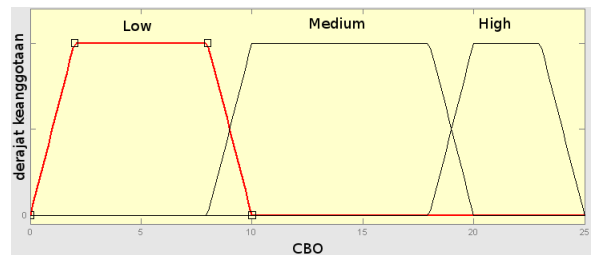
CBO	a	b	c	d
low	0	2	8	10
medium	8	10	18	20
high	18	20	23	25

Tabel X. *Depth of Inheritance*

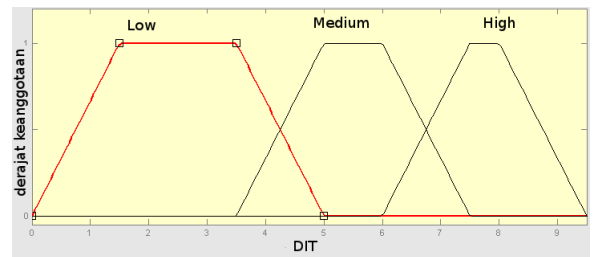
DIT	a	b	c	d
low	0	1.5	3.5	5
medium	3.5	5	6	7.5
high	6	7.5	8	9.5

Tabel XI. *Number of Children*

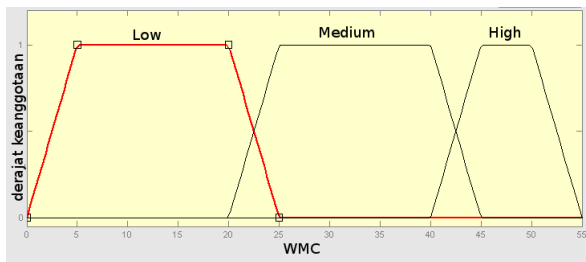
NOC	a	b	c	d
low	0	1	5	6
medium	5	6	7.5	8.5
high	7.5	8.5	9	10



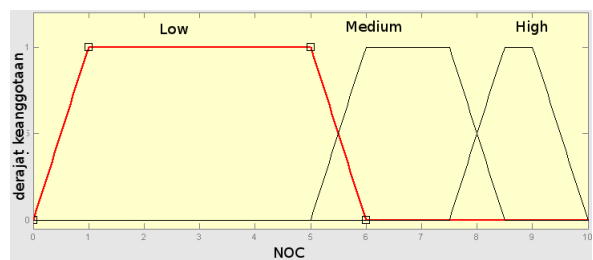
Gambar 3. Fungsi Keanggotaan Input CBO



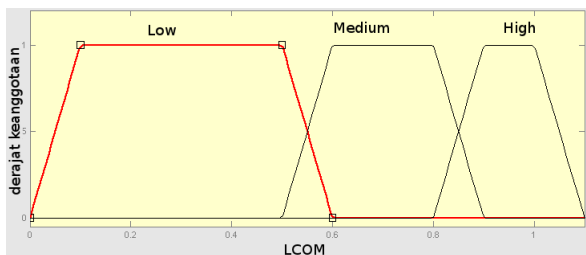
Gambar 4. Fungsi Keanggotaan Input DIT



Gambar 1. Fungsi Keanggotaan Input WMC



Gambar 5. Fungsi Keanggotaan Input NOC



Gambar 2. Fungsi Keanggotaan Input LCOM

Dari kelima kurva input *metrics* tersebut dapat ditentukan kisaran nilai input CK *Metrics* untuk kategori *low*, *medium* dan *high*. Kisaran nilai-nilai tersebut adalah sebagai berikut:

