

## PENGUKURAN ASPEK USABILITAS DENGAN ISO 9126 UNTUK PERANGKAT LUNAK BERBASIS KOMPONEN

Lastri Widya Astuti

Dosen Fakultas Ilmu Komputer Universitas Indo Global Mandiri Palembang

E-mail : [Lastriwidya@yahoo.com](mailto:Lastriwidya@yahoo.com)

### ABSTRACT

*Perangkat lunak berbasis komponen banyak digunakan pada pengembangan sistem berskala besar dengan cara mengintegrasikan komponen – komponen perangkat lunak yang ada. Pada proses pengintegrasian komponen sering ditemukan beberapa komponen yang tidak kompatibel dengan sistem baru. Dalam rangka mewujudkan penggunaan kembali komponen secara efektif dan efisien maka perlu dibuat aturan pengukuran (metric) untuk mengetahui apakah komponen fungsional dan non fungsional telah memenuhi standar untuk digunakan pada sistem baru. Aturan pengukuran (metric) yang dihasilkan akan diterapkan untuk mengukur komponen sehingga komponen yang sudah terstandarisasi dapat digunakan pada sistem yang berbeda. Makalah ini menitikberatkan pada aturan pengukuran (metric) untuk aspek usabilitas.*

*Kunci - Komponen, Kegunaan-ulang, usabilitas dan metric.*

### PENDAHULUAN

Pembangunan perangkat lunak berbasis komponen, menekankan pada desain dan konstruksi sistem menggunakan komponen perangkat lunak yang dapat digunakan kembali. Perangkat lunak berbasis komponen mampu mengurangi biaya pengembangan dan peningkatan keandalan seluruh sistem perangkat lunak. Keuntungan utama perangkat lunak berbasis komponen adalah biaya rendah, hemat waktu, solusi berkualitas tinggi, Produktivitas yang lebih tinggi, fleksibilitas, usabilitas, replaceability, efisien, perawatan, dan skalabilitas adalah beberapa manfaat tambahan perangkat lunak berbasis komponen. Dalam Survei terbaru yang dilakukan pada pengembangan perangkat lunak berbasis komponen dari 118 perusahaan dari seluruh dunia, ditemukan bahwa sekitar 53% organisasi yang menggunakan pendekatan perangkat lunak berbasis komponen dalam pengembangannya. Jika ada sejumlah komponen yang tersedia, maka perlu dibuat standarisasi komponen berdasarkan atribut dan sub – attribute untuk mengetahui berbagai karakteristik komponen.

Ada beberapa definisi yang berkaitan dengan komponen, beberapa di antaranya:

a. Sebuah komponen perangkat lunak adalah potongan kode yang dapat digunakan kembali atau perangkat lunak dalam bentuk biner, yang dapat dipasang ke komponen dari vendor lainnya dengan relatif sedikit usaha.

b. Sebuah komponen perangkat lunak adalah satuan komposisi dengan ditentukan dalam kontrak interface, sebuah komponen perangkat lunak dapat digunakan mandiri.

c. Sebuah komponen perangkat lunak adalah bahasa netral, independen yang diimplementasikan pada layanan paket perangkat lunak, disampaikan dalam sebuah paket dan bisa diubah serta diakses melalui satu atau lebih antarmuka.

d. Sebuah komponen perangkat lunak adalah satuan paket, distribusi atau pengiriman yang menyediakan layanan dalam integritas data atau batas enkapsulasi. Singkatnya, sebuah komponen perangkat lunak adalah dapat digunakan kembali, selfcontained bagian dari perangkat lunak dalam bentuk biner ditentukan dengan baik antarmuka yang independen dari aplikasi manapun.

Jika ada begitu banyak komponen yang tersedia, maka diperlukan untuk merancang beberapa aturan pengukuran ( metric) untuk mengkualifikasi berbagai karakteristik komponen. Berdasarkan ISO 9126 maka kualitas perangkat lunak ditandai dengan melihat kelengkapan unsur attribut dan sub – attribute yang merupakan standarisa. Berdasarkan ISO 9126 ada beberapa karakteristik mutu, namun dalam penelitian ini akan membahas unsur usabilitas yang merupakan unsur penting dalam pemakaian ulang komponen. Pengukuran usabilitas komponen dimaksudkan untuk mewujudkan penggunaan kembali komponen secara efektif.

