

Pengukuran Kompleksitas Sistem E-Learning di UIN Maulana Malik Ibrahim Malang Menggunakan Metrik Function Oriented

Fajar Rohman Hariri¹, Johan Ericka Wahyu Prakasa²

^{1,2}Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik

Ibrahim Malang

E-mail: *¹dosendes@gmail.com, ²johan@uin-malang.ac.id

Abstrak – Aspek kualitas perangkat lunak menjadi suatu hal yang penting dalam pembangunan proyek perangkat lunak. Pengukuran terhadap perangkat lunak guna mengestimasi kualitas produk dengan cara membandingkan volume sistem dengan banyaknya error/kesalahan yang ditemukan selama proses pengerjaan. Setiap pengukuran yang dilakukan dibutuhkan tersedianya ukuran kuantitatif yang disebut metrik. Metrik dapat dipisahkan dalam dua kategori, yaitu Metrik Size Oriented dan Metrik Function Oriented. Metrik Size Oriented dilakukan dengan menghitung LOC (Line Of Code) dimana metode ini kurang akurat arena pengukurannya didasarkan pada bahasa pemrograman dan gaya pengkodean programmer sehingga tidak dapat dipastikan bahwa dua program yang mempunyai LOC sama akan membutuhkan waktu implementasi yang sama walaupun keduanya diimplementasikan dengan kondisi pemrograman yang standard. Dalam penelitian ini pengukuran kualitas menggunakan Metrik Function Oriented karena metode ini merupakan pengukuran fungsionalitas yang ditujukan pada sistem pemrosesan data yang didominasi oleh operasi input dan output sehingga produktivitas pada bahasa pemrograman yang berbeda dapat dibandingkan. Metode melibatkan 5 komponen sistem software, diantaranya External Input, External Output, External Inquiry, Internal Logical File, External Interface File. Berdasarkan hasil tersebut, maka ditarik kesimpulan sesuai tujuan penelitian dengan pengujian menyatakan perangkat lunak e-learning di UIN Maliki Malang sangat baik.

Kata Kunci — e-learning, pengukuran, metric function oriented

1. PENDAHULUAN

Aspek kualitas perangkat lunak merupakan hal yang sangat penting dalam suatu pengembangan perangkat lunak. Kualitas perangkat lunak tidak hanya dilihat dari hasil produknya tetapi juga kualitas terhadap tahap pengembangan perangkat lunak itu sendiri. imana dalam menjamin kuitas perangkat lunak maka perlu dilakukan pengukuran terhadap perangkat lunak.

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].

Pengukuran merupakan dasar dari disiplin dalam perekayasa perangkat lunak. Melalui pengukuran, maka akan diperoleh tingkat pencapaian di dalam proyek perangkat lunak yang sedang diamati. Untuk setiap pengukuran yang dilakukan dibutuhkan tersedianya suatu ukuran kuantitatif yang disebut metrik. Metrik berdasarkan istilah rekayasa perangkat lunak didefinisikan sebagai sebuah ukuran kuantitatif yang dimiliki oleh suatu sistem, komponen atau proses tertentu dengan atribut-atribut yang diberikan.

Penelitian pengukuran kompleksitas perangkat lunak e-learning pernah dilakukan oleh Ritzkal pada tahun 2017. Pengukuran kualitas melibatkan variabel, instrumen, dan standar. Dengan pengukuran yang menggunakan instrumen berupa metrik, dan standar ISO 9126, perangkat lunak aplikasi elearning mencapai kualitas sebesar 74,19%. [1]

Septi Noer Laila pada tahun 2018 melakukan penelitian dengan judul Pengukuran Kualitas Perangkat Lunak Aplikasi Sisfo_Nilai di Politeknik Piksi Ganesha Berdasarkan Iso 9126. Kesimpulan yang didapat, perangkat lunak SISFO_Nilai telah memenuhi standar ISO 9126 dengan kriteria rata-rata “cukup”. [2]

Pada tahun 2017 Khairullah juga melakukan pengukuran kualitas perangkat lunak menggunakan metode McCall untuk mengukur kualitas Sistem Inventaris Aset. Dengan hasil penilaian cukup untuk factor *correctness, usability, integrity, reliability*. Serta baik untuk factor efisiensi.[3]

Penelitian kali ini akan membahas kualitas perangkat lunak berdasarkan pengukuran kompleksitas menggunakan metric function oriented pada system e-learning yang ada pada UIN Maulana Malik Ibrahim Malang.

