

PENGUJIAN SISTEM INFORMASI AKADEMIK MENGGUNAKAN MCCALL'S SOFTWARE QUALITY FRAMEWORK

Agus Mulyanto

Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga
JI Marda Adisucipto Yogyakarta 55281
e-mail : agus.mulyanto@uin-suka.ac.id

Abstract

Software testing is a critical element for determining the software quality. It covers design, specification, and coding. This study aims to test software quality of academic information systems at State Islamic University Sunan Kalijaga. It uses McCall's software quality framework on product operation aspects. Five metrics tested were correctness, reliability, efficiency, integrity and usability. The result showed that academic information system has a good quality with values above 70.

Keywords : Academic Information System, McCall, Software Testing

Abstrak

Pengujian perangkat lunak merupakan elemen kritis dalam menentukan kualitas suatu perangkat lunak. Pengujian ini meliputi desain, spesifikasi, dan koding. Penelitian ini bertujuan menguji kualitas perangkat lunak sistem informasi akademik di Universitas Islam Negeri Sunan Kalijaga. Pengujian ini menggunakan kerangka kerja McCall pada aspek product operation. Lima metrik yang diuji adalah correctness, reliability, efficiency, integrity, dan usability. Hasil pengujian menunjukkan bahwa sistem informasi akademik ini memiliki kualitas yang baik dengan nilai di atas 70.

Kata kunci : Sistem Informasi Akademik, McCall, Pengujian Perangkat Lunak.

1. PENDAHULUAN

1.1. Latar Belakang

Kualitas perangkat lunak diperlukan dalam pengembangan sistem. Kualitas akan berpengaruh terhadap performa perangkat lunak tersebut. Kualitas perangkat lunak merupakan pemenuhan kebutuhan fungsionalitas dan kinerja yang didokumentasikan. Agar perangkat lunak memiliki performa yang baik maka dibutuhkan penggalan kebutuhan pengguna secara tepat. Sebuah produk memiliki kualitas yang baik apabila dapat memuaskan sebagian besar penggunanya. Menurut (Pressman, 2002) jaminan kualitas perangkat lunak merupakan hal yang penting karena akan memberi informasi bagi manajemen untuk mengetahui kualitas produk. Produk berkualitas dapat memberi kepastian dan kefidensi bahwa produk dapat memenuhi sasaran mutu.

Pengujian perangkat lunak merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean. Pengujian perangkat lunak adalah suatu proses yang digunakan untuk mengidentifikasi ketepatan, kelengkapan dan mutu dari perangkat lunak. Setiap pengujian membutuhkan tersedianya suatu ukuran kualitatif dengan strategi yang mengintegrasikan metode design test case perangkat lunak ke dalam sederetan langkah yang direncanakan dengan baik. Hasilnya adalah konstruksi perangkat lunak yang berhasil. Oleh karena itu harus ada patokan yang jelas mengenai kualitas perangkat lunak tersebut. Salah satu parameter dalam pengukuran kualitas suatu perangkat lunak adalah dengan menggunakan McCall's Software Quality Factor.

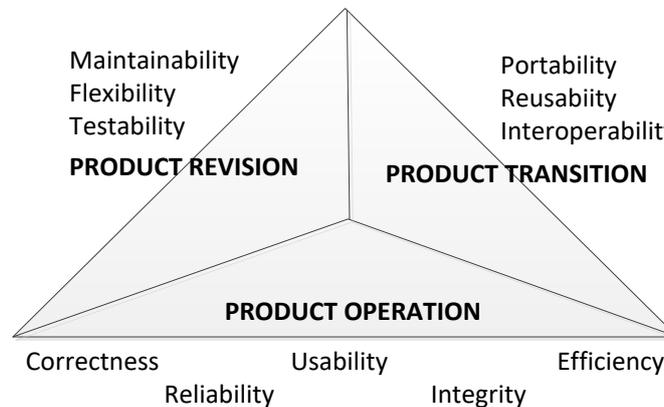
Sistem Informasi Akademik merupakan perangkat lunak aplikasi yang dipakai untuk membantu proses akademik di Perguruan Tinggi. Universitas Islam Negeri (UIN) Sunan Kalijaga sudah

mengimplementasikan Sistem Informasi Akademik untuk melayani kegiatan akademik civitas akademika. Sistem ini digunakan oleh tiga kelompok pengguna, yaitu dosen, mahasiswa dan staf akadmik sebagai administrator sistem.

