

Pemetaan Secara Sistematis Pada Metrik Kualitas Perangkat Lunak

Mamluatul Hani'ah¹, Yogi Kurniaawan², Umi Laili Yuhana³

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Jl. Teknik Kimia, Gedung Teknik Informatika, Kampus ITS Sukolilo, Surabaya, 60111, Jawa Timur
E-mail: ¹mamluatul14@mhs.if.its.ac.id, ²yogi.kur@ub.ac.id, ³yuhana@cs.its.ac.id

Masuk: 6 Mei 2015; Direvisi: 27 Mei 2015; Diterima: 1 Juni 2015

Abstract. *Software quality assurance is one method to increase quality of software. Improvement of software quality can be measured with software quality metric. Software quality metrics are part of software quality measurement model. Currently software quality models have a very diverse types, so that software quality metrics become increasingly diverse. The various types of metrics to measure the quality of software create proper metrics selection issues to fit the desired quality measurement parameters. Another problem is the validation need to be performed on these metrics in order to obtain objective and valid results. In this paper, a systematic mapping of the software quality metric is conducted in the last nine years. This paper brings up issues in software quality metrics that can be used by other researchers. Furthermore, current trends are introduced and discussed.*

Keywords: *Software Quality, Software Assesment, Metric*

Abstrak. *Penjaminan kualitas suatu perangkat lunak merupakan salah satu cara meningkatkan kualitas suatu perangkat lunak. Metrik kualitas perangkat lunak merupakan bagian dari model pengukuran kualitas perangkat lunak. Model kualitas perangkat lunak memiliki jenis yang sangat beragam, sehingga metrik kualitas perangkat lunak menjadi semakin beragam jenisnya. Beragamnya jenis metrik pengukuran kualitas perangkat lunak memberikan permasalahan pemilihan metrik yang tepat agar sesuai dengan parameter pengukuran kualitas yang diinginkan. Permasalahan yang lain adalah validasi yang harus dilakukan terhadap metrik tersebut agar diperoleh hasil yang obyektif dan valid. Dalam makalah ini akan dilakukan pemetaan sistematis terhadap metrik pengukuran kualitas perangkat lunak pada sembilan tahun terakhir. Diharapkan dengan pemetaan sistematis akan dapat memunculkan permasalahan-permasalahan pada metrik kualitas perangkat lunak yang dapat digunakan sebagai penelitian untuk peneliti yang lain.*

Kata Kunci: *Kualitas Perangkat Lunak, Penjaminan Perangkat Lunak, Metrik*

1. Pendahuluan

Penjaminan kualitas dan manajemen proses secara aktif dapat meningkatkan kualitas suatu perangkat lunak. Peningkatan kualitas suatu perangkat lunak harus diukur secara objektif atau dapat diukur secara kuantitatif dari sebuah pengukuran kualitatif, salah satu cara untuk mengukur kualitas sebuah perangkat lunak adalah dengan menggunakan sebuah metrik. Metrik merupakan standar pengukuran atribut dari sebuah proses, produk, dan pelayanan perangkat lunak (Westfall, 2008).

Metrik pengukuran kualitas suatu perangkat lunak merupakan bagian dari model pengukuran perangkat lunak. Penelitian Wagner, dkk. (2009) menyatakan bahwa mayoritas model yang digunakan untuk menjaga kualitas perangkat lunak adalah model yang spesifik terhadap suatu organisasi atau perusahaan. Dengan penggunaan model yang beragam mengakibatkan bentuk-bentuk metrik pengukuran yang semakin beragam. Beragamnya metrik pengukuran tersebut juga menimbulkan permasalahan penggunaan metrik yang tepat untuk mendapatkan data pengukuran yang diinginkan. Tantangan lain yang ada adalah pemilihan metrik yang tepat dan sesuai dengan pengukuran kualitas yang diinginkan agar hasil

pengukuran dapat objektif dan sesuai. Setelah dilakukan pengukuran kualitas perangkat lunak tahap selanjutnya adalah diperlukan validasi terhadap hasil metrik pengukuran sebuah perangkat lunak. Dengan demikian hasil pengukuran dapat digunakan dan sesuai dengan teori pengukuran kualitas perangkat lunak.

Dari beberapa tantangan yang telah disebutkan perlu dilakukan pemetaan terhadap penggunaan metrik sebagai ukuran sebuah perangkat lunak. Pemetaan yang dilakukan harus sistematis, dimana pemetaan tersebut harus sesuai dengan rencana atau tahapan metode yang telah disusun dengan baik. Pemetaan sistematis dilakukan terhadap penelitian tentang metrik pengukuran perangkat lunak dalam kurun waktu sembilan tahun terakhir. Dengan melakukan penelitian pemetaan secara sistematis pada metrik kualitas perangkat lunak diharapkan dapat memunculkan permasalahan-permasalahan pada metrik kualitas perangkat lunak yang dapat digunakan sebagai penelitian untuk peneliti yang lain.

2. Tinjauan Pustaka

Metrik kualitas perangkat lunak adalah bagian dari metrik perangkat lunak yang fokus pada aspek kualitas produk, proses, dan proyek. Secara umum, metrik kualitas perangkat lunak akan lebih erat terkait dengan proses dan produk metrik dibandingkan dengan metrik proyek. Metrik produk sebuah perangkat lunak menggambarkan karakteristik produk seperti ukuran, kompleksitas, fitur desain, kinerja, dan tingkat kualitas.

