

Review Perangkat Lunak StarUML Berdasarkan Faktor Kualitas McCall

Ni Made Satvika Iswari

Fakultas Teknologi Informasi dan Komunikasi, Universitas Multimedia Nusantara, Tangerang,
Indonesia
satvika@umn.ac.id

Diterima 25 Februari 2015

Disetujui 10 Juni 2015

Abstract—Pembangunan perangkat lunak dilakukan berdasarkan hasil analisis kebutuhan fungsional yang diperlukan. Seringkali, perangkat lunak yang dikembangkan hanya cukup memenuhi kebutuhan fungsional dasar saja, padahal hal tersebut belum cukup untuk menjadikan perangkat lunak yang dibangun memiliki kinerja yang baik ketika dioperasikan. Terdapat beberapa faktor kualitas yang perlu diperhatikan yang akan menentukan kinerja perangkat lunak yang dibangun. Pada tulisan ini, akan diuraikan hasil analisis faktor kualitas terhadap kakas pemodelan perangkat lunak yang banyak digunakan, yaitu StarUML.

Index Terms—faktor kualitas, kinerja, kebutuhan non fungsional, kakas pemodelan, StarUML

I. PENDAHULUAN

Saat ini, penilaian terhadap faktor kualitas dari perangkat lunak menjadi bagian yang sangat penting dalam bidang rekayasa perangkat lunak. Penilaian perangkat lunak tersebut sangatlah penting untuk meningkatkan kinerja dari penggunaan perangkat lunak. Sering kali perangkat lunak yang dikembangkan hanya cukup memenuhi kebutuhan fungsional dasar dari perangkat lunak saja. Namun, hal tersebut belum cukup untuk menjadikan perangkat lunak tersebut memiliki kinerja yang baik ketika dioperasikan. Terdapat beberapa aspek kualitas yang perlu diperhatikan untuk mengembangkan perangkat lunak yang memiliki kinerja yang baik[2].

Terdapat beberapa model dari kualitas perangkat lunak dan kategorinya yang telah

diperkenalkan selama ini. Salah satunya adalah model klasik yang diperkenalkan oleh McCall pada tahun 1977. Adapun model lainnya diajukan oleh Deutsch dan Willis (1988) dan oleh Evans dan Marciniak (1987). Model lainnya tersebut secara substansial tidak jauh berbeda dengan Model McCall. Walaupun Model McCall lebih lama, namun dinilai menyediakan metode yang lebih praktis dalam mengklasifikasi kebutuhan perangkat lunak ke dalam faktor kualitas perangkat lunak[1]. Model tersebut terdiri dari 11 buah faktor kualitas. Faktor-faktor tersebut dikelompokkan dalam 3 buah kategori, yaitu sebagai berikut[3]:

- **Product Operation Factors**, yang terdiri dari faktor *correctness*, *reliability*, *efficiency*, *integrity*, dan *usability*.
- **Product Revision Factors**, yang terdiri dari faktor *maintainability*, *flexibility*, dan *testability*.
- **Product Transition Factors**, yang terdiri dari faktor *portability*, *reusability*, dan *interoperability*.

Selain ke-11 buah faktor yang diajukan oleh McCall, terdapat 5 buah faktor lain yang diajukan oleh Evan dan Marciniak (1987), serta Deutsch dan Willis (1988). Kelima buah faktor tersebut adalah *verifibility*, *expandability*, *safety*, *manageability*, dan *survivability*[3].

Pada makalah ini, akan dikaji kualitas dari sebuah perangkat lunak *open source*, yaitu StarUML berdasarkan faktor – faktor kualitas yang telah dijelaskan sebelumnya. Penulis merasa tertarik untuk mereview perangkat lunak pemodelan dikarenakan bidang yang digeluti banyak membutuhkan perangkat lunak untuk membangun model. Sementara StarUML

dibandingkan dengan perangkat lunak pemodelan lainnya memiliki keistimewaan pada tujuan pembangunannya, yaitu menggantikan perangkat lunak pemodelan UML komersil seperti Rational Rose, Together, dll[11]. Berdasarkan tujuan tersebut, penulis berpendapat bahwa StarUML akan terus disempurnakan demi mencapai tujuannya tersebut, sehingga diharapkan StarUML dapat menjadi perangkat lunak pemodelan UML yang paling berkualitas.

II. STAR UML

StarUML merupakan proyek *open source* untuk mengembangkan *platform Unified Modeling Language (UML)* atau *Model Driven Architecture (MDA)* yang cepat, fleksibel, dapat diperluas, memiliki banyak fitur, dan tidak dipungut biaya. Tujuan dari proyek ini dalam untuk membangun sebuah perangkat lunak pemodelan dan sekaligus *platform* yang dapat menggantikan perangkat UML berbayar lain,

