

PENGUKURAN KUALITAS PERANGKAT LUNAK SISTEM MANAJEMEN PELAPORAN KEGIATAN BERBASIS WEB PERINGATAN BERBASIS EMAIL

Ritzkal^{1*} dan Moh. Subchan².

1. Universitas Ibnu Khaldun

Jl. KH. Sholeh Iskandar KM.2, Kedung Badak, Tanah Sereal, Kota Bogor 16162 Jawa Barat
Telp. 0251-8380993 Fax 0251-8380993, 0817145016
Website: <http://www.uika-bogor.ac.id> E-mail: ritzkal@ft.uika-bogor.ac.id

2. STMIK Muhammadiyah Banten

Jl. Syekh Nawawi Km. 4 No. 13 Matagara Tigaraksa Kab. Tangerang
Telp. 021-22599028, 085888877388
Website: <http://www.stmikmb.ac.id>
E-mail: mohamad.subchan@stmikmbanten.ac.id

Abstrak – Pentingnya kualitas perangkat lunak memang masih diperhadapkan pada banyaknya standar kualitas yang ada. Para ahli rekayasa perangkat lunak tampaknya masih memerlukan sedikit waktu untuk menetapkan sebuah standar tunggal yang dapat menggugurkan standar-standar kualitas yang banyak ini. Di sisi lain, pencapaian kualitas perangkat lunak tidak bisa menunggu sampai standar tunggal itu ditetapkan. Tujuan dari penelitian yaitu mendapatkan nilai kualitas perangkat lunak sistem manajemen pelaporan. Penelitian ini dilakukan dengan teknik kuantitatif. Penelitian ini adalah penerapan dari teori-teori mengenai kualitas perangkat lunak, dan dilakukan sebagai pembuktian teori-teori tersebut. Secara skematik, Metode penelitian dilakukan dengan beberapa tahapan kumpulan data, komputasi numerik (metric) dan komputasi numerik (indikator). Hasil yang diperoleh berupa pengumpulan data pada perangkat lunak, perhitungan komputasi numerik dan perhitungan indicator kualitas ISO 9126 yang meliputi indikator kualitas fungsional, indicator kualitas reliabilitas, indicator kualitas usabilitas, indicator kualitas efisiensi, indicator kualitas maintainabilitas, dan indicator kualitas portabilitas. Berdasarkan hasil tersebut, maka ditarik kesimpulan sesuai tujuan penelitian dengan hasil pengujian indikator kualitas ISO 9126 yang terdiri dari hasil 0.1435 fungsionalitas, hasil 988 reliable, hasil 0.4429 usability, hasil 11.5316 person-month, 2.33 Pemeliharaan, hasil 5 portabilitas dan persentase pencapaian kualitas 83.3%.

Kata kunci: *Iso 9126, Persentase pencapaian kualitas, indikator, perangkat lunak*

1 Pendahuluan

Pencapaian kualitas perangkat lunak dapat dinilai melalui pengukuran. Ada banyak atribut-atribut perangkat lunak yang dapat diukur. Jumlah atribut yang digunakan dalam pengukuran tergantung pada banyaknya informasi yang ingin diperoleh melalui pengukuran. Contohnya, ketika seorang manager proyek ingin memperoleh informasi mengenai tingkat keandalan dari perangkat lunak yang dikembangkan maka atribut-atribut yang diukur adalah seperti jumlah kesalahan yang mungkin terjadi dalam kurun waktu tertentu, jumlah fungsi, jumlah baris kode, kerumitan, dan ujicoba

yang dilakukan untuk memastikan tingkat kesalahan yang mungkin terjadi selama proses pengembangan perangkat lunak. Pada akhirnya informasi-informasi tersebut akan digunakan untuk mendukung fungsi-fungsi manajemen seperti perencanaan, pengorganisasian, pelaksanaan, dan pengendalian[1]. ISO 9126 merupakan sebuah standard internasional yang diterbitkan oleh ISO untuk evaluasi kualitas perangkat lunak dan merupakan pengembangan dari ISO 9001. Standar ini dibagi menjadi empat bagian yang masing-masing menjelaskan model kualitas yaitu, metrik eksternal, metrik internal, dan metrik kualitas yang

digunakan. Ada enam ukuran kualitas yang ditetapkan oleh ISO 9126, yaitu fungsionalitas, kehandalan (*reliability*), kebergunaan (*usability*), efisiensi, portabilitas, serta keterpeliharaan (*maintainability*)[2].