Penelitian terhadap metrik kualitas perangkat lunak telah beberapa kali dilakukan. Pada penelitian Barkmann, dkk. (2009) dilakukan pengkajian validitas beberapa metrik kualitas perangkat lunak pada beberapa perangkat lunak *open-source*. Validasi metrik kualitas diperlukan agar metrik kualitas perangkat lunak dapat digunakan secara efisien dan dapat menjamin keefektifan dari metrik tersebut. Telah banyak penelitian yang membahas tentang validasi metrik perangkat lunak, akan tetapi seluruh penelitian kuantitatif dibatasi pada beberapa sistem kecil atau satu sistem besar.

Pada penelitian Wagner, dkk. (2009) dilakukan survei terhadap model-model yang sering digunakan pada perusahaan di Jerman. Penelitian tersebut dilakukan dengan cara wawancara pada empat perusahaan terpilih. Untuk mendapatkan tingkat cakupan yang maksimum dalam populasi penelitian, pewawancara yang dipilih dalam *cluster* berdasarkan organisasi yang berbeda (yang mengembangkan perangkat lunak dalam domain yang berbeda) dan dua peran yang berbeda: *general manager* dan manajer mutu. Dari penelitian tersebut didapatkan bahwa banyak perusahaan yang menggunakan model-model khusus yang sesuai dengan perusahaannya. Hal ini membuat metrik yang digunakan semakin berkembang sesuai dengan kebutuhan perusahaan.

3. Metode Penelitian

Penelitian ini menyajikan pemetaan secara sistematis terhadap metrik kualitas perangkat lunak. Tahapan penelitian ini terdiri dari beberapa langkah seperti yang ada pada Gambar 1.

3.1. Research Question

Untuk membuat penelitian ini tetap fokus maka ditetapkan empat RQ sebagai berikut: (1) RQ-1: Bagaimana sebaran penelitian tentang metrik kualitas perangkat lunak? (2) RQ-2: Bagaimana tipe dan ragam metrik pada pengukuran kualitas perangkat lunak? (3) RQ-3: Apa saja penggunaan metrik berdasarkan karakteristik kualitas ISO 9126? (4) RQ-4: Bagaimana validasi metrik pengukuran kualitas perangkat lunak?

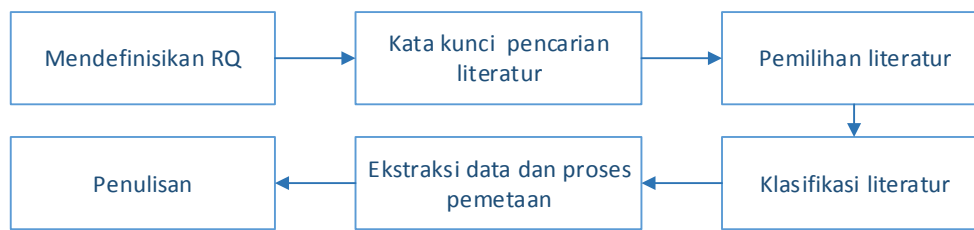
3.2. Strategi Pencarian Literatur

Proses pencarian literatur terdiri dari memilih perpustakaan digital yang digunakan untuk mencari literatur, mendefinisikan kata kunci pencarian, melaksanakan pencarian percontohan, menyempurnakan kata kunci pencarian dan mengambil hasil dari perpustakaan digital yang cocok dengan kata kunci pencarian. Pemilihan perpustakaan digital didasarkan pada

dua sumber rujukan yang dapat diakses melalui ITS selain itu banyak perguruan tinggi di Indonesia yang berlangganan pada sumber tersebut, yaitu *IEEE Computer Society Digital Library* (<http://ieeexplore.ieee.org>) dan *Science Direct* (<http://www.sciencedirect.com/>). Penulis tidak memasukkan *google scholar* karena dikhawatirkan beririsan dengan kedua sumber dikarenakan *google scholar* juga mengindeks kedua sumber tersebut.

Kata kunci pencarian dibangun berdasarkan langkah berikut: (1) Identifikasi istilah pencarian dari *research question*. (2) Identifikasi istilah pencarian dalam makalah yang relevan 'judul, abstrak dan kata kunci'. (3) Identifikasi sinonim dan ejaan alternatif. (4) Menggunakan istilah pencarian Boolean (AND dan OR). (5) Pencarian menggunakan *advanced search*.

Strategi pencarian dilakukan dengan menggunakan *advanced search* dan dengan langkah sebagai berikut (1) Langkah pertama dilakukan pencarian pada situs *IEEE explorer* dengan menggunakan kata kunci pencarian "(*software quality*) AND (*metrics OR assess* OR measure**)" dengan menggunakan kata kunci ini didapatkan hasil 148 literatur. (2) Langkah kedua dilakukan pencarian pada situs *Science Direct* dengan menggunakan kata kunci pencarian "*TITLE-ABSTR-KEY("software quality" AND (measure* OR metrics OR assess*)) AND LIMIT-TO (yearnav,"2014,2013,2012,2011,2010,2009,2008,2007,2006,2005") AND LIMIT-TO (contenttype, "1,2","Journal")*" dengan menggunakan kata kunci ini didapatkan hasil 114 literatur yang relevan termasuk didalamnya terdapat dua bab *e-book* dan tiga buah literatur yang tidak dapat diunduh. Sehingga hasil literatur yang bisa diunduh adalah 99 makalah.