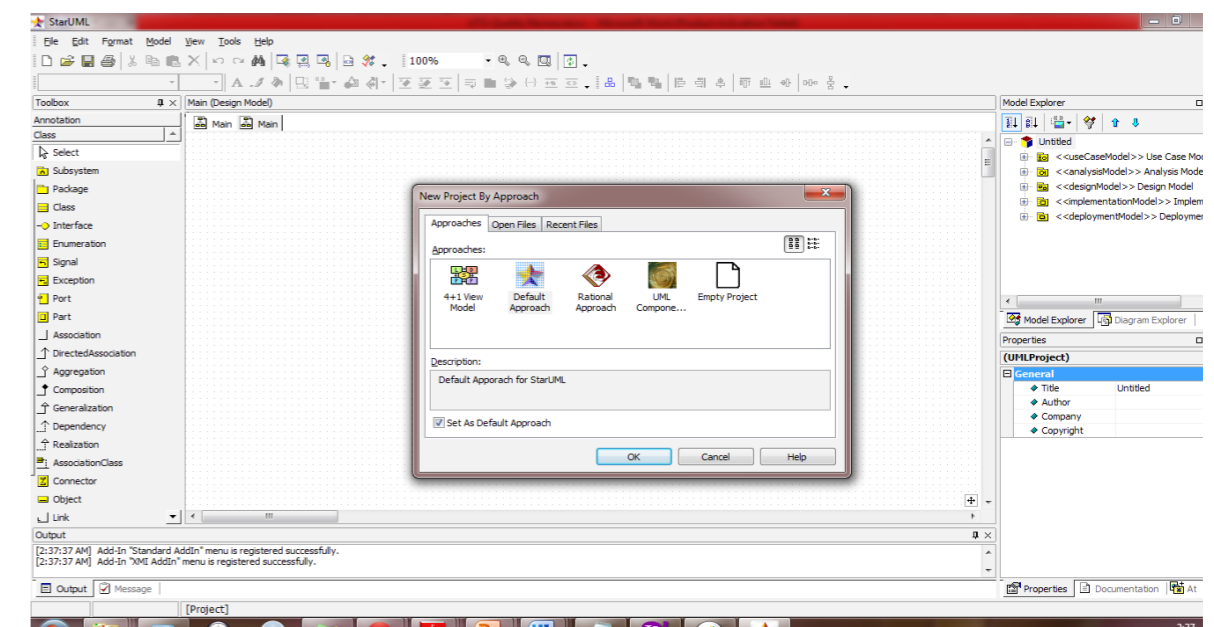
perangkat lunak ini.

seperti Rational Rose, Together, dan sebagainya.

StarUML dikembangkan dalam Bahasa Pemrograman Delphi. Walaupun begitu, StarUML merupakan proyek yang *multi-lingual* dan tidak bergantung pada bahasa pemrograman yang spesifik, sehingga bahasa pemrograman apapun dapat digunakan untuk mengembangkan StarUML, seperti C/C++, Java, Visual Basic, Delphi, Jscript, VBScript, C#, VB.NET, dan sebagainya. Gambar 1 merupakan tampilan utama dari StarUML.

III. REVIEW KUALITAS PERANGKAT LUNAK

Pada bagian ini akan dijelaskan mengenai kajian kualitas perangkat lunak StarUML berdasarkan faktor-faktor kualitas yang telah dijelaskan sebelumnya. Kajian ini dilakukan berdasarkan dokumentasi perangkat lunak yang tersedia dan hasil studi literatur, serta pendapat-pendapat orang lain yang telah menggunakan

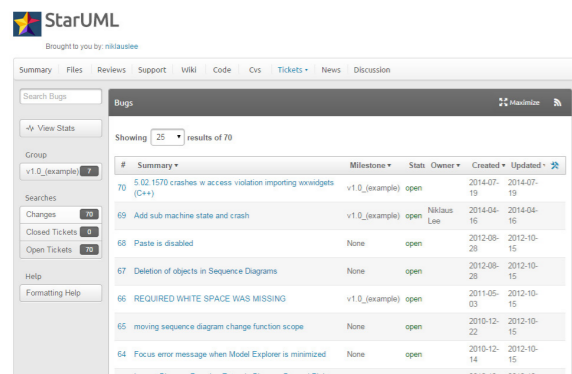


Gambar 1. Tampilan Antarmuka StarUML

A. Correctness

Correctness merupakan kemampuan produk perangkat lunak untuk menjalankan fungsi – fungsinya sesuai dengan spesifikasi yang telah didefinisikan sebelumnya. Faktor *correctness* dapat dikatakan sebagai faktor utama dalam menentukan kualitas dari sebuah perangkat lunak. Apabila sebuah perangkat lunak tidak dapat menjalankan fungsi-fungsi yang telah didefinisikan untuknya, atau dalam arti tidak dapat memenuhi faktor *correctness*, maka perangkat lunak tersebut dapat dianggap gagal dan tidak dapat digunakan oleh pengguna[9].

Untuk perangkat lunak StarUML, terdapat *bugs* yang dilaporkan oleh penggunanya, yang berjumlah 70 buah, seperti yang dapat kita lihat pada Gambar 2.



Gambar 2. Beberapa Bugs yang Dilaporkan oleh Pengguna StarUML [7]

Berdasarkan data tersebut, dapat kita lihat bahwa perangkat lunak StarUML masih memiliki titik-titik yang memungkinkan terjadinya kesalahan. Sehingga secara kualitatif, dapat dikatakan bahwa StarUML belum memenuhi faktor kualitas *correctness* sepenuhnya.

Namun, secara umum, StarUML dapat bekerja dengan baik di beberapa lingkungan sistem operasi, seperti yang dibahas pada website appdb.winehq.org[8]. Pada website tersebut dijelaskan bahwa StarUML telah diuji di beberapa sistem operasi, seperti Ubuntu, openSUSE, Zenwalk, dan Gentoo. Adapun pengujian yang dilakukan

meliputi instalasi, *running*, membuat file baru, membuka file yang dibuat di Windows, dan sebagainya. Secara umum, hasil pengujian dinilai baik, kecuali untuk lingkungan Debian GNU/Linux 4.0 “Etch”, yang menyatakan bahwa StarUML tidak dapat beroperasi pada lingkungan tersebut. Adapun hasil pengujian tersebut dapat dilihat pada Gambar 3.