## I. KARAKTERISTIK KUALITAS KOMPONEN

Selain fungsionalitas dari sebuah komponen, aspek non fungsionalitas juga menjadi bahan pertimbangan dalam pemilihan komponen yang akan dipakai ulang. Dalam konteks membangun sistem dari komponen-komponen yang ada, karakterisasi dari komponen "ilities" dan dampaknya terhadap sistem yang disertakan sangat penting. Karakteristik komponen yang berkualitas baik akan dibahas di bawah ini:

- a. *Understandability* : Upaya untuk membuat pengguna mengenali dan memahami konsep dasar komponen dan penerapannya. *Understandability* lebih lanjut dapat dihubungkan dengan dokumentasi. Dokumentasi komponen mencakup bagaimana menggunakan komponen (misalnya user manual) dan bagaimana mengkonfigurasi komponen (misalnya setup atau referensi manual). Jika dokumentasi komponen disediakan, pengguna dapat dengan mudah memahami konsep dan penggunaan komponen
- b. *Adaptability* : Kemampuan komponen untuk menampung perubahan atau kemudahan sebuah komponen untuk dimodifikasi agar dapat digunakan dalam lingkungan aplikasi yang berbeda.
- c. *Interoperability* : kemampuan suatu komponen untuk berinteraksi dengan komponen lain dari sebuah host sistem perangkat lunak dalam cara yang sempurna, atau kemampuan komponen untuk beradaptasi dengan sedikit usaha untuk mencapai tujuan yang sama. Dependensi pada komponen lain dapat dinilai oleh antarmuka yang diperlukan dan disediakan oleh komponen antarmuka yang membutuhkan.
- d. *Portability* : Kemampuan suatu komponen untuk beroperasi pada berbagai platform komputer dengan sedikit modifikasi, jika dibutuhkan.
- e. *Genericity* : Sebuah komponen yang dapat digunakan kembali harus generik, yaitu memiliki fitur yang tepat memungkinkan reuser untuk mengaplikasikan komponen. Penilaian *genericity* berkaitan dengan betapa sulitnya meletakkan komponen

operasi seperti instalasi, uninstalasi dan mengatur fitur.

- f. *Dependability* : properti ini mengacu pada keandalan komponen yang memungkinkan kepercayaan untuk menjadi pantas ditempatkan pada layanan yang ditawarkannya. Ini meliputi aspek keandalan, ketersediaan, keselamatan, keamanan, kegunaan dan *extendibility*.
  - g. *Cohesive* : Kemampuan melakukan layanan terkait tetapi tidak spesifik (khusus) sehingga berpotensi untuk digunakan kembali pada aplikasi yang lain (*reuse*).
  - h. *Independence* : Sebuah komponen harus independen dari komponen lainnya. Meminimalkan ketergantungan pada komponen lain membuat komponen lebih berguna dan mudah dioperasikan. Komponen kompleks cenderung memiliki beberapa dependensi dengan komponen lainnya.
  - i. *Having well defined Interface* : Dalam pengembangan berbasis komponen, setiap komponen akan memberikan (*ekspor*) dan membutuhkan (*impor*) pra-layanan dari komponen lainnya. Antarmuka komponen memainkan peran utama dalam perekatan dan hubungan antar komponen
- Meskipun sifat – sifat teknis dari penggunaan ulang komponen harus dimiliki, beberapa sifat-sifat non-teknis juga harus dipertimbangkan seperti kinerja bisnis masa lalu dan pengembang organisasi. Beberapa properti pendukung tersebut dapat diuraikan sebagai berikut:
- a. Didokumentasikan dengan baik - dokumentasi yang baik membantu user untuk menilai kredibilitas dan memvalidasi kualitas fungsional dan properti dari komponen, dokumentasi yang menyeluruh adalah salah satu cara. Informasi dalam dokumentasi komponen dibagi menjadi empat kategori - informasi umum, informasi rinci, uji penerimaan, dukungan dan informasi tambahan (seperti *debug*)
  - b. Kesesuaian dengan standar - Standardisasi adalah pendekatan yang baik untuk mengembangkan pasar komponen. Sebuah model komponen harus memenuhi standar-standar dan konvensi yang ditetapkan pada pengembangan komponen. Standarisasi membantu dalam mengurangi kemungkinan