Gambar 1. Tahapan Metode Penelitian

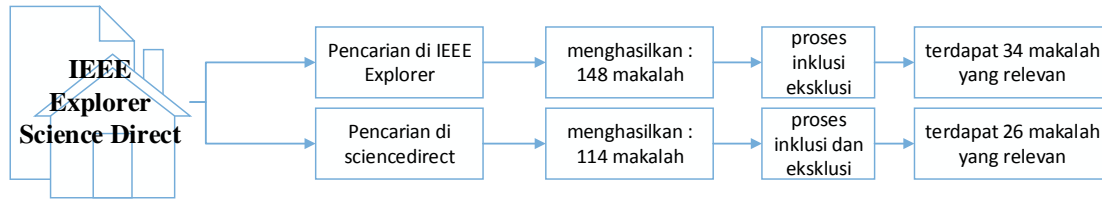
3.3. Pemilihan Literatur

Pemilihan literatur yang sesuai dengan RQ dalam pemetaan metrik kualitas perangkat lunak ini digunakan kriteria inklusi dan eksklusi sebagai berikut: (1) Kriteria Inklusi: (a) Penelitian yang berkaitan dengan metrik khususnya *product metric*. (b) Makalah metrik perangkat lunak yang di terbitkan sembilan tahun terakhir yaitu mulai tahun 2005 sampai 2014. (2) Kriteria Eksklusi: (a) Penelitian tentang metrik perangkat lunak yang tidak memiliki hasil eksperimen atau validasi empiris seperti makalah perbandingan dan makalah survei. (b) Penelitian yang membahas tentang teknik pemodelan dan bukan metrik perangkat lunak.

Setelah ditentukan kriteria inklusi dan eksklusi dilakukan proses seleksi kriteria dengan menggunakan tahapan berikut: (1) Membaca judul. Untuk makalah-makalah dengan judul yang tidak sesuai dengan tema dan RQ yang diangkat akan dieliminasi. (2) Membaca kata kunci, misal: *software quality*, *quality metrics*, *software metrics*. (3) Membaca abstrak. Proses pembacaan abstrak digunakan untuk melakukan seleksi terhadap makalah-makalah yang masuk kriteria eksklusi dan kriteria inklusi. (4) Membaca keseluruhan makalah. Proses ini bertujuan untuk mengelompokkan makalah kemudian mendapatkan skema klasifikasi dan pemetaan terhadap metrik kualitas perangkat lunak. Proses pencarian literatur sampai hasil inklusi dan eksklusi dapat dilihat pada Gambar 2.

4. Pembahasan

Pada pemetaan sistematis ini jumlah literatur yang digunakan setelah dilakukan pemilihan literatur pada proses sebelumnya adalah sebanyak 60 buah penelitian pada metrik kualitas perangkat lunak.



Gambar 2. Tahapan Strategi Pencarian

4.1. RQ-1: Bagaimana sebaran penelitian tentang metrik kualitas perangkat lunak?

Penelitian tentang metrik kualitas perangkat lunak memang tidak sebanyak penelitian pada bidang model kualitas perangkat lunak. Akan tetapi dalam sembilan tahun terakhir penelitian tentang metrik kualitas perangkat lunak memiliki jumlah yang konstan di setiap tahunnya. Sebaran penelitian dapat dilihat pada Gambar 3.

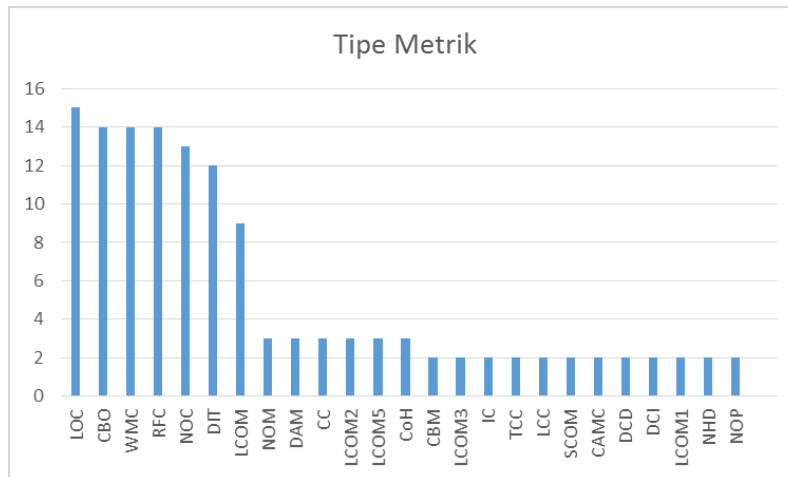


Gambar 3. Sebaran Penelitian Sembilan Tahun Terakhir

4.2. RQ-2: Bagaimana tipe dan ragam metrik pada pengukuran kualitas perangkat lunak?

Beberapa penelitian yang menggunakan lebih dari satu metrik untuk melakukan pengukuran terhadap sebuah perangkat lunak. Pada penelitian ini metrik yang dipetakan adalah metrik yang digunakan oleh lebih dari satu penelitian. Tipe metrik dan berapa banyak metrik tersebut digunakan dalam penelitian dapat dilihat pada Gambar 4.