2. METODE PENELITIAN

2.1 Jaminan Kualitas Perangkat Lunak (Software Quality Assurance)

Menurut Agarwal dkk Software Quality merupakan sebagai kesesuaian terhadap kebutuhan performa dan fungsionalitas, standar pengembangan yang terdokumentasi, serta karakteristik implisit dari sebuah perangkat lunak yang dikembangkan secara profesional. Secara umum, definisi Software Quality yang disebutkan oleh adalah *as effective software process applied in a manner that creates a useful product that provides measurable value for those who produce it and those who use it*. Proses dalam pembuatan sebuah barang dimana kita harus memastikan apakah barang tersebut sudah sesuai yang diharapkan atau belum, pengembangan perangkat lunak atau software juga menuntut hal yang sama.[4]

Pengujian adalah proses mengeksekusi program secara intensif untuk menemukan kesalahan-kesalahan. Pengujian tidak hanya untuk mendapatkan program yang benar, namun juga memastikan bahwa program tersebut bebas dari kesalahan-kesalahan untuk segala kondisi.[5] Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan spesifikasi, desain dan pengkodean.[6]

Ukuran merupakan faktor utama untuk menentukan biaya, penjadwalan, dan usaha. Kegagalan dari perkiraan ukuran yang tepat akan mengakibatkan penggunaan biaya yang berlebih atau keterlambatan penyelesaian proyek.

2.2 Metrik Perangkat Lunak

Menurut Sommerville (2003), pengukuran terbagi atas dua, yaitu pengukuran (metrik) kontrol dan pengukuran (metrik) prediktor. Metrik kontrol biasanya dihubungkan dengan

proses perangkat lunak, misalnya usaha dan waktu rata-rata yang dibutuhkan untuk memperbaiki cacat yang dilaporkan. Sedangkan metrik prediktor berhubungan dengan produk perangkat lunak, misalnya kompleksitas sikomatik modul, panjang rata-rata identifier pada program dan jumlah atribut dan operasi yang berhubungan dengan objek pada suatu rancangan

Metrik juga dapat dipisahkan dalam dua kategori, yaitu metrik langsung dan metrik tidak langsung. Metrik langsung dalam proses rekayasa perangkat lunak berhubungan dengan biaya dan sumber daya yang diperlukan, misalnya: pengukuran jumlah baris kode, kecepatan eksekusi, ukuran memori, dan kesalahan yang ditemui dalam suatu periode waktu. Metrik tidak langsung dari suatu produk berhubungan dengan fungsionalitas, kualitas, kompleksitas, efisiensi, reliabilitas, dan lain sebagainya. Pengukuran secara langsung lebih mudah dilakukan, karena hasil dapat diperoleh secara langsung, sedangkan pengukuran tidak langsung lebih sulit dilakukan, karena harus melalui proses yang lebih kompleks.

Metrik dipisahkan menjadi metrik proses, proyek, dan produk. Metrik produk bersifat privat untuk individu dan sering dikombinasikan untuk membuat metrik proyek yang bersifat publik bagi tim pengembang. Metrik proyek kemudian dikonsolidasikan untuk membuat metrik proses yang publik untuk seluruh organisasi atau perusahaan. Kesulitan yang biasanya dihadapi adalah pada saat melakukan kombinasi pada metrik-metrik yang diukur disebabkan karena sering terjadi perbedaan metrik antara individu satu dengan individu lainnya. Masalah tersebut biasa diatasi apabila dilakukan normalisasi pada proses pengukuran. Dengan adanya normalisasi maka dapat dilakukan perbandingan metrik pada cakupan yang lebih luas.[7]