2 Landasan Teori

Pengukuran merupakan kegiatan untuk mengindikasikan kuantitatif dari luasan, jumlah, dimensi, kapasitas, atau atribut sebuah proses atau produk. Pengukuran dikategorikan menjadi dua cara yaitu pengukuran langsung (contohnya jumlah mahasiswa yang terdapat pada suatu perguruan tinggi) dan pengukuran tidak langsung (contohnya kualitas pendidikan pada suatu perguruan tinggi yang didapat dari menghitung penolakan calon mahasiswa).

Pengukuran langsung pada perangkat lunak adalah mengenai biaya dan usaha yang dibutuhkan untuk sebuah produk perangkat lunak tersebut. Beberapa contoh pengukuran langsung perangkat lunak adalah seperti pengukuran jumlah baris kode, fungsi, kecepatan eksekusi, ukuran memori, dokumentasi, jumlah masukan, keluaran, dan cacat dalam satuan tertentu. Pengukuran tidak langsung pada perangkat lunak contohnya adalah pengukuran kualitas. Fungsionalitas, efisiensi, reliabilitas, dan kemampuan pemeliharaan merupakan sebagian atribut kualitas perangkat lunak yang pengukurannya dilakukan secara tidak langsung.

Pengukuran terjadi ketika pengumpulan data mulai dilakukan. Ukuran kuantitatif dari tingkatan suatu atribut objek, sistem atau proses disebut dengan metrik. Untuk dapat mengukur kualitas perangkat lunak, metrik-metrik harus dikumpulkan untuk mendapatkan indikator. Indikator merupakan sebuah metrik atau kombinasi dari beberapa metrik yang menjadi sebuah informasi utuh terhadap suatu produk. Informasi yang didapatkan dari indikator-indikator tersebut dapat membantu pihak manajemen dalam mengendalikan proses dan produk yang dibuat.

3 Metodologi Penelitian

Metode penelitian dilakukan dengan beberapa tahapan kumpulan data, komputasi numerik (metric) dan komputasi numerik (indikator). Hasil yang diperoleh berupa pengumpulan data pada perangkat lunak, perhitungan komputasi numerik pada metric dan perhitungan komputasi numerik dengan indicator kualitas ISO 9126 yang meliputi indikator kualitas fungsional, indikator kualitas reliabilitas, indikator kualitas usabilitas, indikator kualitas efisiensi, indikator kualitas maintainabilitas, dan indikator kualitas portabilitas.

4 Hasil dan Pembahasan

4.1. Hasil Pengamatan

Berdasarkan hasil pengamatan, maka didapatkanlah data-data seperti yang diperlihatkan oleh table 1.