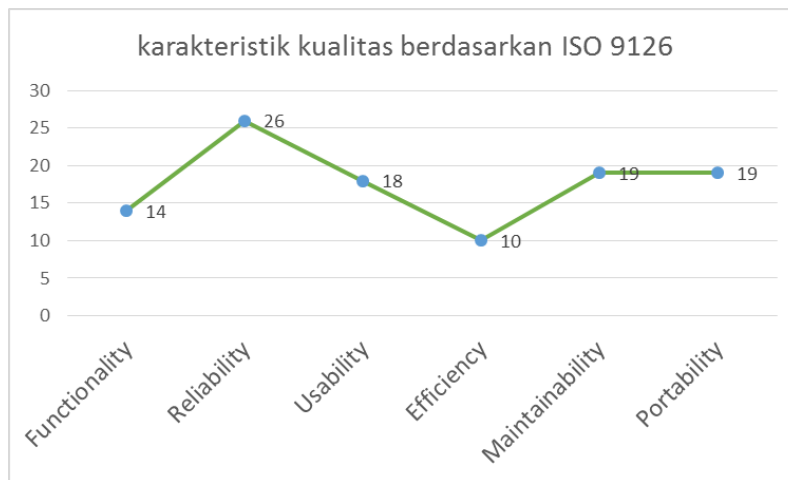
Gambar 4 dapat dilihat bahwa metrik yang banyak digunakan dalam penelitian merupakan metrik untuk perangkat lunak berbasis *object oriented*. Metrik-metrik tersebut adalah: (1) *Line of Code* (LOC) atau yang sering disebut *Source Lines of Code* SLOC merupakan setiap baris pada program tetapi bukan baris kosong ataupun komentar. LOC mencakup semua kode program mulai dari *header*, deklarasi variabel, dan kode utama yang dieksekusi. (Landman, dkk., 2014). (2) *Coupling Between Objects* (CBO) untuk setiap kelas adalah menghitung jumlah kelas yang dihubungkan dengan kelas tertentu. (3) *Weighted Methods per Class* (WMC) melakukan perhitungan terhadap kompleksitas dari semua *methods* yang ada dalam sebuah kelas. (4) *Response For a Class* (RFC) mengukur jumlah *methods* yang dieksekusi untuk memberikan respon ketika sebuah *object* dari kelas tersebut menerima pesan. (5) *Number of Children* (NOC) merupakan metrik yang mengukur jumlah *child* dari sebuah kelas. (6) *Depth of Inheritance Tree* (DIT) dalam *inheritance hierarchy* adalah jumlah maksimum dari node kelas sampai ke node *root*. (7) *Lack of Cohesion in Methods* (LCOM) mengukur perbedaan dari *methods* dalam setiap kelas dengan melihat *instance variable* atau atribut yang digunakan oleh *methods* tersebut. (Aggarwal, dkk., 2006).



Gambar 4. Tipe Metrik dan Jumlah Penelitian yang Menggunakannya

4.3. RQ-3: Apa saja penggunaan metrik berdasarkan karakteristik kualitas ISO 9126?

Berdasarkan ISO 9126 terdapat enam faktor metrik yaitu *maintainability*, *portability*, *functionality*, *usability*, *reliability*, *efficiency*. Strategi pengambilan data yang dilakukan adalah dengan membaca keseluruhan makalah dan mengambil tujuan dari makalah tersebut. Kemudian dibandingkan dengan subfaktor yang dimiliki oleh ISO 9126. Untuk penelitian yang menggunakan karakteristik kualitas berdasarkan ISO 9126 pada sembilan tahun terakhir ini dapat dilihat pada Gambar 5. Pada Tabel 1 telah dipetakan keseluruhan penelitian pada sembilan tahun terakhir berdasarkan karakteristik kualitas ISO 9126.



Gambar 5. Karakteristik Berdasarkan ISO 9126 dan Jumlah Penelitian yang Menggunakannya

Tabel 1. Pemetaan Makalah

No	Referensi	Metrik	Karakteristik berdasar ISO 9126
1	Hulse & Khoshgoftaar, 2008	Independent variables, Logical operators, Total lines of code, Executable lines of code, Unique operands, Total operands, Unique operators, Total operators, Cyclomatic complexity,	
2	Shim, dkk., 2008	Service Internal Metrics, Service External Metrics, System Metrics	Portability Usability, Efficiency
3	Pizzi, 2013	lines of code, the comment ratio, the true comment ratio ,the total number of operands, and operators, the total number of unique operands and operators	Maintainability
4	Ampatzoglou, dkk., 2012	DIT, NOC, MPC, RFC, LCOM, Range, DAC, WMPC, NOM, SIZE1, SIZE2	Maintainability
5	Ma, dkk., 2009	NOR, NOL, AIT-LAN, AFRC, Coh, AAFC, NMIS, AVAI	Reliability
6	Hsueh, dkk., 2008	QMOOD	maintainability, Portability
7	Akingbehin, 2005	KLOC	Reliability