Test Results							
	Distribution	Test date	Wine version	Installs?	Runs?	Rating	Submitter
Current	Linux Mint 15 "Olivia"	Jul 04 2013	1.6-rc4	Yes	Yes	Silver	an anonymous user
Show	Ubuntu 10.04 "Lucid" (386 (+ variants like Kubuntu))	Nov 24 2010	1.3.6	Yes	Yes	Gold	Michael Franz
Show	openSUSE 11.0	Feb 02 2009	1.1.13	Yes	Yes	Bronze	an anonymous user
Show	Ubuntu 9.04 "Jaunty" (386 (+ variants like Kubuntu))	Sep 11 2009	1.0.1	Yes	Yes	Gold	an anonymous user
Show	openSUSE 10.2 x86_64	Sep 06 2008	1.1.3	Yes	Yes	Gold	an anonymous user
Show	Zenwalk 5	Aug 28 2008	1.1.3	Yes	Yes	Platinum	an anonymous user
Show	Gentoo Linux AMD64	Jun 09 2008	1.0-rc4	Yes	Yes	Bronze	an anonymous user
Show	Gentoo Linux AMD64	May 17 2008	1.0-rc1	Yes	Yes	Gold	an anonymous user
Show	Ubuntu 7.10 "Gutsy" (386 (+ variants like Kubuntu))	Apr 21 2008	0.9.59	Yes	Yes	Silver	an anonymous user
Show	Ubuntu 7.10 "Gutsy" (386 (+ variants like Kubuntu))	Feb 18 2008	0.9.54	Yes	Yes	Gold	an anonymous user

Gambar 3 Hasil Pengujian StarUML oleh appdb.winehq.org[8]

B. Reliability

Reliability merupakan ukuran ketahanan sebuah perangkat lunak dari resiko dan kemungkinan potensi kegagalan. Faktor ini dapat ditentukan dengan menghitung nilai probabilitas dari operasi perangkat lunak yang bebas kesalahan dalam waktu tertentu dan lingkungan tertentu. Faktor *reliability* erat kaitannya dengan faktor *Correctness* dan *Robustness*. Ketika sebuah perangkat lunak mampu menjalankan fungsionalitasnya dengan tepat sesuai dengan kebutuhan yang telah didefinisikan sebelumnya (*correctness*) dan mampu mengatasi kesalahan yang terjadi pada perangkat lunak (*robustness*), maka sistem tersebut dikatakan andal, dan memenuhi faktor *Reliability*[9].

Apabila dilihat dari faktor *correctness*, StarUML dapat dikatakan belum memenuhi faktor kualitas tersebut, karena masih terdapat *bugs* yang dilaporkan oleh pengguna perangkat lunak tersebut, seperti yang ditunjukkan pada Gambar 2. Sementara itu, faktor *robustness* saling

berkomplemen (melengkapi) dengan faktor *correctness*[9]. Dengan demikian, karena terdapat *bugs* yang dilaporkan oleh pengguna, maka hal tersebut menggambarkan bahwa perangkat lunak belum dapat mengatasi kesalahan yang terjadi. Sehingga, Berdasarkan fakta tersebut, perangkat Lunak StarUML juga belum memenuhi faktor *robustness*. Dengan demikian, mengacu pada syarat bahwa sebuah perangkat lunak dikatakan memenuhi faktor *reliability*[9], dapat disimpulkan bahwa Perangkat Lunak StarUML belum dapat memenuhi faktor kualitas *Reliability*.

C. Efficiency

Faktor kualitas efisiensi (*efficiency*) berhubungan dengan kemampuan perangkat lunak untuk menggunakan sesedikit mungkin sumber daya perangkat keras (*hardware*), seperti waktu prosesor, *space* yang dibutuhkan pada memory internal dan eksternal, *bandwidth* yang digunakan pada perangkat komunikasi. Dengan demikian, selain berhubungan langsung dengan penggunaan perangkat keras, faktor kualitas ini juga berhubungan erat dengan kualitas algoritma kode yang digunakan dalam pengembangan perangkat lunak[9].

Perangkat Lunak StarUML dikembangkan dengan menggunakan Bahasa Pemrograman Delphi. Apabila dilihat dari sisi bahasa pemrograman, Delphi tidak menggunakan *garbage collector*, seperti yang digunakan pada Bahasa Pemrograman Java atau C#. Penggunaan *garbage collector* sendiri memungkinkan adanya *delay* dan memakan memori yang lebih dari sumber daya. Dengan menggunakan Bahasa Pemrograman Delphi, StarUML memiliki keunggulan dalam faktor efisiensi sumber daya.