Tabel 1 Hasil Pengamatan

Nama Data	Item	Jumlah
Inputan Pengguna	Entry Data User	5
	Entry Data Unit Kegiatan	
	Entry Data Kegiatan	
	Pengaturan Ganti Password	
	Pengaturan Alamat Email	
Output Pengguna	Daftar Data User	11
	Daftar Unit Kegiatan	
	Daftar Data Kegiatan	
	Daftar Statistik Kegiatan UK & Bidang	
	Pesan Error Login	
	Laporan Kegiatan	
	Laporan Kegiatan Pdf	
	List Unit Kerja	
	List Bidang	
	List Tahun	
	Pencarian Data Kegiatan	
Permintaan Pengguna	Menu Home	14
	Menu Entry Kegiatan	
	Menu Pencarian	
	Menu Laporan	
	Menu Statistik	
	Menu Pengaturan	
	Menu Logout	
	Tombol Login	
	Tombol Ubah Data User	
	Tombol Hapus Data User	
	Tombol Ubah Unit Kerja	
	Tombol Hapus Unit Kerja	
	Tombol All Data Kegiatan	
Lihat Grafis		
Interface Eksternal	Tcp/IP	1
File	Data User	4
	Data Unit Kerja	
	Data Kegiatan	
	Data User	
Kegagalan	Format Laporan Kegiatan	4
	Format Laporan Kegiatan Grafis	
	Terjadinya Keterlambatan Pengiriman Email	
	Tombol Update Tidak Berfungsi	
Modul Program Saat Ini	Pengaturan Alamat Email	3
	Pengaturan Tahun	
	Manajemen Akun	
Modul Terganti	Manajemen Akun	2
	Pengaturan Tahun	
Modul Terhapus		0
Bahasa Pemrograman	PHP 5	1

4.2. Perhitungan Komputasi Numerik

Perhitungan komputasi numerik pada metric merupakan metric perangkat lunak berorientasi fungsi ditarik berdasarkan

sebuah pengukuran fungsionalitas yang disampaikan oleh aplikasi sebagai suatu nilai normalisasi. Karena fungsionalitas tidak dapat diukur secara langsung, maka fungsionalitas harus ditarik secara tidak langsung dari pengukuran langsung lainnya. Metrik berorientasi fungsi dibuat oleh Alan J. Albrecht (1979) yang disebut dengan function point. Saat ini ada banyak variasi cara perhitungan function point setelah metode ini dikembangkan dan direvisi oleh International Function Point User Group (IFPUG) sejak tahun 1986. Namun pada penelitian ini penulis akan memfokuskan penggunaan function point yang dibuat oleh Albrecht. Function point ditarik dengan menggunakan sebuah hubungan empiris berdasarkan pengukuran langsung domain informasi perangkat lunak yang dapat dihitung serta perkiraan kompleksitas perangkat lunak.

Param Pengukuran	Jumlah	X	Faktor Pembobolan			=	Nilai
			Sederhana	Rata - Rata	Kompleks		
J. Input Pengguna	5	X	3			=	15
J. Output Pengguna	11	X	4			=	44
J. Permintaan Pengguna	14	X		5		=	70
J. File	4	X		10		=	40
J. Interface External	1	x	6			=	6
Total							176

Gambar 1 Perhitungan Komputasi Numerik

4. 3. Perhitungan indikator kualitas ISO 9126

Setelah data dikumpulkan maka langkah berikutnya adalah mencari indikator kualitas ISO 9126 yaitu fungsionalitas, reliabilitas, usabilitas, efisiensi, pemeliharaan, dan portabilitas

4. 4. Fungsionalitas

Indikator fungsionalitas dapat ditarik dari function point. Perhitungan function point membutuhkan data berupa input pengguna, output pengguna, permintaan pengguna, file, dan interface eksternal. Setiap data tersebut harus dinilai kompleksitasnya secara umum yaitu sederhana, sedang atau kompleks. Dengan data dan penilaian tersebut, maka perhitungan function pointnya adalah sebagai berikut.

Param Pengukuran	Jumlah	X	Faktor Pembobolan			=	Nilai
			Sederhana	Rata - Rata	Kompleks		
J. Input Pengguna	5	X	3			=	15
J. Output Pengguna	11	X	4			=	44
J. Permintaan Pengguna	14	X		5		=	70
J. File	4	X		10		=	40
J. Interface External	1	x	6			=	6
Total							176

Gambar 2 Indikator Fungsionalitas

4. 5. Faktor perubah kompleksitas

Faktor peubah kompleksitas didapatkan dari penilaian keempatbelas atribut yang terdapat pada perangkat lunak. Keempatbelas atribut tersebut digunakan sebagai faktor untuk menormalisasi perhitungan function point. Perhitungan faktor peubah kompleksitas function point adalah sebagai berikut