8	Gomathy & Rajalakshmi, 2014	Task completion, WSIC, SS, SRP,SAM, Time on task, DSSS, NOVS, SRT, SFR	Portability, Usability, Efficiency, Functionality, Reliability
9	Oyetyan, dkk., 2013	Component In-Cycle, Depend On cycle, minimum number of cycle	Maintainability
10	Kohichi, dkk., 2012	LOC	Usability
11	Chirila & Cretu, 2011	NoOfPackages, NoOfClasses, NoOfInterfaces, NoOfClassifiers	Portability, Usability, Efficiency, Reliability, Maintainability
12	Costal & Franch, 2012	WMC, DIT, NOC, DAC, NOM, PIM, NIM, NIV, NCM, NCV, NOH, ANA, APPM, PK3, AO, DAM, DCC, MOA, DSC, NOP, NA,Ndep	
13	Sim, dkk., 2006	usability ranking	Usability
14	Chiu, 2009	LOC, CL, Tco, TCh, CC, DC, N, Nb, Nf, V(G), BW	
15	Hansen, dkk., 2011	ODR, ROU, WMC,AMC	
16	Al Dallal & Briand, 2010	SCC	Reliability
17	Thwin & Quah, 2005	WMC, DIT, NOC, CBO, RFC, IC, CBM, NOMA	Maintainability
18	Srivisut & Muenchaisri, 2007	NAdP, NAdAsP, NSAsP, NNSAsP, SJP, NOAsP	
19	Lubke, dkk., 2009	BPEL.test coverage metrics	Functionality, Reliability
20	Chirinos, dkk., 2005	MOSME	Functionality, Usability, Efficiency, Portability, Reliability, Maintainability
21	Catal, dkk., 2011	LOC, V(g), eV(g), iv(g), Iocode, IoComme, IOBlank	Reliability
22	Damm & Lundberg, 2007	POQ metric, PIQ metric, FST	Reliability
23	Gui, 2006	LCOM, TCC, Wtcoh	Portability
24	Pai & Dugan, 2007	WMC, CBO, RFC, SLOC, DIT, NOC, LCOM	Reliability
25	Alshayeb, 2009	DIT, NOC, CBO, RFC, FOUT, WMC, NOM, LOC, LCOM	Usability, Maintainability, Portability
26	Ikemoto, dkk., 2013	KLOC	Reliability
27	Selvarani, dkk., 2009	WMC, RFC, DIT	Reliability
28	Duque-Ramos, dkk., 2013	LCOM, WMC, DIT, NAC, NOC, RFC, NOM, PR, AR, CR, AN	Functionality, Maintainability
29	Pántos, dkk., 2008	LOC, ILOC, NPAR, McCC, WMC, NII, NOI, CBO	Portability, Usability, Reliability, Maintainability
30	Al Dallal, 2012	Cohesion based on member connectivity (CBMC), iCBMC	Reliability
31	Monga, dkk., 2014	LOC, LOCM, C, DIT, V, CBO, perCM, NOM	Portability, Usability, Efficiency, Maintainability
32	Al Dallal, 2011	LSCC, LCOM3, LCOM4, Coh, TCC, LCC, DCD, DCI, LCOM1, LCOM2, LCOM5, CC, SCOM, CAMC, NHD	Reliability
33	Binkley, 2009	TCC, LCC, CC, SCOM, CAMC, DCD, DCI, LCOM1, LCOM2, LCOM5, CoH, NHD	Reliability
34	Molina & Toval, 2009	WQM (Web Quality Metrics)	Usability
35	Sahraoui, dkk., 2010	LOC	Functionality, Portability, Maintainability
36	Yahaya, dkk., 2010	Incomplete result, Incorrect result, Unexpected results	Portability, Usability, Efficiency, Functionality, Reliability, Maintainability
37	Stefani & Xenos, 2009	LOC, TM maturity, TN nature, TS team size	Functionality, Reliability
38	Mishra & Sureka, 2014	CHN, NCL, NFL, NCS, ANF, ACF	Functionality
39	Chaparro, dkk., 2014	LCOM5, CBO, RFC, LCOM2, DAC, MPC, LOC, NOM, CYCLO, NOC, DIT	
40	Conejero, dkk., 2012	CBO, DIT, LOC, WMC, RFC, LCOM, CE, NPM, CBM, WMC, AMC, CAM, LCOM3, MAX_CC, CA, DAM, MOA, MFA, IC, NOC, AVG_CC	Maintainability
41	Khaer, dkk., 2008	WMC, CBO, RFC, SLOC, DIT, NOC, LCOM	Portability, Usability, Functionality, Maintainability
42	Nagappan, dkk., 2005	SLOC, CBO, DIT, WMC, TLOC	Reliability
43	Goyal & Joshi, 2014	DSC, NOH, ANA, DAM, DCC, CAM, MOA, MFA, NOP, NOM, CIS	Portability, Usability, Efficiency, Functionality
44	Steidl, dkk., 2013	coherence Coefficient (c coeff)	Functionality
45	Domínguez-Mayo, dkk., 2012	Matrix of Influences	Functionality, Usability, Efficiency, Portability, Maintainability, Reliability
46	Ahn, dkk., 2013	DCM	
47	Zhou & Liu, 2007	LOC, NQS, SC, AC, CD, EIL, EFD	Portability, Usability
48	Khoshgoftaar, dkk., 2005	CBO, RFC, LOC, DCM	
49	Rotaru & Dobre, 2005	NoC, CBO, LCOM, RFC, DIT, WMC	Portability
50	Takai, dkk., 2011	NOV, CNV, CNUV, ANV, MNV, NVPL, VOV, MTOV, MTFV, VTOV, VTFV, NVPM, NFVPM	Reliability, Maintainability
51	Nikolik, 2012	Artifact testing metrics	Reliability
52	Ryan & Rossi, 2005	ECS, SOS, OMS, EMS, NEI, SSP, NI, ET, IT, MIT, MCT, NC, NU, MC, MU, PC, PU	Efficiency
53	Zhang, dkk., 2008	SSC, SCC, WCC, SVC, AV, CRR, RBR	Portability
54	Marcus & Poshvanyk, 2005	C3, LCSM	
55	Tumu, dkk., 2013	Modified Expected Cost of Misclassification (MECM)	Reliability

56	Ware, dkk., 2008	DIT, NOC, CBO, RFC, CCB, CCF, WCMCM, WMCLOC	Reliability, Maintainability
57	Shrivastava & Jain, 2011	WMC, DIT, NOC, DIP, RFC, LCOM	Portability, Usability, Maintainability
58	Nagappan & Ball, 2007	LOC	Reliability
59	Soliman, dkk., 2010	WMC, DIT, NOC, CBO, RFC, LCOM	Functionality
60	Kulkarni, dkk., 2010	WMC, DIT, NOC, CBO, RFC, LCOM	Portability, Usability, Maintainability

4.4. RQ-4: Bagaimana validasi metrik pengukuran kualitas perangkat lunak?

Validasi pada metrik kualitas perangkat lunak dapat dilakukan dengan dua cara yaitu (Westfall, 2008): (1) *Internal validity* apakah metrik dapat mengukur sebuah atribut. (2) *Eksternal validity* merupakan kemampuan untuk menggeneralisir hasil metrik pada beberapa kondisi yang lain. Teknik ekstraksi data dari RQ-4 diambil dengan membaca pada bagian pengujian dari sebuah makalah rujukan kemudian dibandingkan dengan definisi dari *internal* dan *eksternal validity*. Pada Tabel 2 dipetakan penelitian-penelitian yang menggunakan validasi untuk metrik perangkat lunak berdasarkan jenis validasinya.