- ketidakcocokan yang menghambat komposisi komponen.
- c. Ketersediaan informasi terkait - Beberapa persyaratan dan kegiatan pengujian potensial dapat mengurangi masalah yang terkait dengan penggunaan
  - d. kembali komponen dan dengan demikian meningkatkan kualitas perangkat lunak yang dihasilkan.
  - e. Dukungan Vendor - sebuah komponen akan bertahan di pasaran jika didukung oleh vendor dan stabil dari segi ekonomi serta adanya jaminan bahwa kode sumber tetap tersedia meskipun vendor tidak lagi memasok komponen.
  - f. Sertifikasi - jangka panjang keberhasilan teknologi komponen perangkat lunak bergantung pada ketersediaan komponen kualitas tinggi dan kepercayaan pihak konsumen bahwa komponen yang mereka beli berkualitas tinggi. Sertifikasi komponen dapat memberikan keyakinan penuh pada user bahwa komponen yang digunakan memiliki kehandalan. Kualitas sangat penting dalam sebuah sistem, hal ini dimaksudkan agar tidak ada efek samping yang menyebabkan biaya tambahan lebih tinggi ketika komponen digunakan pada suatu sistem.

## II. USABILITAS

Pada kasus pembangunan berbasis komponen, penggunaan kembali perangkat lunak I mengacu untuk pemanfaatan komponen perangkat lunak C dalam produk P, di mana motivasi asli untuk membangun C berbeda dengan digunakan dalam P. Dengan kata lain, menggunakan kembali adalah proses adaptasi komponen yang umum untuk berbagai konteks penggunaan. Gagasan tentang penggunaan kembali perangkat lunak yang mengandung beberapa keuntungan, antara lain: meningkatkan produktivitas, Kemampuraawatan, portabilitas dan kualitas sistem perangkat lunak. penggunaan kembali Sebuah komponen dapat dipandang sebagai kotak, yang berisi kode dan dokumentasi . Dimana kotak ini mendefinisikan sebagai berikut :

### A. *Black Box reuse*

*Black Box Reuse*, pengguna melihat antarmuka, bukan pelaksanaan komponen. Antarmuka berisi metode penggunaan

dokumentasi umum, persyaratan dan pembatasan dari komponen. Jika seorang programmer mengubah kode komponen, kompilasi dan menghubungkan komponen akan menyebabkan perubahan pada aplikasi yang menggunakan komponen. Kepercayaan pengguna komponen terhadap interface, perubahan seharusnya tidak mempengaruhi perilaku logis komponen. User akan mendapatkan komponen yang dibutuhkan walupun terjadi perubahan pelaksanaan pada internal komponen.

### B. *Glass Box Reuse*

*Glass Box Reuse*, bagian dalam kotak dapat dilihat dengan baik sebagaimana bagian luar, tetapi tetap tidak dimungkinkan untuk menyentuh bagian dalam. Solusi ini memiliki keuntungan jika dibandingkan dengan menggunakan kembali Black Box, pengguna dapat memahami komponen dan mengoptimalkan penggunaannya. Kelemahannya adalah bahwa adalah kemungkinan bahwa pengguna akan bergantung pada cara tertentu pelaksanaan atau faktor-faktor lain yang tidak terdapat dalam kontrak, hal ini bisa berbahaya jika terjadi perubahan pada pelaksanaan.

### C. *White Box Reuse*

*White Box Reuse*, memungkinkan untuk melihat dan mengubah isi dalam kotak serta antarmukanya. Sebuah White Box dapat berbagi struktur internal atau pelaksanaan dengan Box yang lain melalui warisan atau delegasi. Komponen baru dapat mempertahankan komponen yang digunakan kembali seperti standard atau dapat mengubahnya. Hal ini diperlukan untuk menguji sesuatu yang baru yang diciptakan atau diubah. Salah satu masalah penting dalam penggunaan kembali perangkat lunak adalah pengambilan dan pemilihan komponen perangkat lunak yang sesuai dari komponen *library*. Menurut Gill [15] membahas karakterisasi komponen penting untuk usability yang lebih baik. Hal ini membahas beberapa manfaat dari karakterisasi komponen , yang mencakup peningkatan katalog, peningkatan penggunaan dan peningkatan pemahaman yang lebih baik tentang pemakaian ulang komponen. Beberapa definisi dari usability berdasarkan literatur:

- a. Kemampuan perangkat lunak (komponen) untuk dipahami , dipelajari, digunakan dan

menarik bagi pengguna, jika digunakan di bawah kondisi tertentu (ISO / IEC 9126-1:2000).