No	Faktor Peubah	Nilai
1	Backup dan recovery	0 1 2 3 4 5
2	Komunikasi data	0 1 2 3 4 5
3	Distribusi pemrosesan data	0 1 2 3 4 5
4	Performa	0 1 2 3 4 5
5	Konfigurasi operasional	0 1 2 3 4 5
6	Inputan onlie (on-line entry)	0 1 2 3 4 5
7	Perubahan data on-line (on-line update)	0 1 2 3 4 5
8	Tingkat transaksi data	0 1 2 3 4 5
9	Efisiensi pengguna	0 1 2 3 4 5
10	Kompleksitas pemrosesan	0 1 2 3 4 5
11	Penggunaan kembali (reusability)	0 1 2 3 4 5
12	Konversi dan Instalasi	0 1 2 3 4 5
13	Penggandaan instalasi	0 1 2 3 4 5
14	Fasilitas perubahan (Facilitate Change)	0 1 2 3 4 5
TOTAL		30

Gambar 3 Faktor Perubah Kompleksitas

4. 6. Function Point

Diketahui : Jumlah Total = 176

$$\sum Fi = 30$$

Rumus untuk mencari function point adalah sebagai berikut.

$$FP = \text{Jumlah Total} \times (0,65 + 0,01 \sum Fi)$$

$$FP = 176 \times (0,65 + (0,01 \times 30))$$

$$FP = 236,2$$

$$FP_{Max} = 1,35 \times \text{Jumlah Total}$$

$$FP_{Max} = 1,35 \times 176$$

$$FP_{Max} = 237,6$$

Dengan keterangan:

$$FP = \text{Function Point}$$

$$FP_{Max} = \text{Function Point Maksimum}$$

$$\text{Jumlah Total} = \text{Nilai Total domain informasi}$$

$$\sum Fi = \text{Jumlah harga penyesuaian kompleksitas}$$

Berdasarkan function point dan function point maksimum, maka tingkat pencapaian fungsionalitas perangkat lunak e-learning adalah sebagai berikut.

$$\text{Diketahui: } FP = 34,1$$

$$FP_{Max} = 237,6$$

$$\text{Fungsionalitas} = \frac{FP}{FP_{Max}}$$

$$\text{Fungsionalitas} = \frac{34,1}{237,6}$$

$$\text{Fungsionalitas} = \frac{34,1}{237,6}$$

$$\text{Fungsionalitas} = 0,1435$$

Jadi, nilai fungsionalitas perangkat lunak Manajemen Pelaporan adalah sebesar 0,1435.

4. 7. Reliable

Indikator reliabilitas (keandalan) perangkat lunak didapatkan dari metrik *rate of failure occurrence (ROCOF)*. Untuk menghitung *ROCOF* dibutuhkan variabel function point (*FP*), dan jumlah kegagalan (*Failure*).

$$\text{Diketahui: } FP = 34,1$$

$$\text{Failure} = 4$$

$$ROCOF = \frac{\text{Kegagalan}}{\text{Function Point}}$$

$$ROCOF = \frac{4}{34,1}$$

$$ROCOF = \frac{4}{34,1}$$

$$ROCOF = 0,117$$

Reliabilitas dapat ditarik menggunakan rumus berikut.

$$\text{Reliabilitas} = 1 - ROCOF$$

$$\text{Reliabilitas} = 1 - 0,117$$

$$\text{Reliabilitas} = 0,9889$$

Dengan demikian maka reliabilitas perangkat lunak manajemen pelaporan adalah sebesar 0,9889. Nilai 0,9889 mengindikasikan bahwa dalam 1.000 kali pengoperasian, diperkirakan perangkat lunak aplikasi manajemen pelaporan mampu bekerja dengan benar selama 988 kali. Dengan kata lain dalam 1000 kali pengoperasian, aplikasi manajemen pelaporan diperkirakan mengalami kegagalan fungsi sebanyak 11 kali.