Tabel 2. Jenis Validasi Metrik

Validasi	Makalah
<i>Internal Validity</i>	(Pai & Dugan, 2007) (Selvarani, dkk., 2009) (Yahaya, dkk., 2010) (Ma, dkk., 2009) (Nagappan & Ball, 2007) (Marcus & Poshvanyk, 2005) (Steidl, dkk., 2013) (Costal & Franch, 2012) (Shim, dkk., 2008) (Oyetoyan, dkk., 2013) (Zhou & Liu, 2007) (Hansen, dkk., 2011) (Al Dallal & Briand, 2010) (Thwin & Quah, 2005) (Al Dallal, 2012) (Al Dallal, 2011) (Stefani & Xenos, 2009) (Ahn, dkk., 2013) (Hulse & Khoshgoftaar, 2008)
<i>Eksternal Validity</i>	(Kulkarni, dkk., 2010) (Srivisut & Muenchaisri, 2007) (Chaparro, dkk., 2014) (Khaer, dkk., 2008) (Chirila & Cretu, 2011) (Damm & Lundberg, 2007) (Pai & Dugan, 2007)

4. Kesimpulan

Pada makalah ini dilakukan pemetaan secara sistematis terhadap metrik kualitas perangkat lunak. Menurut pembahasan didapatkan hasil bahwa penelitian tentang metrik kualitas perangkat lunak cukup stabil dilakukan pada sembilan tahun terakhir. Penggunaan metrik pada penelitian tersebut sangat beragam, akan tetapi pada beberapa penelitian merujuk pada metrik perangkat lunak berbasis *object oriented*. Dari metrik *object oriented* ini metrik LOC yang paling banyak digunakan. Hal tersebut mengindikasikan perangkat lunak berbasis *object oriented* masih banyak digunakan. Faktor kualitas yang diteliti didominasi oleh faktor *reability*, kemudian *maintanability* dan *portability* pada ISO 9126. Pengujian validitas sebuah metrik didominasi oleh *internal validity* dimana metrik diuji oleh seorang ahli, diuji dengan teknik *data mining*, ataupun diuji dengan membandingkan hasil pengujian metrik dengan metrik yang lain. Pada penelitian ini diharapkan membantu peneliti yang lain untuk mempelajari dari penelitian-penelitian metrik pengukuran perangkat lunak yang telah dilakukan sebelumnya dengan mudah dan efektif. Kedepannya diharapkan dilakukan penelitian tentang metrik pada setiap proses pembangunan perangkat lunak.

Referensi

- Aggarwal, K. K., Singh, Y., Kaur, A. & Malhotra, R. 2006. Empirical Study of Object-Oriented Metrics. *Journal of Object Technology*, 5(8), pp. 149-173.
- Ahn, S., Huff, S. M., Kim, Y. & Kalra, D. 2013. Quality Metrics for Detailed Clinical Models. *Information and Software Technology*, 82(5), pp. 408-417.
- Akingbehin, K. 2005. *A Quantitative Supplement to The Definition of Software Quality*. s.l., Software Engineering Research, Management and Applications Third ACIS International Conference ACIS International Conference.
- Al Dallal, J. 2011. Improving The Applicability of Object-Oriented Class Cohesion Metrics. *Information and Software Technology*, 53(9), pp. 914-928..
- Al Dallal, J. 2012. Fault Prediction And The Discriminative Powers of Connectivity-Based Object-Oriented Class Cohesion Metrics. *Information and Software Technology*, 54(4), pp. 396-416.
- Al Dallal, J. & Briand, L. C. 2010. An Object-Oriented High-Level Design-Based Class Cohesion Metric. *Information and Software Technology*, 52(12), pp. 1346-1361.
- Alshayeb, M. 2009. Empirical Investigation of Refactoring Effect on Software Quality.