$$WMC(low) = 1 \leq x < 22.5 \quad (6)$$

$$WMC(medium) = 22.5 \leq x < 42.5 \quad (7)$$

$$WMC(high) = 42.5 \leq x \leq 52.5 \quad (8)$$

$$LCOM(low) = 0 \leq x < 0.55 \quad (9)$$

$$LCOM(medium) = 0.55 \leq x < 0.85 \quad (10)$$

$$LCOM(high) = 0.85 \leq x \leq 1.1 \quad (11)$$

$$CBO(low) = 0 \leq x < 9 \quad (12)$$

$$CBO(medium) = 9 \leq x < 19 \quad (13)$$

$$CBO(high) = 19 \leq x \leq 25 \quad (14)$$

$$DIT(low) = 0 \leq x < 4.2 \quad (15)$$

$$DIT(medium) = 4.2 \leq x < 6.8 \quad (16)$$

$$DIT(high) = 6.8 \leq x \leq 9.5 \quad (17)$$

$$NOC(low) = 0 \leq x < 5.5 \quad (18)$$

$$NOC(medium) = 5.5 \leq x < 8 \quad (19)$$

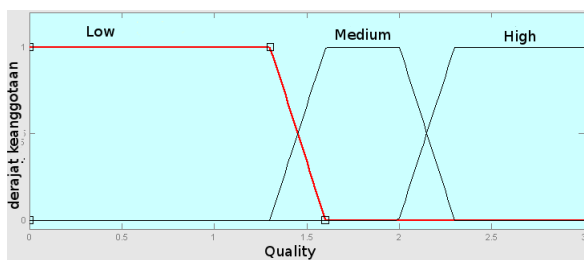
$$NOC(high) = 8 \leq x \leq 10 \quad (20)$$

Setelah mendefinisikan fungsi keanggotaan lima paramater CK *Metrics* sebagai input sistem, maka kemudian mendefinisikan fungsi keanggotaan dari output sistem. Dengan menggunakan tipe fungsi keanggotaan yang sama dengan input *metrics*, maka bentuk kurva keanggotaan output (*reusability*) adalah seperti terlihat pada Gambar 6, dengan kisaran nilai output adalah sebagai berikut.

$$Output(low) = x < 1.45 \quad (21)$$

$$Output(medium) = 1.45 \leq x \leq 2.2 \quad (22)$$

$$Output(high) = x > 2.2 \quad (23)$$



Gambar 6. Fungsi Keanggotaan Output (*Reusability*)

3.2 Pembuatan Basis Pengetahuan

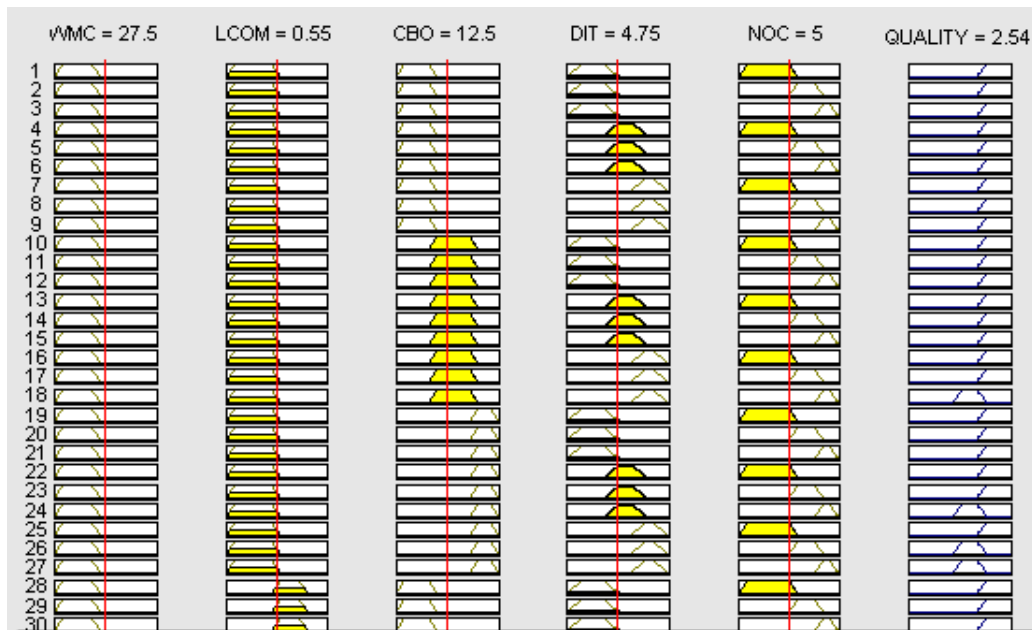
Basis pengetahuan disusun oleh struktur aturan *fuzzy* yang terbagi dalam dua bagian, yaitu bagian *if-part* dan bagian *then-part*. *IF* $\langle antecedent \rangle$ *THEN* $\langle consequent \rangle$. *Antecedent* adalah kondisi dan *consequent* adalah kesimpulan. Aturan tersebut merupakan kombinasi dari semua kondisi input yang mungkin untuk menghasilkan output atau kesimpulan yang bersesuaian.

Terdapat enam input *metrics*, di mana setiap input memiliki tiga kemungkinan kondisi yaitu *low*, *medium* dan *high*. Sehingga total aturan yang mungkin adalah sebanyak 243 aturan. Contoh aturan pertama adalah *IF* (*WCM is LOW*) and (*LCOM is LOW*) and (*CBO is LOW*) and (*DIT is LOW*) and (*NOC is LOW*) *THEN REUSABILITY is HIGH*. Contoh aturan berikutnya diperlihatkan oleh Tabel XII.

