

ESTIMASI KUALITAS PERANGKAT LUNAK BERDASARKAN PENGUKURAN KOMPLEKSITAS MENGGUNAKAN METRIK *FUNCTION ORIENTED*

Rinci Kembang Hapsari¹ dan M Jauhari Husen WP²

Teknik Informatika-ITATS, Jl. Arief Rahman Hakim 100 Surabaya

Email¹: rincikembang@itats.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 karena 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* Hasil pengujian metrik *function oriented*. Dimana estimasi kualitas perangkat lunak yang dilakukan oleh sistem dengan pakar menunjukkan tingkat keakurasian sebesar 75%.

Kata kunci : perangkat lunak, kualitas, estimasi, metrik, *function oriented*

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. Dimana dalam menjamin kualitas perangkat lunak maka perlu dilakukan pengukuran terhadap perangkat lunak.

Pengukuran merupakan dasar dari disiplin dalam perekayasaan 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.

Sehingga dalam penelitian kali ini akan membahas estimasi kualitas perangkat lunak berdasarkan pengukuran kompleksitas menggunakan metrik *function oriented*.

KAJIAN PUSTAKA

Triyanto (2011) menerangkan bahwa pengujian kualitas menggunakan Metode McCall adalah dengan menentukan faktor penilaian kualitas yang meliputi, faktor efisiensi (*efficiency*), kehandalan (*reliability*), kegunaan (*usability*), pemeliharaan (*maintainability*) dan media pembelajaran (*Compatibilitas*). Kemudian dari beberapa faktor yang telah ditentukan, dapat dikembangkan menjadi beberapa kriteria yang akhirnya dapat dihasilkan daftar pertanyaan yang sesuai dengan kriteria tersebut atau yang biasa disebut kuesioner. Penilaian dalam kuesioner ditentukan dengan skala 1 – 10. Kuesioner selanjutnya dibagikan dan diisi oleh responden, sehingga akan diperoleh penilaian kualitas dari sistem ini.

Pengukuran kualitas perangkat lunak dapat dilakukan dengan menggunakan salah satu dari berbagai model yang ada. Pada penggunaan model Star dan BNN, kekurangan yang ada sama yaitu kurangnya kriteria yang digunakan. Model ISO merupakan standar internasional yang ada saat ini, hanya saja model ini bersifat umum. Akan tetapi model ini memiliki kriteria evaluasi dan memisahkan kualitas eksternal dan internal yang ada sehingga model ini cocok digunakan pada saat kebutuhan penilaian kualitas perangkat lunak tidak terlalu dalam. Model McCall dan Star memenuhi banyak komponen penilaian, akan tetapi ada penilaian yang tumpang tindih karena banyaknya komponen yang dinilai. Model IEEE hampir sama dengan model ISO, hanya saja pada model IEEE menggunakan metrik perangkat lunak pada tingkat penilaian terakhir. Salah satu keunggulan model Boehm adalah dilibatkannya faktor perangkat keras. Hanya saja pada model ini kriteria yang digunakan masih sedikit. Model FURPS sendiri tidak memperdulikan *portability*, akan tetapi model ini memisahkan kebutuhan fungsional dan non fungsional. Pada model Kazman tingkat terakhir penilaian adalah menggunakan beberapa skenario untuk pengujian karakteristik. Kualitas perangkat lunak diukur menggunakan beberapa faktor, atribut dan juga karakteristik [2].

Akbar Alif menerangkan bahwa penelitiannya mengenai *systematic mapping study* mempunyai tujuan untuk melakukan overview atau pembahasan riset area, mengidentifikasi jumlah riset area tersebut yang telah ada sampai saat ini, jenis riset yang telah dilakukan dan pertanyaan-pertanyaan lain yang telah dirumuskan di dalam RQ. Kegiatan mapping study terdiri dari tiga aktivitas, yaitu melakukan pencarian makalah yang telah terpublikasi dan relevan, mendefinisikan skema pengklasifikasian dan mapping tiap-tiap makalah ke dalam skema klasifikasi yang telah dibuat [1].