- b. Sejauh mana suatu produk dapat digunakan oleh pengguna untuk mencapai tujuan tertentu dengan efektivitas, efisiensi dan kepuasan dalam konteks penggunaan tertentu (ISO 9241-11:1998).
- c. Kemudahan bagi pengguna agar dapat belajar untuk mengoperasikan, menyiapkan input, dan menginterpretasikan output dari sistem atau komponen (IEEE Std.610.12-1990).
- d. Kegunaan dari produk perangkat lunak adalah sejauh mana produk tersebut mudah dan praktis untuk digunakan (Barry Boehm, 1978).
- e. Probabilitas bahwa sistem operator tidak akan mengalami masalah antarmuka pengguna selama periode tertentu operasi di bawah suatu profil operasional (Fenton, 1998).

ISO 9126 mendefinisikan Usability dalam lima sub-karakteristik: Understandability, Learnability, Operability, Attractiveness, dan Usability Compliance.. Dalam kasus komponen, mereka dapat didefinisikan sebagai berikut.:

- a. *Understandability*: kemampuan komponen untuk memungkinkan pengguna (pengembang sistem) untuk memahami apakah komponen yang digunakan ulang cocok. Pengembang sistem harus dapat memilih komponen yang sesuai dengan tujuan penggunaannya. Sebagai contoh, unsur-unsur komponen (misalnya antarmuka, operasi) harus mudah dimengerti atau jelas.
- b. *Learnability*: kemampuan komponen perangkat lunak untuk memungkinkan pengguna (pengembang sistem) untuk mempelajari aplikasi. Sebuah metrik

learnability harus mampu menilai berapa lama pengembang sistem mempelajari bagaimana menggunakan fungsi-fungsi tertentu (interface, operasi), atau efektivitas dokumentasi (manual, sistem bantuan, demo). Sebagai contoh, pengguna dokumentasi dan sistem bantuan harus lengkap.

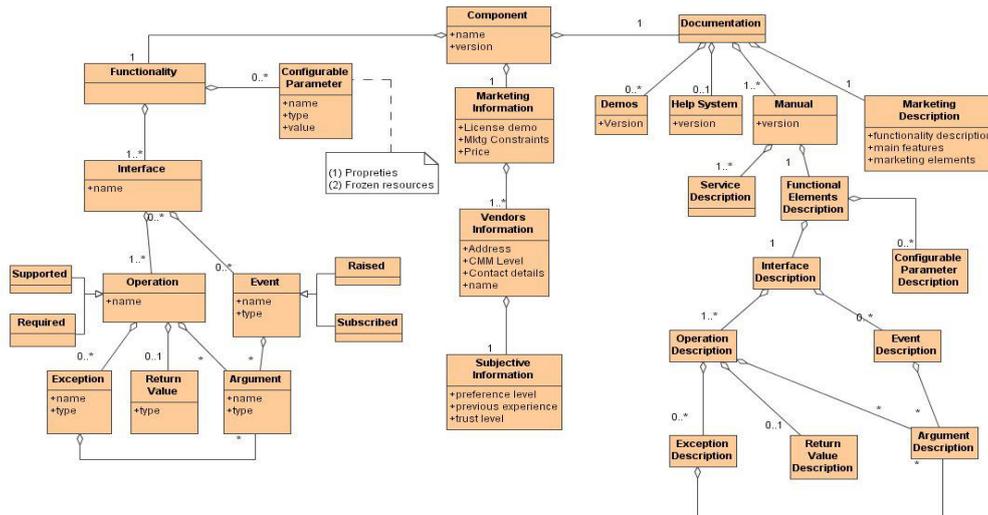
- c. *Operability*: kemampuan komponen perangkat lunak untuk memungkinkan pengguna (pengembang sistem) untuk mengoperasikan dan mengendalikannya. Sebuah metrik operability harus bisa menilai apakah pengembang sistem dapat mengoperasikan dan mengendalikan komponen. Operability metrik dapat dikategorikan dengan prinsip – prinsip dialog dalam ISO 9.241-10:
  1. Kesesuaian tugas komponen
  2. Pengendalian komponen
  3. Kesesuaian komponen dengan harapan pengguna (persyaratan)
  4. Toleransi kesalahan dari komponen
  5. kesesuaian komponen untuk individualisation.