2.3 Metric Function Oriented

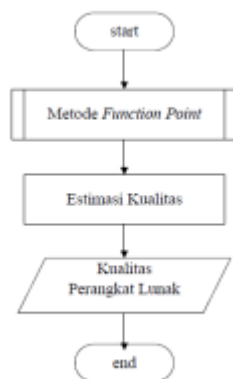
Pengukuran yang berhubungan dengan fungsi (Metric Function Oriented) adalah pengukuran fungsionalitas yang disampaikan oleh aplikasi sebagai suatu nilai normalisasi. Normalisasi dilakukan pada fungsionalitas pada aplikasi, tetapi untuk melakukan hal ini, fungsionalitas harus diukur dengan pengukuran langsung yang lain karena fungsionalitas tidak dapat diukur secara langsung. Maka

pengukuran dapat dilakukan dengan pengukuran function point.

Adapun parameter yang digunakan untuk pengukuran Function Point adalah [8] :

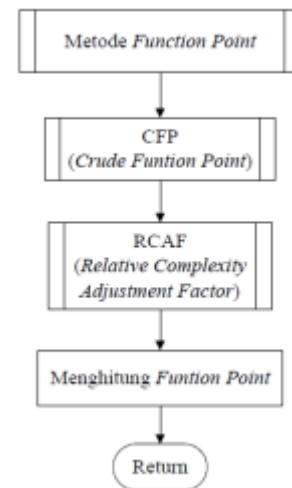
1. External Input (EI). Setiap masukan eksternal berasal dari pengguna atau dikirim dari aplikasi lain dan menyediakan data berorientasi aplikasi yang berbeda atau informasi pengendalian. Masukan sering digunakan untuk memperbarui internal logic file (ILF).
2. External Output (EO). Setiap keluaran eksternal diturunkan dari data aplikasi yang memberikan informasi kepada pengguna. External Output (EI) mengacu pada laporan, layar, pesan kesalahan, dan sebagainya.
3. External Inquiry (EQ). Kombinasi input/output yang dihasilkan oleh input dalam bentuk output sederhana dan singkat. Hasil inquiry misalnya berupa hasil pencarian data.
4. Internal Logical File (ILF). Merupakan data user atau kontrol informasi yang dikendalikan total oleh aplikasi. File logical dapat berupa file flat atau tabel tunggal dalam database relasional.
5. External Interface Files (EIF). File yang dikendalikan oleh aplikasi lain tetapi diperlukan oleh aplikasi. External interface file dapat berupa sekelompok data logical atau kontrol informasi yang keluar dan masuk ke aplikasi.

Alur perancangan sistem estimasi kualitas perangkat lunak yaitu dengan mengukur kompleksitas perangkat lunak menggunakan metode function point kemudian mengestimasi kualitas software dengan membandingkan error (kesalahan) dengan ukuran software yang ditunjukkan pada Gambar 1



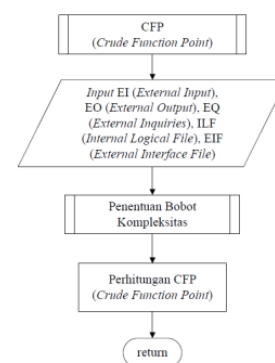
Gambar 1. Alur Pengukuran

Perhitungan metode function point yaitu dengan menghitung Crude Function Points (CFP), kemudian menghitung Relative Complexity Adjustment Factor (RCAF), dan selanjutnya adalah menghitung Funtion Point, dimana perhitungan metode function point ditunjukkan pada Gambar 2.



Gambar 2. Alur Perhitungan Metode Function Point

Langkah pertama dalam menghitung CFP (Crude Function Points) adalah dengan mencari jumlah dari komponen fungsional sistem pertama kali diidentifikasi dan dilanjutkan dengan mengevaluasi kuantitasi bobot kerumitan dari tiap komponen tersebut. Pembobotan tersebut kemudian dijumlahkan dan menjadi angka CFP. Alur perhitungan CFP ditunjukkan pada Gambar 3.