Kualitas Perangkat Lunak

Menurut Roger Pressman dalam bukunya Rekayasa Perangkat Lunak “Kualitas *software* / perangkat lunak didefinisikan sebagai konfirmasi terhadap kebutuhan fungsional dan kinerja yang dinyatakan secara eksplisit, standar perkembangan yang didokumentasikan secara eksplisit dan karakteristik implisit yang diharapkan bagi semua perangkat lunak yang dikembangkan secara profesional.”

Kualitas perangkat lunak adalah keberadaan karakteristik dari suatu produk yang dijabarkan dalam kebutuhannya, artinya kita harus melihat terlebih dahulu karakteristik-karakteristik apa yang berhubungan atau tidak dengan kebutuhan-kebutuhan yang diinginkan oleh pemakai. Karakteristik yang dimaksud yaitu *contra-productive characteristics* dan *neutral characteristic*. Sedangkan factor-faktor kualitas perangkat lunak menurut McCall terdiri dari [7] :

1. **Kebenaran** adalah tingkat dimana program memenuhi spesifikasinya dan memenuhi sasaran misi karyawan.

2. **Reliabilitas** adalah tingkat dimana sebuah program dapat diharapkan melakukan fungsi yang diharapkan dengan ketelitian yang diminta.
3. **Efisiensi** adalah jumlah sumber daya penghitungan kode yang diperlukan oleh program untuk melakukan fungsinya.
4. **Integritas** adalah tingkat dimana akses ke perangkat lunak atau data oleh orang yang tidak berhak dapat di kontrol.
5. **Usabilitas** adalah usaha yang dibutuhkan untuk mempelajari, mengoperasikan, menyiapkan input, dan menginterpretasikan output suatu program.
6. **Maintanabilitas** adalah usaha yang diperlukan untuk mencari dan membetulkan kesalahan pada sebuah program.
7. **Flexibilitas** adalah usaha yang diperlukan untuk memodifikasi program operasional.
8. **Testabilitas** adalah usaha yang diperlukan untuk menguji sebuah program untuk memastikan apakah program melakukan fungsi – fungsi yang dimaksudkan.
9. **Portabilitas** adalah usaha yang diperlukan untuk memindahkan program dari satu perangkat keras dan atau lingkungan.
10. **Reusabilitas** adalah tingkat dimana sebuah program (bagian dari suatu program) dapat digunakan kembali di dalam aplikasi lain.
11. **Interperabilitas** adalah usaha yang diperlukan untuk merangkai satu sistem dengan yang lainnya.

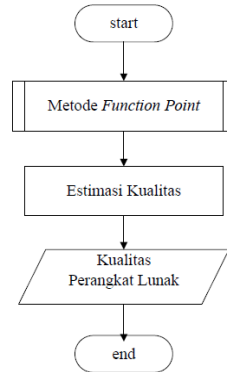
METODE

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:

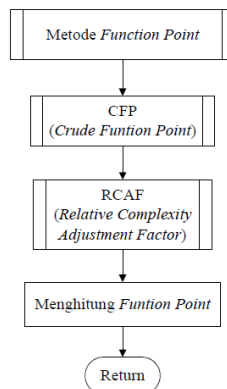
1. **External Input (EI)**. Setiap masukan eksternal berasal dari pengguna atau dikirim dari aplikasi lain dan menyediakan data berorientasi aplikasi yang berbeda atau informasi pengendalian. Masukan sering digunakan untuk memperbarui *internal logic file* (ILF).
2. **External Output (EO)**. Setiap keluaran 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 Perencanaan Sidang

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



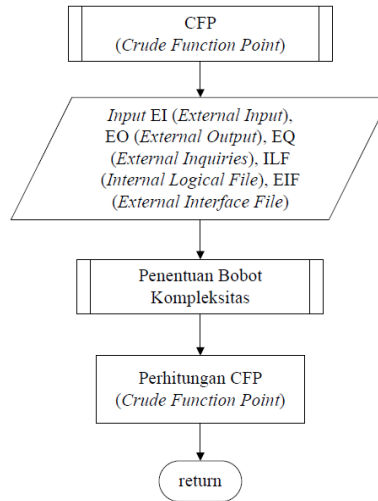
Gambar 2. Alur Perhitungan Metode *Function Point*

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.