d. *Attractiveness*: kemampuan komponen perangkat lunak yang menarik bagi pengguna.

e. *Usability Compliance*: kemampuan komponen perangkat lunak untuk mematuhi standar, konvensi, gaya panduan atau peraturan yang berkaitan dengan usability.

#### IV. METRIK USABILITAS

Isu penting dalam memilih komponen terbaik untuk usability adalah memutuskan komponen yang lebih mudah diadaptasi. Secara umum, pedoman yang baik untuk memprediksi usability adalah: ukuran kode yang kecil, struktur yang sederhana dan dokumentasi yang baik



Gambar 1. Informasi Komponen Perangkat Lunak

ISO 9126 merupakan salah satu model kualitas yang telah distandarisi secara internasional. Perbedaan dengan model kualitas yang lain terdapat pada hubungan antar faktor kualitas dimana ISO 9126 memiliki hubungan 1:N, maksudnya setiap karakteristik memiliki subkarakteristik dan tidak berhubungan dengan satu atau lebih karakteristik yang lainnya. Istilah dokumentasi komponen pada gambar 1 mengacu ke manual, demo, sistem bantuan, dan informasi pemasaran. Meskipun beberapa metrik dapat didefinisikan pada setiap unsur (misalnya, jumlah operasi per antarmuka), kadang-kadang sulit untuk merujuk pada elemen fungsional secara terpisah. Tentu saja, tidak semua informasi ini tersedia untuk semua

komponen komersial, penentuana unsur-unsur yang akan diukur mempertimbangkan tuntutan informasi dan keinginan vendor. Untuk mengevaluasi usability dari serangkaian komponen perangkat lunak yang kandidat akan diintegrasikan dan dipilih sebagai komponen perangkat lunak berkualitas. Pengembang sistem memilih yang paling mudah digunakan (mengintegrasikan) antara seperangkat komponen perangkat lunak yang menyediakan fungsionalitas yang diperlukan. Ada tiga konsep terukur terkait dengan komponen perangkat lunak kegunaan: Kualitas Dokumentasi, Kompleksitas Masalah dan kompleksitas Solusi (Design)

Tabel 1 Konsep Terukur dan Atribut untuk Usability

ENTITY	INFORMATION NEED	MEASURABLE CONCEPT		ATTRIBUTE
SOFTWARE COMPONENT	EVALUATE THE USABILITY	QUALITY OF DOCUMENTATION	QUALITY OF MANUALS	CONTENTS OF MANUALS
				SIZE OF MANUALS
				EFFECTIVENESS OF MANUALS
			QUALITY OF DEMOS	CONTENTS OF DEMOS
				QUALITY OF HELP SYSTEM
			QUALITY OF HELP SYSTEM	CONTENTS OF HELP SYSTEM
				SIZE OF HELP SYSTEM
				EFFECTIVENESS OF HELP SYSTEM
		QUALITY OF MARKETING INFO	CONTENTS OF MARKETING INFO	
			EFFECTIVENESS OF MARKETING INFO	
			COMPLEXITY OF THE DESIGN	DESIGN LEGIBILITY (READABILITY)
				INTERFACES UNDERSTANDABILITY
				I/O UNDERSTANDABILITY
				EASE OF LEARNING
CUSTOMISABILITY				
QUALITY OF ERROR MESSAGES				
INTERFACES COMPLEXITY				

Atribut yang terdapat pada tabel 1 dapat dibagi lagi menjadi subatribut sehingga pengukuran

(metric) usability menjadi lebih detil dan rinci, indicator dan indirect metric disajikan pada tabel 2