4. 8. Usability

Indikator usabilitas perangkat lunak didapatkan dari metrik *speed of operation*. Semakin *speed of operation* mendekati nilai 0 menandakan usabilitas semakin meningkat. Sebaliknya, semakin *speed of operation* mendekati nilai 1 maka usabilitas semakin menurun.

$$\text{Diketahui: } \text{Function Point (FP)} = 4,1$$

$$\text{Inputan Pengguna} = 5$$

$$\text{Permintaan Pengguna} = 14$$

$$\text{Speed of Operation} = \frac{\text{Inputan Pengguna} + \text{Permintaan Pengguna}}{\text{Function Point}}$$

$$\text{Speed of Operation} = \frac{5 + 14}{34,1}$$

$$\text{Speed of Operation} = 0,5571$$

$$\text{Usabilitas} = 1 - \text{Speed of Operation}$$

$$\text{Usabilitas} = 1 - 0,5571$$

$$\text{Usabilitas} = 0,4429$$

Jadi tingkat usabilitas perangkat lunak Manajemen Pelaporan adalah sebesar 0,4429. Nilai 0,4429 mengindikasikan bahwa pekerjaan yang seharusnya diselesaikan dalam 500 kali kerja, sekarang digantikan oleh sistem sebanyak 400 pekerjaan sehingga pekerjaan tersebut dapat diselesaikan hanya dengan 100 kali kerja. Atau lebih jelasnya, pekerjaan yang seharusnya diselesaikan dalam waktu 40 jam (1 minggu kerja), akan selesai dalam waktu 4 jam (setengah hari kerja).

4. 9. Efisiensi

Efisiensi berkaitan dengan performa, sumber daya yang diperlukan, dan penghematan yang diperoleh dari penggunaan produk. Untuk mendapatkan indikator efisiensi maka diperlukan beberapa variabel untuk menghitungnya. Variabel-variabel tersebut adalah total manfaat, total biaya, dan usaha yang diperlukan untuk membangun perangkat lunak.

4. 10. Metric Perkiraan

$$\text{Diketahui: } FP = 34,1$$

$$E = -13,39 + 0,0545 FP$$

$$E = -13,39 + (0,0545 \times 34,1)$$

$$E = 11,5316 \text{ person-month}$$

4. 11. Pemeliharaan

Pada perangkat lunak, semakin banyak perubahan yang terjadi pada perangkat lunak menandakan bahwa pemeliharaan akan semakin sulit untuk dilakukan. Untuk pencarian nilai maintainabilitas, metrik yang digunakan adalah metrik *Software Maturity Index (SMI)*. Semakin SMI mendekati nilai 1 maka produk akan semakin stabil. Sebaliknya semakin SMI menjauhi nilai 1 maka produk akan semakin tidak stabil. Variabel-variabel yang dibutuhkan untuk mencari nilai SMI adalah jumlah modul saat ini (*MT*), jumlah penambahan modul (*Fa*), jumlah modul yang mengalami perubahan (*Fc*), dan jumlah modul yang sudah dihapus sejak perancangan awal (*Fd*).

$$\text{Diketahui:}$$

$$MT = 3$$

$$Fa = 0$$

$$Fc = 2$$

$$Fd = 0$$

$$SMI = \frac{MT - (Fa + Fc + Fd)}{MT}$$

$$SMI = \text{Pemeliharaan}$$

$$\text{Pemeliharaan} = \frac{MT - (Fa + Fc + Fd)}{MT}$$

$$\text{Pemeliharaan} = \frac{3 - (0 + 2 + 0)}{3}$$

$$\text{Pemeliharaan} = 2,33$$

Indikator pemeliharaan sebesar 2,33 menandakan bahwa dari setiap 10 modul program, terdapat 3 modul yang diperkirakan stabil sehingga tidak membutuhkan perubahan yang berarti pada waktu pemeliharaan dan 1 modul lainnya yang diperkirakan akan mengalami perubahan pada tahap pemeliharaan.