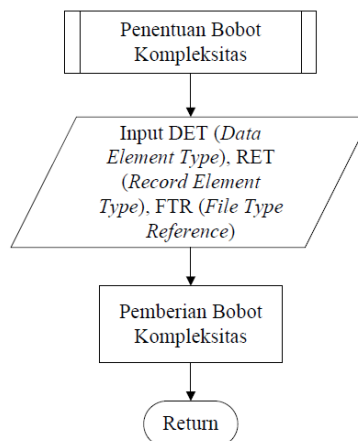
Sebelum dilakukan perhitungan CFP dilakukan penentuan bobot kompleksitas untuk mengetahui level kompleksitas pada masing-masing fungsi pengguna sehingga dapat digunakan untuk memberikan nilai bobot pada masing-masing komponen sistem, dimana alur penentuan bobot kompleksitas ditunjukkan pada Gambar 4. Penentuan bobot kompleksitas yang terdiri atas 3 input, yaitu :

1. *Data Element Type* (DET) yaitu *field* yang tidak berulang dan didefinisikan sebagai *field* unik dalam sebuah *internal logical file*.
2. *Record Element Type* (RET) yaitu sub kelompok data di dalam sebuah *logical file*.

3. *File Type Reference* (FTR) yaitu tipe file yang berupa *Internal Logical File* (ILF) atau *External Interface File* (EIF). Cara mudah untuk mengidentifikasi FTR dengan mengidentifikasi ILF dan EIF Karena jumlah keduanya sesalu sama



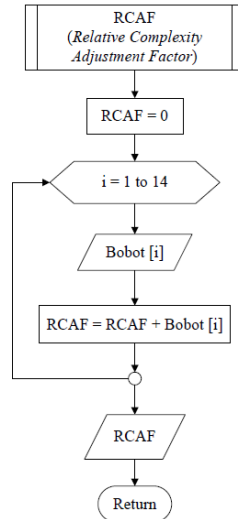
Gambar 3. Alur Perhitungan *Crude Function Points* (CFD)



Gambar 4. Alur Penentuan Bobot Kompleksitas

Langka kedua untuk menghitung *function point* adalah dengan menghitung RCAF (*Relative Complexity Adjustment Factor*), yang dihitung berdasarkan pada keseluruhan kompleksitas sistem. Cara menghitung RCAF (*Relative Complexity Adjustment Factor*) adalah dengan menggunakan 14 (empat belas) GSC (*General System Characteristic*), dimana masing-masing GSC berskala 0 (nol) sampai 5 (lima) . Skala 0 (nol) menunjukkan tidak adanya pengaruh dan skala 5 (lima) menunjukkan adanya pengaruh yang luas terhadap keseluruhan proyek.

Pada Gambar 5 merupakan perhitungan RCAF yang berfungsi untuk menghitung kesimpulan kompleksitas yang didalamnya terdapat 14 point karakteristik dari sistem *software*. Penilaian berskala 0 sampai 5 diberikan pada tiap karakteristik yang paling berpengaruh terhadap usaha pengembangan yang dibutuhkan.