Tabel 1. Perbandingan Kebutuhan Perangkat Keras Antara Tools UMN[5][6]

Kebutuhan Perangkat Keras	Rational Rose v7.0	StarUML	JUDE/Professional and Community 5.4.x or earlier
Prosesor	600 MHz	233 MHz	700 MHz
RAM	512 MB	128 MB	256 MB
Hard Disk Space	173 MB	110 MB	128 MB
Resolusi Monitor	1024x768 pixels	1024x768 pixels	-

Tabel 1 menggambarkan perbandingan kebutuhan perangkat keras (*hardware*) yang diperlukan untuk mengoperasikan masing-masing *tools* UML. Terdapat 3 buah *tools* yang dibandingkan, diantaranya adalah Rational Rose yang dibuat oleh IBM, StarUML, dan JUDE yang dibuat oleh ChangeVision. Dari tabel tersebut, kita dapat melihat bahwa StarUML menggunakan spesifikasi kebutuhan perangkat keras yang paling rendah dibandingkan *tools* lainnya. Hal ini menyatakan bahwa StarUML mampu dioperasikan dengan baik walaupun dengan spesifikasi perangkat keras yang relatif rendah. Dengan demikian, StarUML dinilai memiliki keunggulan dalam faktor efisiensi dibandingkan dengan *tools* serupa, yaitu Rational Rose dan JUDE.

D. Integrity

Faktor integritas berfokus pada keamanan dari perangkat lunak. Faktor tersebut menyatakan bahwa segala akses ke perangkat lunak hanya diizinkan untuk pihak yang berwenang saja[9]. Hal ini perlu dilakukan untuk menghindari adanya pihak yang tidak terotorisasi mengakses perangkat lunak dan melakukan perbuatan yang tidak diinginkan.

Perangkat lunak StarUML merupakan proyek yang bersifat terbuka (*open source*), yang berarti bahwa segala pihak dapat ikut serta dalam pengembangan perangkat lunak ini. Namun, walaupun begitu, pihak StarUML tidak menyerahkan seluruh aspek pengembangan kepada pihak lain yang ingin bergabung. Pihak lain yang ingin ikut mengembangkan StarUML dapat melakukannya dengan cara mengirimkan hasil pengembangan melalui email kepada pihak StarUML. Setelah itu, apabila pengembangan disetujui, maka hasil pengembangan yang dilakukan pihak lain tersebut akan digabungkan dengan proyek utama. Pihak lain yang melakukan kontribusi tersebut kemudian akan mendapatkan penghargaan, yaitu sistem akan menampilkan namanya sebagai nama *default* untuk model yang dibuat oleh pengguna.

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa untuk mengakses kode utama dari StarUML tidaklah mungkin. Pengembang pihak ketiga hanya dapat menyerahkan hasil

pekerjaannya kepada pihak StarUML. Dengan demikian, dapat dikatakan bahwa StarUML memiliki tingkat integritas yang tinggi terhadap akses kode sumbernya.

E. Maintainability

Faktor kualitas Maintainability menyatakan banyaknya biaya yang berkaitan dengan perangkat lunak pada saat perangkat lunak tersebut mulai dioperasikan oleh pengguna. Terdapat tiga buah kategori untuk proses pemeliharaan perangkat lunak, yaitu *corrective maintenance* yang berhubungan dengan perbaikan kesalahan yang muncul, *adaptive maintenance* yang berhubungan dengan perubahan perangkat lunak untuk menyesuaikan dengan kebutuhan pengguna, dan *perfective maintenance* yang berhubungan dengan peningkatan kinerja dari perangkat lunak[9].

Walaupun sempat terhenti pada tahun 2005, saat ini StarUML kembali dikembangkan dengan versi terbarunya, yaitu StarUML 2. Adapun pengembangan yang dilakukan lebih banyak pada *adaptive maintenance*, yaitu untuk menyesuaikan dengan kebutuhan pengguna. Pengembangan yang dilakukan diantaranya adalah fitur untuk meng-ekspor seluruh diagram menjadi file gambar, pembuatan Entity-Relationship Diagram (ERD), XMI Extension, PHP Extension, dan tambahan fitur lainnya untuk kemudahan dalam mengedit diagram[10].

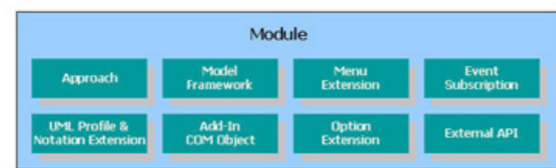
Dengan adanya aktivitas *maintenance* yang masih aktif dilakukan sampai saat ini, dapat dikatakan StarUML memenuhi faktor kualitas *maintainability*.

StarUML hingga saat ini masih dinilai oleh sebagian besar orang sebagai perangkat lunak pemodelan yang paling baik dibandingkan perangkat lunak sejenisnya. Namun, sejak tahun 2005, tidak ada aktifitas pengembangan yang dilakukan oleh pihak StarUML. Hal ini disesali oleh banyak pihak, namun belum ada penjelasan langsung dari pihak StarUML mengenai hal ini. Dengan berakhirnya aktifitas pengembangan StarUML, maka dapat disimpulkan bahwa tidak ada lagi pemeliharaan yang dilakukan terhadap StarUML. Dengan demikian, Perangkat Lunak StarUML tidak memenuhi faktor kualitas

Maintainability.

F. Flexibility

Flexibility merupakan kemampuan dan usaha yang dibutuhkan untuk mendukung *adaptive maintenance*, yang berhubungan dengan perubahan perangkat lunak untuk memenuhi kebutuhan pengguna. Faktor kualitas ini juga mendukung *perfective maintenance*, yang berhubungan dengan perubahan algoritma untuk meningkatkan kinerja dari perangkat lunak. Faktor ini berpengaruh secara signifikan dalam melakukan adaptasi terhadap perubahan-perubahan yang mungkin terjadi pada kebutuhan pengguna[9].