- Information and Software Technology*, 51(9), pp. 1319-1326.
- Ampatzoglou, A., Frantzeskou, G. & Stamelos, I. 2012. A Methodology To Assess The Impact of Design Patterns on Software Quality. *Information and Software Technology*, Volume 54, pp. 331-346.
- Barkmann, H., Lincke, R. & Lowe, W. 2009. *Quantitative Evaluation of Software Quality Metrics in Open-Source Projects*. s.l., Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference.
- Binkley, D., Feild, H., Lawrie, D. & Pighin, M. 2009. Increasing Diversity Natural Language Measures for Software Fault Prediction. *Journal of Systems and Software*, 82(11), pp. 1793-1803.
- Catal, C., Alan, O. & Balkan, K. 2011. Class Noise Detection Based on Software Metrics And ROC Curves. *Information Sciences*, 181(121), pp. 4867-4877.
- Chaparro, O., Bavota, G., Marcus, A. & Penta, M. D. 2014. *On The Impact of Refactoring Operations on Code Quality Metrics*. s.l., Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference.
- Chirila, C. & Cretu, V. 2011. *A Suite of Java Specific Metrics for Software Quality Assessment Based on Static Code Analyzers*. Timișoara, Romania, IEEE International Symposium on Applied Computational Intelligence and Informatics.
- Chirinos, L., Losavio, F. & Bøegh, J. 2005. Characterizing A Data Model for Software Measurement. *Journal of Systems and Software*, 74(2), pp. 207-226.
- Chiu, N. H. 2009. An Early Software-Quality Classification Based on Improved Grey Relational Classifier. *Expert Systems with Applications*, 6(7), pp. 10727-10734.
- Conejero, J. M., Figueiredo, E., Garcia, A., Hernández, J., & Jurado, E. 2012. On The Relationship of Concern Metrics And Requirements Maintainability. *Information and Software Technology*, 54(2), pp. 212-238.
- Costal, D. & Franch, X. 2012. *A Unifying Framework for The Definition of Syntactic Measures Over Conceptual Schema Diagrams*. s.l., Research Challenges in Information Science (RCIS), 2012 Sixth International Conference on. IEEE.
- Damm, L. & Lundberg, L. 2007. *Company-Wide Implementation of Metrics for Early Software Fault Detection*. s.l., IEEE Computer Society.
- Domínguez-Mayo, F. J., Escalona, M. J., Mejías, M., Ross, M., & Staples, G. 2012. Quality Evaluation for Model-Driven Web Engineering Methodologies. *Information and Software Technology*, 54(11), pp. 1265-1282.
- Duque-Ramos, A., Fernández-Breis, J. T., Iniesta, M., Dumontier, M., Aranguren, M. E., Schulz, S., Aussenac-Gilles, N., & Stevens, R. 2013. Evaluation of The OQuaRE Framework for Ontology Quality. *Expert Systems with Applications*, 40(7), pp. 2696-2703.
- Gomathy, G. & Rajalakshmi, S. 2014. *A Software Quality Metric Performance of Professional Management in Service Oriented Architecture*. s.l., IEEE.
- Goyal, P. & Joshi, G. 2014. *QMOOD Metric Sets to Assess Quality of Java Program*. s.l., Issues and Challenges in Intelligent Computing Techniques (ICICT).
- Gui, G. 2006. Component Reusability And Cohesion Measures in Object-Oriented Systems. *Information and Communication Technologies*, Volume 2, pp. 2878-2882.
- Hansen, K. M., Jonasson, K. & Neukirchen, H. 2011. An Empirical Study of Software Architectures' Effect on Product Quality. *Journal of Systems and Software*, 84(7), pp. 1233-1243.
- Hsueh, N. L., Chu, P. H. & Chu, W. 2008. A Quantitative Approach for Evaluating the Quality of Design Patterns. *Journal of Systems and Software*, 81(8), pp. 1430-1439.
- Hulse, J. & Khoshgoftaar, T. M. 2008. A Comprehensive Empirical Evaluation of Missing Value Imputation in Noisy Software Measurement Data. *Journal of Systems and Software*, 81(5), pp. 691-708.
- Ikemoto, S., Dohi, T. & Okamura, H. 2013. *Estimating Software Reliability with Static Project Data in Incremental Development Processes*. s.l., Software Measurement and the 2013

- Eighth International Conference on Software Process and Product Measurement (IWSM-MENSURA).
- Khaer, M. A., Hashem, M. M. A. & Masud, M. R. 2008. *On Use of Design Patterns in Empirical Assessment of Software Design Quality*. s.l., Computer and Communication Engineering, 2008. ICCCE 2008. International Conference.
- Khoshgoftaar, T. M., Seliya, N. & Herzberg, A. 2005. Resource-Oriented Software Quality Classification Models. *Journal of Systems and Software*, 76(2), pp. 111-126.
- Kohichi, U., Igaki, H., Higo, Y. & Kusumoto, S. 2012. *A Study of Student Experience Metrics for Software Development pbl*. s.l., Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD) ACIS International Conference.
- Kulkarni, U. L., Kalshetty, Y. R. & Arde, V. G. 2010. *Validation of CK Metrics for Object Oriented Design Measurement*. s.l., Emerging Trends in Engineering and Technology (ICETET) 3rd International Conference.
- Landman, D., Serebrenik, A. & Vinju, J. 2014. *Empirical Analysis of The Relationship Between CC and SLOC in A Large Corpus of Java Methods*. s.l., Software Maintenance and Evolution (ICSME), 2014 IEEE International Conference.
- Lubke, D., Singer, L. & Salnikow, A. 2009. *Calculating BPEL Test Coverage Through Instrumentation*. s.l., Automation of Software Test, 2009. AST'09. ICSE Workshop.
- Marcus, A. & Poshyvanyk, D. 2005. *The Conceptual Cohesion of Classes*. s.l., ICSM Proceedings of the 21st IEEE International Conference .
- Ma, Y., Ma, X., Liu, S. & Jin, B. 2009. *A Proposal for Stable Semantic Metrics Based on Evolving Ontologies*. s.l., JCAI'09 International Joint Conference.
- Mishra, R. & Sureka, A. 2014. *Mining Peer Code Review System for Computing Effort and Contribution Metrics for Patch Reviewers*. s.l., Mining Unstructured Data (MUD) IEEE 4th Workshop.
- Molina, F. & Toval, A. 2009. Integrating Usability Requirements That Can be Evaluated in Design Time into Model Driven Engineering of Web Information Systems. *Advances in Engineering Software*, 40(12), pp. 1306-1317.
- Monga, C., Jatain, A. & Gaur, D. 2014. *Impact of Quality Attributes on Software Reusability and Metrics to Assess These Attributes*. IEEE International Advance Computing Conference (IACC), pp. 1430-1434.
- Nagappan, N. & Ball, T. 2007. *Using Software Dependencies and Churn Metrics to Predict Field Failures*. s.l., Empirical Software Engineering and Measurement ESEM.
- Nagappan, N., Williams, L., Osborne, J., Vouk, M., & Abrahamsson, P. 2005. *Providing Test Quality Feedback using Static Source Code and Automatic Test Suite Metrics*. s.l., IEEE International Symposium.
- Nikolik, B. 2012. Software Quality Assurance Economics. *Information and Software Technology*, 54(11), pp. 1229-1238.
- Oyetoyan, T. D., Cruzes, D. S. & Conradi, R. 2013. A Study of Cyclic Dependencies on Defect Profile of Software Components. *Journal of Systems and Software*, 86(12), pp. 3162-3182.
- Pai, G. & Dugan, J. 2007. Empirical Analysis of Software Fault Content and Fault Proneness using Bayesian Methods. *IEEE Transactions On Software Engineering*, Volume 33, pp. 675-686.
- Pántos, J., Beszédes, A., Gyenizse, P. & Gyimóthy, T. 2008. Experiences in Adapting a Source Code-Based Quality Assessment Technology. *Software Maintenance and Reengineering*, pp. 311-313.
- Pizzi, N. J. 2013. A Fuzzy Classifier Approach to Estimating Software Quality. *Information Sciences*, Volume 241, pp. 1-11.
- Rotaru, O. P. & Dobre, M. 2005. *Reusability Metrics for Software Components*. s.l., ACS/IEEE International Conference.
- Ryan, C. & Rossi, P. 2005. *Software, Performance and Resource Utilisation Metrics for*