Gambar 5. Alur Perhitungan *Relative Complexity Adjxtment Factor* (RCAF)

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 :

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

Misalkan diketahui nilai $CFP = 74$ dan $RCAF = 60$. Maka nilai FP dapat dicari sebagai berikut :

$$\begin{aligned}
 FP &= 74 \times (0.65 + 0.01 \times 60) \\
 &= 74 \times 1.25 = 92.5
 \end{aligned}$$

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 35 kesalahan, maka dapat dihitung kesalahan per FP-nya adalah sebagai berikut :

$$\begin{aligned}
 \text{Kesalahan} / FP &= 35 / 92.5 \\
 &= 0.3783
 \end{aligned}$$

$$\begin{aligned}
 \text{Kualitas} &= 100\% - (0.3783 \times 100\%) \\
 &= 62.17\%
 \end{aligned}$$

Berdasarkan nilai prosentase kualitas perangkat lunak adalah 62.17% maka perangkat lunak tersebut memiliki tingkatan **cukup baik**.

Tabel 1. Interval Tingkat Kualitas Pengembangan Perangkat Lunak

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

HASIL DAN PEMBAHASAN

Pada perhitungan *Crude Function Point* (CFP) terdapat beberapa *field* fungsi pengguna diantaranya adalah *field* jumlah *External Input* (EI), *External output* (EO), *External Inquiry* (EQ), *Internal Logical File* (ILF), dan *field* jumlah *External Interface File* (EIF).

Crude Function Point (CFP)

ID Software: PL1, Nama Software: Web Sistem Informasi Apotek Perak, ID Pemesan: 081501, Nama: Kusnadi, Tgl Kontrak: 05-11-2011, Lama Kontrak: 4 Bulan, Tgl Akhir Kontrak: 05-03-2012, Rp 2.500.000,00

Fungsi Pengguna	Jumlah	Level Kompleksitas	Jumlah CFP
External Input (EI)	13	Low: 5, Average: 10, High: 6	78
External Output (EO)	20		140
External Inquiry (EQ)	6		36
Internal Logical File (ILF)	8		80
External Interface File (EIF)	0		0
Total			334

Level Kompleksitas

Fungsi Pengguna	DET	RET	FTR	Level
External Input (EI)	10		8	HIGH
External Output (EO)	15		8	HIGH
External Inquiry (EQ)	6		8	HIGH
Internal Logical File (ILF)	8	0		AVERAGE
External Interface File (EIF)	0	0		LOW

Gambar 6(a). *Form Crude Function Point (CFP)* (b). *Form Level Kompleksitas*

Form Gambar 6(b) merupakan form digunakan untuk menentukan level kompleksitas perangkat lunak termasuk low, average, atau high. Untuk menentukan level kompleksitas perangkat lunak adalah dengan mengisi *field* jumlah *Data Element Type* (DET) dan *File Type Reference* (FTR) untuk EI, EO, dan EQ, dan *field* jumlah *Data Element Type* (DET) dan *Record Element Type* (RET) untuk ILF dan EIF.

Relative Complexity Adjustment Factor (RCAF)

ID Software: PL1, Nama Software: Web Sistem Informasi Apotek Perak, Masukkan Jumlah Oponen: 3, Total RCAF: 42

Hitung Function Point

Function Point : $CFP \times (0,65 + 0,01 \times RCAF)$
 : $334 \times (0,65 + 0,01 \times 42)$
 Nilai FP : 357,38

Estimasi Perangkat Lunak

ID Software: PL1, Nama Software: Web Sistem Informasi Apotek Perak

Estimasi Kualitas

Jumlah Error: 53, Persentase Kualitas: 85.17%, Sangat Baik

Gambar 7(a). *Form Relative Complexity Adjustment Factor(RCAF)*
 7(b). *Form Perhitungan Estimasi Perangkat Lunak*

Pengujian Sistem Dengan Pakar

Tahap ini merupakan tahap pengujian antara hasil estimasi yang dilakukan oleh sistem dengan hasil estimasi yang dilakukan oleh pakar. Data pakar diperoleh dengan melakukan survei pada perusahaan-perusahaan yang bergerak dibidang pengembangan perangkat lunak.