4.12. Portabilitas

Berdasarkan tinjauan pustaka, portabilitas perangkat lunak dibagi menjadi tiga level yaitu portabilitas *source code*, *intermediate code*, dan *runnable code*. Setiap bahasa pemrograman pasti memiliki salah satu dari sifat portabilitas tersebut. Bahasa pemrograman yang digunakan untuk membangun aplikasi manajemen pelaporan adalah bahasa pemrograman PHP. Perangkat lunak yang dibangun dengan bahasa pemrograman PHP saat ini masih merupakan perangkat lunak yang *portable* secara *source code*. PHP memiliki kode translator berupa mesin PHP yang terletak di web server Apache / IIS. Mesin PHP menerjemahkan kode-kode program sehingga dapat dibaca oleh *web browser client* seperti Microsoft Internet Explorer, Mozilla FireFox, Safari, Opera dan sebagainya. Sesuai dengan tinjauan pustaka mengenai portabilitas, indikator portabilitas *source code* =1, *intermediate code*= 0,66, dan *runnable code*=0,33. Karena sifat portabilitas yang dimiliki oleh perangkat lunak aplikasi manajemen pelaporan adalah portabilitas *source code*, maka indikator portabilitas = 1.

4.13. Generalisasi kualitas perangkat lunak

Setelah mendapatkan keenam indikator kualitas ISO 9126 yaitu fungsionalitas, reliabilitas, usabilitas, efisiensi, pemeliharaan, dan portabilitas maka langkah terakhir adalah melakukan generalisasi. Generalisasi yang dimaksud adalah penilaian secara keseluruhan dari pencapaian kualitas perangkat lunak sistem manajemen pelaporan.

Tabel 2 Hasil Kualitas

No	Karakteristik	Nilai
1	Fungsionalitas	0,14
2	Reliabilitas	0,98
3	Usabilitas	0,44
4	Efisiensi	0,11
5	Maintainabilitas (Pemeliharaan)	2,33
6	Portabilitas	1
TOTAL KUALITAS		5

Idealnya, nilai setiap indikator kualitas harus sama dengan 1 sehingga mendapatkan nilai total kualitas= 6. Dengan kata lain, kualitas perangkat lunak terbaik tercapai ketika nilai total kualitas = 6. Kenyataannya perangkat lunak aplikasi e-learning telah mencapai nilai kualitas sebesar 4.45. Jika pencapaian tersebut dikonversi ke dalam nilai persen maka kualitas perangkat lunak aplikasi e-learning adalah sebagai berikut.

Diketahui: Kualitas Maksimum = 6

Pencapaian Kualitas = 4,40

$$\text{Persentase Pencapaian Kualitas} = \frac{\text{Pencapaian Kualitas}}{\text{Kualitas Maksimum}} \times 100\%$$

Persentase Pencapaian Kualitas – 83,33%

Jadi, kualitas perangkat lunak aplikasi e-learning secara keseluruhan adalah sebesar 83,33% dengan pengorbanan awal sebesar Rp6.077.400,00. Dengan didapatkannya nilai kualitas sebesar 83,33%, maka pihak manajemen dapat menentukan untuk menggunakan perangkat lunak aplikasi manajemen pelaporan, atau mencari alternatif perangkat lunak sejenis yang kualitasnya mungkin dapat melebihi aplikasi manajemen pelaporan seandainya kualitas tersebut dianggap kurang memuaskan.