Gambar 3. Alur perhitungan CFP

Menghitung Function Point (FP) Setelah setiap karakteristik diberi bobot masing-masing dan dijumlahkan, maka langkah selanjutnya adalah proses melakukan perhitungan untuk mendapat nilai Function point (FP) dari software yang dibangun. Untuk menghitung Function Point menggunakan rumus pada persamaan sebagai berikut :

$$FP = CFP \times (0.65 + 0.01 \times RCAF) \dots\dots (1)$$

3.

4. HASIL DAN PEMBAHASAN

Berdasarkan hasil pengamatan, maka didapatkanlah data-data seperti yang diperlihatkan oleh Tabel 1 berikut ini.

Tabel 1. Fungsionalitas sistem

NAMA DATA	ITEM	JUMLAH
Inputan pengguna	Link ke halaman Login	5
	Bahasa	
	Username & Password	
	Tombol login	
	Tombol login sebagai tamu	
Output Pengguna	Form login	11
	Pilihan matakuliah	
	Topik perkuliahan	
	Materi perkuliahan berupa teks	
	Materi perkuliahan berupa gambar diam (still image)	
	Materi perkuliahan berupa gambar bergerak (video)	
	Materi perkuliahan berupa file (PDF / PPT)	
	Materi perkuliahan berupa link ke website lain	
	Forum diskusi	
	Soal tugas	
	Soal ujian (quiz)	
Permintaan pengguna	Menu Bahasa	12
	Menu kelas perkuliahan	
	Menu Pengguna	
	Tombol profile	
	Tombol grades	
	Tombol messages	
	Tombol preferences	
	Tombol Log Out	
	Tombol upload tugas	
	Tombol mulai ujian	
	Tombol memilih soal ujian	
Tombol selesai ujian		
File	File materi perkuliahan	3
	File jawaban tugas	
	File jawaban ujian	
Interface eksternal	External database authentication	3
	External database enrolment	
	Email	
Modul program saat ini	Modul user	8
	Modul activities	
	Modul courses	
	Modul User Group	
	Modul soal ujian	
	Modul quiz	

NAMA DATA	ITEM	JUMLAH
	Modul tugas	
	Modul message	
Bahasa pemrograman	PHP 7.2	1
Basis data	MySQL 5.2	1
Web Server	Apache2	1
Sistem Operasi Sever	Linux Ubuntu Server 18.04	1
Kegagalan	Matakuliah PKBBA Perubahan Password Siakad	2

3.1 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.

Tabel 2. Indicator fungsionalitas

Parameter	Jml	Faktor Pembobotan			Tot
		Sederhana	Rata	Komplek	
Input	5	3			15
Output	11	4			44
Permintaan	12		5		60
File	3		10		30
Interface External	3	6			18
				TOTAL	167

3.2 Faktor peubah 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.

Pada setiap karakteristik tersebut diberi bobot dari nilai 0 sampai 5 dengan asumsi nilai sebagai berikut:

0. Tidak berpengaruh
1. Insidental
2. Moderat
3. Rata-rata
4. Signifikan
5. Essential

Perhitungan faktor peubah kompleksitas function point adalah sebagai berikut:

Tabel 3. Faktor perubah kompleksitas

Faktor Peubah	Nilai
Backup dan recovery	1
Komunikasi data	5
Distribusi pemrosesan data	2
Performa	2
Konfigurasi operasional	2
Inputan onlie (on-line entry)	2
Perubahan data on-line	2
Tingkat transaksi data	2
Efisiensi pengguna	3
Kompleksitas pemrosesan	3
Penggunaan kembali (reusability)	3
Konversi dan Instalasi	2
Penggandaan instalasi	1
Fasilitas perubahan	3
TOTAL	30

3.3 Function Point

Diketahui : Jumlah Total = 382

$$\sum Fi = 30$$

Rumus untuk mencari function point adalah sebagai berikut.