Gambar 4. Modul-Modul yang Tersedia pada StarUML

Gambar 4 menunjukkan modul-modul yang tersedia pada StarUML. Menurut dokumentasi yang tersedia, modul-modul tersebut memungkinkan pengembang untuk menambahkan fungsionalitas atau fitur baru ke perangkat lunak yang dikembangkan. Hal ini mendukung salah satu sub-faktor dari Flexibility, yaitu Modularity. Dengan demikian, StarUML memiliki modularitas yang cukup baik. Modul-modul ini terdokumentasi dengan baik sehingga dapat menghemat biaya yang dibutuhkan untuk perubahan perangkat lunak apabila dibutuhkan. Selain itu, modul-modul tersebut tidak saling bergantung antara satu modul dengan modul lainnya. Masing-masing modul menjalankan fungsi yang spesifik dan tidak saling bergantung. Dengan demikian, StarUML memiliki desain modul yang sederhana dan mendukung sub-faktor dari Flexibility, yaitu Simplicity.

StarUML tidak hanya spesifik untuk membuat model UML tertentu saja, melainkan dapat digunakan untuk membuat berbagai macam model UML seperti *use case diagram*, *sequence diagram*, *class diagram*, dan sebagainya. Selain

itu, walaupun StarUML didesain untuk dijalankan di Sistem Operasi Windows, aplikasi ini juga dapat dijalankan di berbagai Sistem Operasi lainnya, seperti hasil pengujian yang ditunjukkan pada Gambar 3. Dengan demikian, StarUML memenuhi salah satu sub-faktor dari Flexibility, yaitu Generality.

G. Testability

Faktor kualitas Testability menekankan pada pengujian perangkat lunak yang sedang/akan dibangun. Adapun tujuan dari pengujian tersebut adalah untuk memastikan bahwa fungsionalitas dari perangkat lunak telah sesuai dengan spesifikasi kebutuhan perangkat lunak yang telah didefinisikan sebelumnya di awal[9].

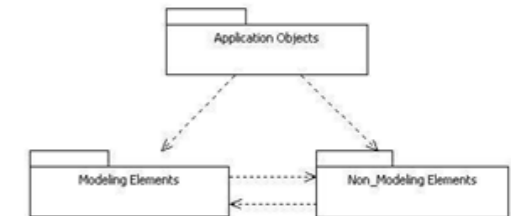
Terdapat beberapa faktor Testability yang sebaiknya diimplementasikan pada proyek pembangunan perangkat lunak, diantaranya adalah *item testing*, *integration testing*, *system testing*, dan *acceptance testing*. Namun, pengujian-pengujian ini tidak dapat dilakukan pada Perangkat Lunak StarUML, dikarenakan tidak adanya akses untuk mendapatkan kode program StarUML secara utuh dan tidak adanya dokumentasi terkait pembangunan perangkat lunak yang dapat diakses secara bebas.

H. Reusability

Reusability merupakan penggunaan ulang bagian perangkat lunak, dapat berupa modul, dari perangkat lunak yang telah dikembangkan sebelumnya. Penggunaan ulang modul ini memberikan beberapa keuntungan, salah satunya adalah meminimalisasi adanya kesalahan yang mungkin terjadi. Perangkat lunak yang telah dikembangkan lebih dulu tentunya telah melalui tahapan-tahapan pengujian untuk mengurangi kesalahan yang muncul. Dengan menggunakan modul dari perangkat lunak yang telah teruji tersebut, pengembangan selanjutnya dapat menghindari kesalahan yang sama dan mempercepat proses pengembangan[4].

StarUML menyediakan *Application Programming Interface* (API) yang cukup luas. API eksternal dari StarUML merupakan *interface* standard yang memungkinkan penggunaan program internal dari luar. Gambar 5 menunjukkan 3 buah bagian utama dari API eksternal StarUML. API ini selain memudahkan

pengembang untuk memodifikasi perangkat lunak, juga memungkinkan pengembang lain dapat menggunakan bagian-bagian yang diperlukan saja untuk mengembangkan perangkat lunak lainnya. Dengan demikian StarUML dapat memenuhi faktor kualitas Reusability.



Gambar 5. Tiga Bagian Utama dari API Eksternal StarUML

I. Interoperability

Interoperability merupakan salah satu faktor kualitas yang berfokus pada hubungan antara perangkat lunak dengan perangkat lainnya. Dengan adanya faktor kualitas ini, diharapkan dua atau lebih perangkat dapat saling berbagi dan memroses informasi yang independen terhadap bahasa dan platform yang digunakan. Tujuan dari interoperability adalah mencapai integrasi yang *smooth* antara *legacy* dengan aplikasi yang baru bahkan dengan perangkat keras yang beragam dan platform perangkat lunak yang berbeda. Dengan demikian, hal ini bukan hanya menguntungkan pengguna saja, namun juga dapat meningkatkan kinerja dari perangkat lunak tersebut[9].

Seperti yang telah dijelaskan sebelumnya, bahwa arsitektur Perangkat Lunak StarUML terdiri dari beberapa buah modul. Modul-modul ini merupakan paket perangkat lunak yang memungkinkan adanya penambahan dari fungsionalitas dan fitur yang baru. Modul tersebut terdiri dari berbagai macam elemen, yang dapat dikembangkan untuk tujuan yang berbeda, salah satunya untuk integrasi dengan *tools* yang spesifik, seperti Visual SourceSafe, CVS, Ms Word, Eclipse, Visual Studio.NET, dan sebagainya. Dengan adanya modul tersebut, Perangkat Lunak StarUML dapat dikatakan memenuhi faktor kualitas Interoperability, karena memungkinkan adanya integrasi dengan berbagai *tools* spesifik.