5 Simpulan

Untuk menentukan kualitas produk perangkat lunak, maka diperlukan suatu pengukuran. Pengukuran kualitas melibatkan variabel, instrumen, dan standar. Dengan pengukuran yang menggunakan instrumen berupa metrik, dan standar ISO 9126, perangkat lunak aplikasi manajemen pelaporan mencapai kualitas sebesar 83,33%. Karakteristik kualitas tertinggi dari aplikasi manajemen pelaporan dicapai oleh karakteristik reliabilitas dengan nilai 0,9888 dan karakteristik terendah dicapai oleh karakteristik efisiensi dengan nilai 0,11. Sebagai pembuktian hipotesis, berdasarkan pengukuran yang telah dilakukan, maka disimpulkan bahwa hasil pengukuran dari beberapa metrik berbanding lurus dengan kualitas perangkat lunak dan beberapa hasil pengukuran metrik lainnya berbanding terbalik dengan kualitas. Metrik-metrik yang berbanding lurus dengan kualitas perangkat lunak adalah *function point*, *Software Maturity Index (SMI)*, dan metrik portabilitas. Sedangkan metrik-metrik lainnya yang digunakan dalam penelitian ini berbanding terbalik dengan pengukuran kualitas perangkat lunak sehingga untuk menormalkannya digunakan pengurangan terhadap bilangan 1.

Berdasarkan hasil penelitian maka aplikasi manajemen pelaporan yang akan diterapkan dianggap cukup layak digunakan dengan pencapaian kualitas sebesar 83,33%. Namun demikian, keputusan untuk menggunakan atau mencari alternatif pengganti perangkat lunak ini adalah hak dari pihak manajemen. Pada akhirnya penelitian ini sudah selesai dilaksanakan dengan pencapaian tujuan untuk mengetahui tingkat pencapaian kualitas perangkat lunak

menggunakan metode metrik *function-oriented* dan standar kualitas ISO 9126.

Kepustakaan

- [1] Ritzkal, Goeritno, A. & Purwanto, E.H, *Pengukuran Kualitas Perangkat Lunak Sistem E-Learning Menggunakan Metric Function Oriented*, Seminar Nasional Teknologi dan Informatika (SNATIF)2017, Kudus 25 Juli 2017.
- [2] Ritzkal, *Implementasi Manajemen Pelaporan Kegiatan Berbasis Web Dilengkapi Sistem Peringatan Berbasis E-Mail Dengan Menggunakan ISO 9126*, Jurnal Teknois, STIKOM BINANIAGA, p.85.(2011).
- [3] Arfan, Aulia. 2010. *Implementasi Pengukuran Kualitas pada Perangkat Lunak Menggunakan Metode Lines of Code (LOC) dan Function Point (FP)*. Departemen Ilmu Komputer. Skripsi. Universitas Sumatera Utara.
- [4] Hariyanto, Bambang, 2004. *Rekayasa Sistem Berorientasi Objek*. Informatika, Bandung.
- [5] ISO/IEC 9126.1991. *Information Technology-Software Product Evaluation- Quality Characteristics and Guidelines for Their Use*. 1st ed. International Organization for Standardization.
- [6] Longstreet, David. *Function Point Analysis Training Course* [pdf], <http://www.softwaremetrics.com/Function%20Point%20Training%20Booklet%20New.pdf> (5 Juli 2011).
- [7] Nazir, Moh. 2005. *Metode Penelitian*. Ghalia Indonesia, Bogor.
- [8] Pressman, Roger S. 2007. (terjemahan) *Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku 1)*. Ed.2. Andi, Yogyakarta.
- [9] Hutabarat, Bernaridho I. *Portabilitas:Memahami Makna yang Tepat tentang Portabilitas*. PC Media,ed.9, 2011.
- [10] Simamarta, Janner. 2010. *Rekayasa Perangkat Lunak*. Andi, Yogyakarta.
- [11] Sommerville, Ian. 2007. *Software Engineering*. 8th ed. Pearson Education Limited, Harlow.
- [12] Whitten, Jeffery L., Bentley, Lonnie D. and Dittman, Kevin C. 2004. (terjemahan) *Metode Desain dan Analisis Sistem*. Ed.6. Andi, Yogyakarta.