- Context-Aware Mobile Applications*. s.l., n Software Metrics, 2005. 11th IEEE International Symposium.
- Sahraoui, H., Briand, L. C., Guéhéneuc, Y. G. & Beurepaire, O. 2010. Investigating The Impact of A Measurement Program on Software Quality. *Information and Software Technology*, 52(9), pp. 923-933.
- Selvarani, R., Nair, T. & Prasad, V. K. 2009. *Estimation of Defect Proneness using Design Complexity Measurements in Object-Oriented Software*. s.l., International Conference on Signal Processing Systems IEEE.
- Shim, B., Siho, C. & Suntae, K. S. P. 2008. *A Design Quality Model for Service-Oriented Architecture*. s.l., IEEE, pp. 403-410.
- Shrivastava, D. P. & Jain, R. C. 2011. *Unit Test Case Design Metrics in Test Driven Development*. s.l., Communications, Computing and Control Applications (CCCA).
- Sim, G., MacFarlane, S. & Read, J. 2006. All Work and No Play Measuring Fun, Usability, and Learning in Software for Children. *Computers & Education*, 46(3), pp. 235-248.
- Soliman, T. H. A., El-Swesy, A. & Ahmed, S. H. 2010. *Utilizing CK Metrics Suite to UML Models a Case Study of Microarray MIDAS Software*. s.l., Informatics and Systems (INFOS), 2010 The 7th International Conference.
- Srivisut, K. & Muenchaisri, P. 2007. *Bad-Smell Metrics for Aspect-Oriented Software*. s.l., Computer and Information Science IEEE/ACIS International Conference.
- Stefani, A. & Xenos, M. 2009. Meta-Metric Evaluation of E-Commerce-Related Metrics. *Electronic Notes in Theoretical Computer Science*, Volume 233, pp. 59-72.
- Steidl, D., Hummel, B. & Juergens, E. 2013. *Quality Analysis of Source Code Comments*. s.l., Program Comprehension (ICPC), 2013 IEEE 21st International Conference .
- Takai, Y., Kobayashi, T. & Agusa, K. 2011. *Software Metrics Based on Coding Standards Violations*. Software Measurement, 2011 Joint Conference of the 21st Int'l Workshop on and 6th Int'l Conference on Software Process and Product Measurement (IWSM-MENSURA). IEEE, 2011.
- Thwin, M. M. T. & Quah, T. S. 2005. Application of Neural Networks for Software Quality Prediction using Object-Oriented Metrics. *Journal of systems and software*, 76(2), pp. 147-156.
- Turnu, I., Concas, G., Marchesi, M. & Tonelli, R. 2013. The Fractal Dimension of Software Networks as A Global Quality. *Information Sciences*, Volume 245, pp. 290-303.
- Wagner, S., Lochmann, K., Winter, S., Goeb, A., & Klaes, M. 2009. *Quality Models in Practice: A Preliminary Analysis*. s.l., Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium.
- Ware, M. P., Wilkie, F. G. & Shapcott, M. 2008. *The Use of Intra-Release Product Measures in Predicting Release Readiness*. s.l., In Software Testing, Verification, and Validation International Conference.
- Westfall, L. 2008. *The Certified Software Quality Engineer Handbook*. Milwaukee: ASQ Quality Press.
- Yahaya, Haji, J. & Deraman, A. 2010. *Measuring Unmeasurable Attributes of Software Quality using Pragmatic Quality Factor*. s.l., Computer Science and Information Technology (ICCSIT).
- Zhang, T., Deng, L., Wu, J., Zhou, Q., & Ma, C. 2008. *Some Metrics for Accessing Quality of Product Line Architecture*. s.l., Computer Science and Software Engineering, 2008 International Conference.
- Zhou, S. & Liu, X. 2007. *Quality Metrics for Internet Applications: Developing " New" from " Old"*. s.l., In Computer Software and Applications Conference COMPSAC.