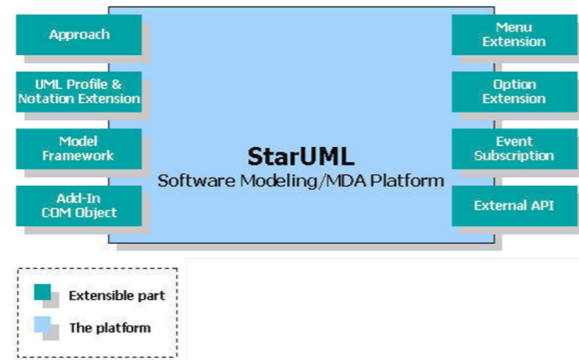
J. Verifiability

Faktor kualitas Verifiability merupakan pengukuran seberapa memungkinkan untuk melakukan pemeriksaan terhadap desain dan pemrograman yang dilakukan untuk mengembangkan perangkat lunak. Faktor ini dinilai cukup penting untuk menentukan kualitas dari suatu perangkat lunak, karena faktor kualitas ini dapat menentukan apakah pengujian dapat dilakukan dengan mudah dan efisien. Dengan demikian, kita dapat mengetahui apakah perangkat lunak telah berjalan sesuai dengan spesifikasi kebutuhan yang telah ditetapkan sebelumnya[9].

Untuk menentukan faktor kualitas verifiability, biasanya mengacu pada subfaktor modularity, simplicity, dan kesesuaian pada aturan dokumentasi dan panduan[9]. Untuk StarUML, telah dijelaskan sebelumnya bahwa perangkat lunak memenuhi subfaktor Modularity, seperti yang telah ditunjukkan pada Gambar 4, bahwa perangkat lunak terdiri dari modul-modul yang saling independen. Selain itu, hal ini juga menunjukkan bahwa StarUML juga memenuhi subfaktor Simplicity. Namun, untuk kesesuaian pada dokumentasi dan panduan tidak dapat dilakukan pengujian karena kode sumber (*source code*) dan dokumentasi terkait pembangunan perangkat lunak untuk StarUML tidak didapatkan secara utuh, sehingga hasil review untuk faktor kualitas verifiability ini tidak dapat ditentukan.

K. Expandability

Faktor kualitas Ekspandability menekankan pada seberapa banyak usaha yang mungkin dibutuhkan untuk mengembangkan suatu perangkat lunak setelah perangkat lunak tersebut mulai dioperasikan[9]. Terkadang, ketika sebuah perangkat lunak mulai dioperasikan oleh pengguna, terdapat perubahan-perubahan yang diperlukan untuk memenuhi kebutuhan pengguna. Perubahan tersebut dapat berupa penambahan fungsional atau meningkatkan layanan dari perangkat lunak.



Gambar 6. Arsitektur StarUML

StarUML merupakan sebuah platform untuk memodelkan perangkat lunak yang dapat dikembangkan. Hal ini berarti StarUML tidak hanya menyediakan fungsi-fungsi yang telah didefinisikan sebelumnya saja, namun memungkinkan untuk menambah fungsi-fungsi baru. Gambar 6 merupakan arsitektur dari StarUML. Bagian kotak yang berwarna biru muda merupakan platform dan bagian kotak yang berwarna hijau tua merupakan bagian yang dapat dikembangkan. Bagian tersebut dapat dikembangkan oleh pengguna atau pihak lainnya, lalu ditambahkan ke platform untuk integrasi. Berdasarkan arsitektur ini, kita dapat melihat bahwa Perangkat Lunak StarUML memenuhi faktor kualitas Expandability, yaitu memungkinkan adanya pengembangan lebih jauh dari fungsional yang sudah ada saat ini.

L. Manageability

Faktor kualitas manageability menekankan pada perubahan yang mungkin terjadi pada perangkat lunak, baik pada tahap pengembangan maupun pada tahap pemeliharaan. Perubahan yang terjadi dapat berupa manajemen konfigurasi, prosedur perubahan perangkat lunak, dan sebagainya[9].

Pada perangkat lunak StarUML, tidak ditemukan prosedur-prosedur yang dapat dilakukan untuk melakukan perubahan pada perangkat lunak, baik pada tahap pengembangan maupun pada tahap pemeliharaan perangkat lunak. Pada dokumentasi perangkat lunak, hanya dijelaskan modul-modul yang dapat dikembangkan dari fungsional yang sudah ada

sebelumnya, bukan untuk mengubah fungsional yang sudah ada. Dengan demikian, dapat dikatakan bahwa StarUML tidak memenuhi faktor kualitas Manageability.

M. Safety

Faktor kualitas Safety menekankan pada faktor keselamatan pengguna dari suatu perangkat lunak. Perangkat lunak yang memenuhi faktor Safety diharapkan dapat mengeliminasi kondisi berbahaya untuk pengguna perangkat lunak sebagai hasil dari kesalahan proses pada kontrol perangkat lunak[9].