Tabel 2. Data pengujian Sistem dan Pakar

No	Nama Perangkat Lunak	Hasil Estimasi		Keterangan
		Pakar	Sistem	
1	Web Sistem Informasi Apotek Perak	Sangat Baik	Sangat Baik	Sesuai
2	Web Sistem Informasi Butik Sahara	Sangat Baik	Sangat Baik	Sesuai
3	Web Sistem Informasi Butik Maharani	Sangat Baik	Cukup Baik	Tidak Sesuai
4	Web Sistem Informasi Butik Salma Collection	Sangat Baik	Sangat Baik	Sesuai
5	Web Sistem Informasi Apotek Kapasan	Sangat Baik	Sangat Baik	Sesuai
6	Web Online Shop	Sangat Baik	Cukup Baik	Tidak Sesuai
7	Web Hijab Online Shop	Sangat Baik	Sangat Baik	Sesuai
8	Web Sistem Informasi Apotek	Cukup Baik	Cukup Baik	Sesuai
9	Web Sistem Informasi JMCell	Sangat Baik	Sangat Baik	Sesuai
10	Web Sistem Informasi Butik Riayaba	Cukup Baik	Cukup Baik	Sesuai
11	Sistem Informasi Hotel Alisha	Sangat Baik	Sangat Baik	Sesuai
12	Sistem Informasi Jogja HandPhone	Cukup Baik	Sangat Baik	Tidak Sesuai
13	Aplikasi Toko Putri Diana	Cukup Baik	Cukup Baik	Sesuai
14	Sistem Informasi Graha Komputer	Sangat Baik	Sangat Baik	Sesuai
15	Sistem Informasi Habib Koko	Cukup Baik	Sangat Baik	Tidak Sesuai
16	Hotel Olympic	Sangat Baik	Sangat Baik	Sesuai
17	Sportivo	Sangat Baik	Sangat Baik	Sesuai
18	All Season	Sangat Baik	Cukup Baik	
19	My Property	Sangat Baik	Sangat Baik	Sesuai
20	Visicom	Sangat Baik	Sangat Baik	Sesuai

Hasil perbandingan estimasi yang dilakukan oleh sistem dengan pakar ditunjukkan oleh tabel 2. Berdasarkan tabel 1 didapat tingkat akurasinya adalah $\frac{15}{20} \times 100\% = 75\%$

KESIMPULAN

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* dengan untuk estimasi kualitas perangkat lunak menunjukkan tingkat keakurasian sebesar 75 %.

DAFTAR PUSTAKA

- [1]. Akbar, Alif; Christian dan Laili, Umi. 2015. Pengukuran Kualitas Perangkat Lunak Berdasarkan ISO/IEC 25000: Systematic Mapping. *Jurnal Manajemen Informatika*. Vol 04 (01).
- [2]. Wayan, Gede dan Arida, Luh. 2012. Komponen Penilaian Kualitas Perangkat Lunak Berdasarkan Software Quality Model. *Seminar Teknologi Informasi & Komunikasi Terapan*. ISBN 978-26-0255-0. Semarang.
- [3]. Daniari, Imania. 2013. Perkiraan Biaya Pembuatan Enterprise Resource Planning (ERP) Untuk Unit Bisnis Gula Pada PT. Perkebunan XYZ Dengan Metode Function Point. *Jurnal Teknik POMITS*. Vol 1 (1).
- [4]. Fathoni. 2009. Pengukuran Kualitas Perangkat Lunak Berdasarkan Kompleksitas Menggunakan Menggunakan Metode Function Point. *Jurnal Sistem Informasi (JSI)*. Vol 1 (2).
- [5]. Hariyanto, Bambang. 2011. *Eseni-Esendi Bahasa Pemrograman Java*. Informatika Bandung. Indonesia.
- [6]. Kristanto, Andri. 2004. *Rekayasa Perangkat Lunak Konsep Dasar*. Gava Media. Yogyakarta.
- [7]. Pressman, S Roger.. 2001. *Software Engineering: A Practitioner's Approach*. McGraw Hill. New York.
- [8]. Simarmata, Janner. 2010. *Rekayasa Perangkat Lunak*. Andi Offset. Yogyakarta

Halaman ini sengaja dikosongkan