Tabel XII. Cuplikan aturan *fuzzy*

No	Aturan
1	IF(WCM is LOW) and (LCOM is LOW) and (CBO is LOW) and (DIT is LOW) and (NOC is MEDIUM) THEN <i>REUSABILITY</i> is HIGH
2	IF(WCM is LOW) and (LCOM is LOW) and (CBO is LOW) and (DIT is LOW) and (NOC is LOW) THEN <i>REUSABILITY</i> is HIGH
...	...
18	IF(WCM is LOW) and (LCOM is LOW) and (CBO is MEDIUM) and (DIT is HIGH) and (NOC is HIGH) THEN <i>REUSABILITY</i> is MEDIUM
...	...
120	IF(WCM is MEDIUM) and (LCOM is MEDIUM) and (CBO is MEDIUM) and (DIT is LOW) and (NOC is HIGH) THEN <i>REUSABILITY</i> is MEDIUM
121	IF(WCM is MEDIUM) and (LCOM is MEDIUM) and (CBO is MEDIUM) and (DIT is MEDIUM) and (NOC is LOW) THEN <i>REUSABILITY</i> is HIGH
...	...
242	IF(WCM is HIGH) and (LCOM is HIGH) and (CBO is HIGH) and (DIT is HIGH) and (NOC is MEDIUM) THEN <i>REUSABILITY</i> is LOW
243	IF(WCM is HIGH) and (LCOM is HIGH) and (CBO is HIGH) and (DIT is HIGH) and (NOC is HIGH) THEN <i>REUSABILITY</i> is LOW

Keenam paramater input CK *Metrics* memiliki kecenderungan semakin memiliki nilai rendah (*LOW*) maka semakin baik kualitas. Semakin tinggi nilai input maka semakin buruk kualitas. Sehingga dalam aturan ini diberikan pembobotan tertinggi adalah saat kondisi *LOW* dan terendah adalah saat kondisi *HIGH*. Urutan pembobotan adalah *LOW*



Gambar 7. Simulasi Aturan Fuzzy Hasil Defuzzifikasi)

= 3, *MEDIUM* = 2 dan *HIGH* = 1. Pembobotan ini digunakan untuk menentukan kualitas dari output *reusability* (Q) [15], yang diperoleh melalui persamaan:

$$Q = \frac{WMC + LCOM + CBO + DIT + NOC}{5} \tag{24}$$

Sebagai contoh, implementasi persamaan (24) untuk aturan nomor 18 pada Tabel XII adalah sebagai berikut:

$$\begin{aligned} Q &= \frac{L + L + M + H + H}{5} \\ &= \frac{(3 + 3 + 2 + 1 + 1)}{5} \\ &= 2(\text{medium}) \end{aligned}$$

4. SIMULASI DAN ANALISA

Melalui perangkat lunak Matlab, maka 243 aturan yang telah ditetapkan, selanjutnya melewati tahapan *defuzzifikasi* dengan teknik *centroid*. Hasil *defuzzifikasi* menampilkan kombinasi input output dalam kondisi awal. Nilai-nilai input yang ditampilkan mengambil nilai tengahnya masing-masing dan nilai outputnya menunjukkan kualitas dengan kategori *high*, sesuai aturan yang ditetapkan. Hasil simulasi dapat dilihat pada Gambar 7.

Dari kondisi awal (Gambar 7), maka dicoba untuk melakukan perubahan pada salah satu input secara bergantian untuk melihat efek perubahan pada

outputnya. Ketika nilai WMC (Tabel XIII) dinaikkan mendekati nilai terburuknya, maka nilai kualitas (Q) menurun. Saat nilai WMC kembali diturunkan menuju nilai optimumnya, maka kualitas kembali membaik. Hal yang sama dilakukan terhadap nilai NOC (Tabel XIV) dan nilai DIT (Tabel XV).

Tabel XIII. Efek Perubahan Nilai WMC

	WMC=27.5	WMC =43.7	WMC = 20
Q	2.54	2.09	2.54

Tabel XIV. Efek Perubahan Nilai NOC

	NOC=5	NOC =10	NOC = 3
Q	2.54	1.5	2.54

Tabel XV. Efek Perubahan Nilai DIT

	DIT=4.75	DIT =7	DIT = 6
Q	2.54	2.11	2.54

5. KESIMPULAN

Dari hasil simulasi, secara umum diperoleh output yang sesuai dengan aturan *fuzzy* yang dibuat. Nilai WMC yang tinggi karena jumlah metode per kelas yang tinggi, menandakan disain memiliki kompleksitas tinggi., sehingga *reusability* menurun. Sementara itu, nilai NOC yang tinggi pada dasarnya adalah baik, karena menaikkan faktor *reusability*.