Pada Perangkat Lunak StarUML, faktor Safety tidak terlalu dipertimbangkan. Hal tersebut dikarenakan tidak ada hal-hal yang mungkin dapat membahayakan pengguna ketika kesalahan proses terjadi pada perangkat lunak. Dengan demikian, faktor Safety tidak dapat dipenuhi oleh Perangkat Lunak StarUML.

N. Usability

Usability berhubungan dengan usaha yang dibutuhkan untuk mempelajari, mengoperasikan, menyiapkan *input* dan menafsirkan *output* dari sebuah program. Faktor kualitas ini sangat dipengaruhi oleh kemampuan pengguna untuk menjalankan perangkat lunak sesuai dengan kebutuhannya[9].

StarUML selain menyediakan dokumentasi untuk pengembangan bagi *developer*, juga menyediakan dokumentasi yang ditujukan untuk menjalankan perangkat lunak. Dokumentasi ini sangatlah penting, terutama bagi pengguna yang pertama kali menggunakan perangkat lunak ini untuk memandu cara menggunakan StarUML. Dokumentasi ini dinilai cukup baik dalam membantu pengguna menggunakan perangkat lunak ini. Hal ini dapat dilihat dari isi dokumentasi yang memberikan penjelasan kepada pengguna mulai dari konsep dasar, manajemen proyek, hingga *print* diagram.

Dengan demikian, dengan adanya dokumentasi yang baik ini mengenai Perangkat Lunak StarUML, membuat perangkat lunak tersebut menjadi mudah dipahami (*understandability*), mudah dipelajari (*learnability*), mudah dioperasikan (*operability*), dan atraktif (*attractiveness*). Karena memenuhi sub-subfaktor

dari *usability*, maka Perangkat Lunak StarUML dinilai memenuhi faktor kualitas Usability.

IV. DISKUSI KETERKAITAN FAKTOR KUALITAS PERANGKAT LUNAK

Faktor-faktor kualitas sangatlah penting untuk selalu diperhatikan dalam pengembangan suatu perangkat lunak. Faktor-faktor kualitas tersebutlah yang akan menentukan apakah suatu perangkat lunak dapat diterima dan dioperasikan oleh pengguna atau tidak. Apabila sebuah perangkat lunak dikembangkan dengan selalu memperhatikan faktor-faktor kualitas, maka perangkat lunak niscaya akan berjalan sesuai dengan spesifikasi yang telah didefinisikan sebelumnya. Namun, apabila sebaliknya, perangkat lunak yang hanya fokus pada kebutuhan dasar saja tanpa memperhatikan faktor kualitas, biasanya hanya akan menghabiskan biaya yang berlipat ganda dalam proses pemeliharaan.

Berikut ini adalah sebuah diskusi mengenai keterkaitan antara suatu faktor kualitas dengan faktor kualitas lainnya secara umum yang didapatkan setelah melakukan review perangkat lunak.

A. Correctness, Robustness, dan Reliability

Hingga saat ini, membangun sebuah perangkat lunak yang dapat berjalan sesuai dengan spesifikasi tanpa adanya kesalahan (*bugs*) sama sekali adalah sangat sulit untuk dilakukan. Namun, meningkatkan faktor *correctness* maupun *robustness* dapat dilakukan sejak proses pengembangan perangkat lunak. Adapun caranya adalah dengan melakukan pendekatan yang sistematis pada pembangunan perangkat lunak, spesifikasi kebutuhan didefinisikan dengan lebih jelas dan formal, proses pengecekan yang dapat terus menerus dilakukan selama pengembangan, pendefinisian kebutuhan akan *correctness* dan *robustness* dapat didefinisikan dengan jelas di awal, dan lebih disiplin ketika *exception* dengan memperhatikan seluruh aspek yang berkaitan dengan *exception* tersebut.

Faktor *correctness* dinilai sangat dekat kaitannya dengan *robustness*, bahkan saling melengkapi. Sistem yang memenuhi faktor *correctness* akan mampu berjalan dengan baik

sesuai dengan spesifikasinya tanpa adanya *bug*. Namun, apabila *bug* tidak mungkin dihindari, sistem harus mampu menanganinya agar perangkat lunak dapat terus berjalan dan *bug* yang ada tidak sampai menyebabkan *failure*. Sistem tersebut dapat dikatakan sistem yang *robust*. Dengan demikian, *correctness* dan *robustness* memiliki kaitan yang sangat erat, dan apabila memenuhi keduanya, sistem dikatakan *reliable*.

B. Expandability, Reusability, dan Modularity

Tidak seperti perangkat keras yang mungkin akan rusak seiring dengan berjalannya waktu, perangkat lunak cenderung abadi dan tidak akan pernah rusak. Oleh karena itu, selama penggunaan sebuah perangkat lunak, cenderung akan ada perubahan-perubahan yang dibutuhkan untuk memenuhi kebutuhan pengguna. Dengan demikian, perangkat lunak yang dikembangkan haruslah dapat dengan mudah dimodifikasi untuk penyesuaian dengan kebutuhan pengguna. Salah satu caranya adalah dengan mendesain perangkat lunak berdasarkan modul-modul yang menjalankan fungsional spesifik tertentu. Desain yang modular seperti ini akan memudahkan pengembang selanjutnya untuk melakukan perubahan-perubahan pada perangkat lunak apabila diperlukan. Pengembang cukup mempelajari modul terkait dengan perubahan, tanpa mendalami modul lainnya. Dengan demikian, waktu yang dibutuhkan untuk adanya pengembangan lebih lanjut (*expand*) dapat diminimalisasi. Selain itu, apabila terdapat pengembangan perangkat lunak lain yang memiliki beberapa fungsionalitas yang sama dapat menggunakan modul yang sudah dibuat sebelumnya. Hal ini akan mempersingkat waktu pengembangan dan menghasilkan sistem yang lebih andal karena modul yang digunakan kembali (*reuse*) tentunya sudah diuji terlebih dahulu sehingga akan meminimalisasi kesalahan yang terjadi.