ATTRIBUTE	INDICATOR	INDIRECT METRIC
CONTENTS OF MANUALS	MANUALS COVERAGE	PROPORTION OF FUNCTIONAL ELEMENTS DESCRIBED IN MANUALS
	MANUALS CONSISTENCY	PROPORTION OF FUNCTIONAL ELEMENTS INCORRECTLY DESCRIBED IN THE MANUAL
		COMPLETENESS OF MANUALS
		DIFFERENCE BETWEEN THE COMPONENT VERSION AND THE MANUAL VERSION
	MANUALS LEGIBILITY	RATIO OF FIGURES PER MANUAL PAGES
		RATIO OF TABLES PER MANUAL PAGES
RATIO OF UML DIAGRAMS PER MANUAL PAGES		
SIZE OF MANUALS	MANUALS SUITABILITY	AVERAGE PAGES PER FUNCTIONAL ELEMENTS
EFFECTIVENESS OF MANUALS	EFFECTIVENESS RATIO	PROPORTION OF FUNCTIONAL ELEMENTS CORRECTLY USED AFTER READING THE MANUAL
	UNDERSTANDABILITY RATIO	PROPORTION OF FUNCTIONAL ELEMENTS CORRECTLY UNDERSTOOD AFTER READING THE MANUAL
CONTENTS OF DEMOS	DEMONSTRATION COVERAGE	PROPORTION OF FUNCTIONAL ELEMENTS SHOWED IN DEMOS
	DEMONSTRATION CONSISTENCY	DIFFERENCE BETWEEN DEMO VERSION AND COMPONENT VERSION
CONTENTS OF HELP SYSTEM	HELP SYSTEM COVERAGE	PROPORTION OF FUNCTIONAL ELEMENTS SHOWED IN HELP SYSTEM
	HELP SYSTEM CONSISTENCY	PROPORTION OF FUNCTIONAL ELEMENTS INCORRECTLY DESCRIBED IN HELP SYSTEM
		COMPLETENESS OF HELP SYSTEM
SIZE OF HELP SYSTEM	HELP SYSTEM SUITABILITY	HELP SYSTEM WORD RATIO
EFFECTIVENESS OF HELP SYSTEM	HELP SYSTEM EFFECTIVENESS RATIO	PROPORTION OF FUNCTIONAL ELEMENTS CORRECTLY USED AFTER USING THE HELP SYSTEM
	HELP SYSTEM UNDERSTANDABILITY RATIO	PROPORTION OF FUNCTIONAL ELEMENTS CORRECTLY UNDERSTOOD AFTER USING THE HELP SYSTEM
CONTENTS OF MARKETING INFO	COVERAGE OF MARKETING INFO	NUMBER OF MARKETING INFO ELEMENTS DESCRIBED
	COMPLETENESS OF MARKETING INFO	PROPORTION OF SERVICES UNDERSTOOD AFTER READING THE COMPONENT DESCRIPTION
	MARKETING INFO CONSISTENCY	DIFFERENCE BETWEEN THE COMPONENT VERSION AND THE MARKETING INFO VERSION
EFFECTIVENESS OF MARKETING INFO	MARKETING INFO UNDERSTANDABILITY RATIO	PROPORTION OF SERVICES UNDERSTOOD AFTER READING THE COMPONENT DESCRIPTION

Tabel 2 Metrik diasosiasikan dengan Kualitas Dokumentasi

Pada titik ini, kita perlu menentukan hubungan antara atribut termasuk metrik yang terukur Usability untuk menilai konsep yang berkaitan dengan atribut (particularized) kualitas model yang diusulkan oleh ISO 9126. Dengan demikian, kita harus mendefinisikan sebuah hubungan antara Kualitas Dokumentasi dan Desain Kompleksitas ke Understandability, Learnability dan Operability. Oleh karena itu bahwa tidak ada hubungan langsung yang unik antara metrik dan kualitas subcharacteristic, tetapi ada perbedaan tingkat hubungan antara setiap metrik dengan masing-masing subkarakteristik.