Akan tetapi jika nilai NOC terlalu tinggi, maka faktor kompleksitas akan menjadi tinggi. Nilai DIT yang tinggi adalah baik untuk *reuse*, akan tetapi jika terlalu tinggi, akan semakin sulit untuk memahami sistem karena terlalu banyak metode yang diturunkan (*high complexity*).

Dengan kondisi *high complexity* akibat nilai-nilai DIT dan NOC yang terlalu tinggi, maka secara otomatis akan menurunkan nilai *reusability*. Untuk itu diperlukan kisaran nilai-nilai yang optimum untuk parameter NOC, DIT, WMC dan yang lainnya, agar nilai *reusability* tetap terjaga.

Meskipun hasil simulasi model aturan *fuzzy* ini secara umum menunjukkan kesesuaian dengan apa diinginkan, akan tetapi perlu dilakukan perbandingan dengan model pendekatan lain atau tools yang dapat dijadikan bahan perbandingan. Sehingga dapat mengoptimalkan model *fuzzy* ini dan dapat diterapkan untuk mengukur aplikasi-aplikasi yang telah ada.

Daftar Pustaka

- [1] R. Subramanyam and M. S. Krishnan, "Empirical analysis of ck metrics for object-oriented design complexity: Implications for software defects," *IEEE Trans. Softw. Eng.*, vol. 29, no. 4, pp. 297–310, Apr. 2003.
- [2] S. Ragab and H. Ammar, "Object oriented design metrics and tools a survey," in *Informatics and Systems (INFOS), 2010 The 7th International Conference on*, 2010, pp. 1–7.
- [3] M. P. C. Neelamegam, "A survey object oriented quality metrics," *Global Journal of Computer Science and Technology*, pp. 183–186, 2009.
- [4] A. Deepak, K. Pooja, T. Alpika, S. Shipra, and S. Sanchika, "Software quality estimation through object oriented design metrics," *International Journal of Computer Science and Network Security*, pp. 100–104, 2011.
- [5] S. Amit and D. K. Sanjay, "Comparison of software quality metrics for object-oriented system," *International Journal of Computer Science & Management Studies*, pp. 12–24, 2012.
- [6] S. Chidamber and C. Kemerer, "A metrics suite for object oriented design," *Software Engineering, IEEE Transactions on*, vol. 20, no. 6, pp. 476–493, 1994.
- [7] H. Anand, "Software quality analysis by object oriented approach in design phase using uml," *International Journal of Computer Technology and Electronics Engineering*, pp. 48–52, 2011.
- [8] C. Arti and R. Chhillar, "Analyzing the complexity of java programs using object - oriented software metrics," *International Journal of Computer Science Issues*, pp. 364–372, 2012.
- [9] S. Ahmed and Q. Abrar, "Analysis of inconsistencies in object oriented metrics," *Journal of Software Engineering and Applications*, pp. 123–128, 2011.
- [10] T. Mythii and J. Swthi, "Quality metrics tool for object oriented programming," *International Journal of Computer Theory and Engineering*, pp. 1793–8201, 2010.
- [11] Rosenberg, H. Linda, and L. W. Hyatt, "Software quality metrics for object-oriented environments," *Crosstalk Journal*, 1997.
- [12] P. S. Sandhu, H. Kaur, and A. Singh, "Modeling of reusability of object oriented software system," *World Academy of Science, Engineering and Technology*, vol. 56, no. 32, 2009.
- [13] S. Anju, S. Parvinder, G. Vikas, and A. Sanyam, "Prediction of reusability of object oriented software systems using clustering approach," *Academy of Science, Engineering and Technology*, no. 43, pp. 853–856, 2010.
- [14] P. Singh Sandhu and H. Singh, "A fuzzy-inference system based approach for the prediction of quality of reusable software components," in *Advanced Computing and Communications, 2006. ADCOM 2006. International Conference on*, 2006, pp. 349–352.
- [15] S. K. M. Jagmohan, K. Sarabjit, "Fuzzy model to analyze and interpret object oriented software design," *International Journal of Electrical, Electronics and Computer Engineering*, no. 1, pp. 41–46, 2012.
- [16] P. E. L. E. Chandra, "Class break point determination using ck metrics thresholds," *Global Journal of Computer Science and Technology*, no. 14, pp. 73–77, 2010.
- [17] G. T. Nair and R. Selvarani, "Software reusability estimation model using metrics governing design architecture," *Knowledge Engineering for Software Development Life Cycles: Support Technologies and Applications*, p. 196, 2011.