C. Safety

Tidak semua perangkat lunak membutuhkan faktor keamanan bagi penggunaannya. Perangkat lunak yang tidak berkaitan dengan perangkat keras lain saat beroperasi cenderung tidak membutuhkan faktor keamanan. Hal ini dikarenakan bahaya yang disebabkan oleh adanya kesalahan pada perangkat lunak tidak

akan berdampak langsung pada pengguna yang mengoperasikannya. Lain halnya dengan perangkat lunak yang berkaitan dengan perangkat keras lain, seperti alat-alat medis. Perangkat tersebut mutlak harus memperhatikan faktor keamanan karena berhubungan langsung dengan nyawa pasien.

D. Dokumentasi

Dokumentasi dinilai sangat penting baik dalam penggunaan, maupun proses pemeliharaan sebuah perangkat lunak. Perangkat lunak yang bagus, sebaiknya memiliki dua jenis dokumentasi, yaitu dokumentasi yang ditujukan untuk pengguna perangkat lunak, dan dokumentasi yang ditujukan untuk pengembang lainnya.

Dokumentasi yang ditujukan untuk pengguna sangat berperan dalam meningkatkan faktor *usability* dari perangkat lunak. Pengguna yang baru pertama kali menggunakan perangkat lunak tentunya akan sangat terbantu dengan adanya dokumentasi tersebut. Namun, perlu juga diperhatikan bahwa dokumentasi hendaknya dibuat dengan sesederhana mungkin namun cukup jelas. Hal ini, akan mempersingkat waktu pengguna untuk mempelajari dokumentasi tersebut.

Dokumentasi yang ditujukan untuk pengembang lainnya sangat berperan dalam proses pemeliharaan perangkat lunak. Perubahan perangkat lunak ketika mulai dioperasikan mungkin saja akan terjadi, dan pengembang yang melakukan perubahan belum tentu sama dengan pengembang yang mengembangkan perangkat lunak tersebut untuk pertama kalinya. Dengan demikian, dokumentasi perangkat lunak untuk pengembang sangat diperlukan untuk meningkatkan faktor *maintainability*, *expandability*, *reusability*, dan faktor-faktor kualitas dalam hal revisi lainnya. Selain itu, dokumentasi yang dibuat dengan sistematis, sederhana, namun jelas dapat mempersingkat waktu untuk mempelajarinya dan sekaligus mempersingkat waktu pemeliharaan perangkat lunak.

V. KESIMPULAN

Berdasarkan hasil review yang dilakukan

terhadap perangkat lunak StarUML berdasarkan faktor kualitas yang diperkenalkan oleh McCall, didapatkan beberapa kesimpulan, yaitu:

1. Berdasarkan penilaian secara kualitatif yang didapatkan dari berbagai fakta terkait, StarUML dinilai memenuhi faktor-faktor : *efficiency, integrity, flexibility, reusability, interoperability, expandability, usability, dan maintainability*
2. Sementara itu terdapat beberapa faktor yang belum dapat ditentukan dan perlu untuk dikaji lebih lanjut, yaitu: *correctness, reliability, testability, verifiability, manageability, dan safety*.

Dengan demikian, hasil review ini diharapkan dapat menjadi umpan balik bagi pengembang perangkat lunak StarUML dan sekaligus dapat dijadikan pertimbangan oleh pengguna untuk menggunakan StarUML.

DAFTAR PUSTAKA

- [1] Pressman, Roger S. 2000. Software Engineering: A Practitioner's Approach. Mc Graw Hill Education.
- [2] Fitzpatrick, Ronan. 1996. Software Quality: Definition and Strategic Issues. Staffordshire University, School of Computing Report.
- [3] Galin, Daniel. 2004. Software Quality Assurance from Theory to Implementation. Pearson/Addison Wesley.
- [4] Venners, Bill. 2005. *Erich Gamma on Flexibility and Reuse: A Conversation with Erich Gamma, Part II*. Aritma <URL: <http://www.artima.com/lejava/articles/reuse.html>>
- [5] *Detailed System Requirements for IBM Rational Rose v7.0*. <URL: <http://www-01.ibm.com/support/docview.wss?uid=swg21240715>>
- [6] *System Requirements (JUDE)*. <URL: <http://jude.change-vision.com/jude-web/product/system.html>>
- [7] *StarUML Bugs Report* <URL: <http://sourceforge.net/p/staruml/bugs/>>
- [8] *Wine HQ StarUML 5.0 Test Result* <URL: <https://appdb.winehq.org/objectManager>.
- [9] <https://appdb.winehq.org/objectManager.php?b>ShowAll=true&blsQueue=false&blsRejected=false&class=version&title=&returnTo=&id=7423>> *bject-oriented Software Construction*. Prentice Hall
- [10] StarUML: Sophisticated Modeling <URL: blog.staruml.io>
- [11] About StarUML <URL: <http://staruml.sourceforge.net/v1/about.php>>