Dari beberapa kali percobaan pengukuran menunjukkan hubungan antara atribut dan kegunaan subkarakteristik maka diketahui

bahwa Kualitas Dokumentasi terutama mempengaruhi Understandability dan Learnability. Di sisi lain, Kompleksitas Desain mempengaruhi untuk Operability dengan derajat (Low, Medium, High), uraian dapat dilihat pada tabel berikutnya:

Tabel 3 Hubungan Atribut Perangkat Lunak Komponen dengan Usabilitas Subkarakteristik

ATTRIBUTE/QUALITY CHAR.	UNDERSTANDABILITY	LEARNABILITY	OPERABILITY
CONTENTS OF MANUALS	LOW	HIGH	MEDIUM
SIZE OF MANUALS	LOW	HIGH	MEDIUM
EFFECTIVENESS OF MANUALS	LOW	HIGH	HIGH
CONTENTS OF DEMOS	HIGH	LOW	LOW
CONTENTS OF HELP SYSTEM	-	HIGH	HIGH
SIZE OF HELP SYSTEM	-	HIGH	MEDIUM
EFFECTIVENESS OF HELP SYSTEM	-	HIGH	HIGH
CONTENTS OF MARKETING INFO	HIGH	-	-
EFFECTIVENESS OF MARKETING INFO	HIGH	-	-
DESIGN'S LEGIBILITY (READABILITY)	MEDIUM	LOW	HIGH
INTERFACES UNDERSTANDABILITY	LOW	LOW	HIGH
UNDERSTANDABILITY OF I/O	HIGH	LOW	HIGH
EASE OF COMPONENT LEARNING	-	HIGH	MEDIUM
CUSTOMISABILITY	LOW	MEDIUM	HIGH
CONTENTS OF ERROR MESSAGE	LOW	LOW	HIGH
INTERFACES DENSITY	LOW	HIGH	HIGH

#### IV. KESIMPULAN

Dalam makalah ini membahas tentang pengukuran aspek usabilitas pada perangkat lunak berbasis komponen. Metrik pengukurannya menggunakan ISO 9126 yang merupakan standarisasi untuk kualitas perangkat lunak. Kualitas Dokumentasi terutama mempengaruhi Understandability dan Learnability. Di sisi lain, Kompleksitas Desain mempengaruhi untuk Operability. Penetapan atribut dan sub – atribut untuk pengukuran sangat dibutuhkan karena akan membuat hasil

pengukuran menjadi lebih terarah dan terstruktur.

#### V. FUTURE WORK

Penetapan nilai standarisasi pengukuran perangkat lunak berbasis komponen agar seluruh komponen berada pada frame yang sama dan lebih terukur. Pembuatan alat bantu (tool) untuk membantu mengukur nilai standard komponen sehingga dapat diketahui kelas atau tingkatan dari sebuah komponen yang akan dilempar ke pasaran.

## REFERENSI

- [1] Arun Sharma, Rajesh Kumar, and P. S. Grover: A Critical Survey of Reusability Aspects for Component-Based Systems: World Academy of Science, Engineering and Technology 33 2007
- [2] Arun Sharma, Rajesh Kumar, P S Grover<sup>2</sup> Managing Component-Based Systems With Reusable Components, International Journal of Computer Science and Security, Volume 1 : Issue (2)
- [3] Kuljit Kaur, Parminder Kaur, Dr. Hardeep Singh, Quality Constraints on Reusable Components
- [4] K.S.Jasmine, Vasanta.R, DRE-A Quality Metric for Component Based Software Product, World Academy of Science, Engineering and Technology 34 2007
- [5] Jean-Guy Schneider: "Component Scripts and Glue: A Conceptual framework for software composition" Ph.D. thesis, Institute für Informatik (IAM), Universität Bern, Berne, Switzerland, 2003.
- [6] Michael Sparling: "Lessons Learned through Six Years of Component Based Development" published in Castek (as published in "Communications of the ACM") date:04-09-03.
- [7] Szyperski C: Component Software: beyond Object Oriented Programming, New York: ACM Press/ Addison Wesley 1998.
- [8] Hironori Washizaki, Hirokazu Yamamoto and Yoshiaki Fukazawa: "A Metrics Suite for Measuring Reusability of Software Components", Proceedings of the 9th International Symposium on Software Metrics September 2003.
- [9] Miguel Goulão: "CBSE: a Quantitative Approach" PhD Workshop at ECOOP'2003, Darmstadt, Germany. July, 2003.