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

$$FP = 167 \times (0,65 + (0,01 \times 30))$$

$$FP = 108,85$$

Dengan keterangan:

FP = Function Point

Jumlah Total = Nilai Total domain informasi

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

Setelah mendapatkan nilai FP, maka dapat digunakan sebagai acuan untuk mengestimasi kualitas perangkat lunak dengan cara membandingkan nilai FP dengan banyak error (kesalahan) yang terdapat pada perangkat lunak yang dibangun. Jika dalam pembangunan perangkat lunak tersebut ditemukan kesalahan sebanyak 2 kesalahan, maka dapat dihitung kesalahan per FP-nya adalah sebagai berikut :

$$\text{Kesalahan / FP} = 2 / 108,85$$

$$= 0,018$$

$$\text{Kualitas} = 100\% - (0,018 \times 100\%)$$

$$= 98,2$$

Berdasarkan nilai prosentase kualitas perangkat lunak adalah 98,2% maka perangkat lunak tersebut memiliki tingkatan sangat baik.

Tabel 4. Interval Tingkat Kualitas Pengembangan Perangkat Lunak

Interval Prosentase	Tingkat Kualitas
0-100 %	Sangat Baik
60-79.99%	Cukup Baik
0-59.99%	Cukup

5. SIMPULAN

Dari hasil pengamatan mulai tahap analisis, perancangan, implementasi dan uji coba, penulis mengambil kesimpulan sebagai berikut :

1. Metrik function oriented dapat dijadikan salah satu alternatif untuk mengestimasi kualitas perangkat lunak dengan mengukur kompleksitasnya.
2. Hasil pengujian yang dilakukan oleh sistem menggunakan metrik function oriented prosentase kualitas perangkat lunak adalah 98,2% maka perangkat lunak tersebut memiliki tingkatan sangat baik.

6. SARAN

Berdasarkan hasil penelitian dapat diberikan saran atau rekomendasi sebagai berikut:

1. Universitas Islam Negeri Maulana Malik Ibrahim Malang perlu melakukan penilaian terhadap system perangkat lunak yang lain untuk evaluasi perbaikan ke depan.
2. Perlu dilakukan pengukuran perangkat lunak menggunakan metode lain dengan standard ISO maupun standard yang lain.
3. Ditemukan 2 kegagalan dalam Perangkat Lunak e-Learning UIN Maulana Malik Ibrahim Malang, perlu adanya pembenahan terhadap 2 kegagalan yang ditemukan.

DAFTAR PUSTAKA

- [1] Ritzkal, Moh. Subchan. 2017. *Pengukuran Kualitas Perangkat Lunak Sistem E-Learning Menggunakan Metric Function Oriented*. Prosiding SNATIF . ISBN: 978-602-1180-50-1
- [2] Lailela, Septi Noer, R., 2018. *Pengukuran Kualitas Perangkat Lunak Aplikasi Sisfo_Nilai Di Politeknik Piksi Ganeshha Berdasarkan Iso 9126*. Seminar Nasional Telekomunikasi dan Informatika.
- [3] Khairullah. 2017. *Pengukuran Kualitas Sistem Informasi Inventaris Aset Universitas Muhammadiyah Bengkulu Menggunakan Metode Mccall*. Jurnal Informasi Interaktif Vol. 2 No. 2.

- [4] Agarwal, B. B., Tayal, S. P., & Gupta, M. (2010). *Software Engineering and Testing*. Sudbury: Jones and Bartlett Publishers.
- [5] Andri Kristanto, 2003, Perancangan Sistem Informasi dan Aplikasinya, Penerbit :Gava. Media, Jakarta.
- [6] Pressman, Roger, S, 1997, Rekayasa Perangkat Lunak : Pendekatan Praktisi. (Edisi Satu), Penerbit : Andi, Yogyakarta.
- [7] Sommerville, Ian. 2003. *Software Engineering (Rekayasa Perangkat Lunak)*. Erlangga, Jakarta.
- [8] Fathoni. 2009. Pengukuran Kualitas Perangkat Lunak Berdasarkan Kompleksitas Menggunakan Metode Function Point. *Jurnal Sistem Informasi (JSI)*. *Journal of Information Technology*. vol 18. hal 152-159.