Permasalahan yang terjadi adalah Sistem Informasi Akademik sebagai perangkat lunak aplikasi sudah lama diimplementasikan namun belum pernah diuji kehandalannya menggunakan metric yang standar. Oleh karena itu, penelitian ini bertujuan menganalisis Sistem Informasi Akademik di UIN Sunan Kalijaga menggunakan atribut McCall. Fokus penelitian ini pada aspek *product operation* yang terdiri dari lima metrik yaitu: *correctness*, *reliability*, *efficiency*, *integrity*, dan *usability*. Pengujian kelima metrik ini dibatasi untuk kelompok pengguna pada level dosen.

1.2. McCall's Software Quality Framework

McCall telah menemukan faktor-faktor yang mempengaruhi suatu kualitas perangkat lunak. *McCall's Software Quality Framework* merupakan suatu penggolongan kriteria yang mempengaruhi kualitas perangkat lunak. Framework ini menekankan kriteria tersebut menjadi tiga aspek penting yang saling berhubungan, sebagaimana disajikan dalam gambar 1. McCall dan kawan-kawan pada tahun 1977 telah mengusulkan suatu penggolongan faktor-faktor atau kriteria yang mempengaruhi kualitas perangkat lunak.



Gambar 1. McCall's Software Quality Framework (Pressman, 2002).

McCall menitikberatkan faktor-faktor tersebut menjadi tiga aspek penting, yaitu:

1. Sifat-sifat operasional dari perangkat lunak (*Product Operation*).

Sifat-sifat yang berhubungan dengan operasional perangkat lunak harus diperhatikan oleh para perancang dan pengembang dalam menciptakan sebuah aplikasi. Hal-hal yang diukur adalah yang berkaitan dengan teknik analisis, perancangan, dan konstruksi sebuah perangkat lunak. Metrik yang berhubungan dengan sifat-sifat operasional perangkat lunak adalah:

- a. *Correctness*: tingkat suatu perangkat lunak memenuhi spesifikasi dan mission objective dari pengguna. Pengukuran *correctness* ini dipengaruhi oleh 3 faktor kualitas, yaitu:
 - (1). *Completeness*, yaitu sejauh mana implementasi penuh dan fungsi-fungsi yang diperlukan telah tercapai.
 - (2). *Consistency*, yaitu derajat penggunaan teknik-teknik desain dan dokumentasi yang seragam pada seluruh proyek pengembangan.
 - (3). *Traceability*, yaitu kemudahan yang merujuk baik imlementasi atau komponen program kebutuhan pengguna software.

Nilai metric *correctness* ini dihitung menggunakan persamaan 1.

$$\text{Nilai correctness} = \frac{\text{completeness} + \text{consistency} + \text{traceability}}{3} \quad (1)$$

- b. *Reliability*: tingkat perangkat lunak yang dapat diharapkan menjalankan fungsinya dengan ketelitian yang dibutuhkan. Pengukuran *reliability* ditentukan oleh 5 faktor kualitas, yaitu:
- (1). Accuracy, pengujian yang dilakukan untuk menentukan atau nilai ketelitian pada komputasi program dan control.
 - (2). Error Tolerance, pengujian yang dilakukan untuk menentukan nilai toleransi terhadap kesalahan yang terjadi pada program.
 - (3). Consistency, merupakan derajat penggunaan teknik-teknik desain dan dokumentasi yang seragam pada seluruh proyek pengembangan.
 - (4). Modularity, pengujian ini dilakukan untuk menentukan nilai seberapa independen atau mandiri sebuah modul yang menjadi komponen sebuah software.
 - (5). Simplicity, pengujian kesederhanaan berpengaruh pada pengujian modularitas pengambilan kesederhanaan diambil dari nilai modularitas.
- Nilai metric *reliability* ini dihitung menggunakan persamaan 2.

$$\text{Nilai reliability} = \frac{\text{accuracy} + \text{error tolerance} + \text{consistency} + \text{modularity} + \text{simplicity}}{5} \quad (2)$$

- c. *Efficiency*: banyaknya sumber daya komputasi dan kode program yang diperlukan suatu perangkat lunak untuk melakukan fungsinya. Pengukuran *efficiency* ditentukan oleh 3 faktor kualitas, yaitu:
- (1). *Conciseness*, pengujian dengan cara menghitung keringkasan dan kepadatan program dalam jumlah baris kodenya.
 - (2). *Execution efficiency*, pengujian dengan menghitung nilai keefisiensi program dari kinerja run-timanya saat digunakan pada device. Penghitungan kinerja run-time meliputi memori yang digunakan saat sudah dipasang, sampai pengguna memori (RAM) saat program tersebut berjalan.
 - (3). *Operability*, pengujian dari kemudahan pengguna/pengoperasian program. Dalam tahap operability ini mempergunakan pernyataan-pernyataan yang diberikan pada pengguna program (kuisisioner).
- Nilai metrik *efficiency* ini dihitung menggunakan persamaan 3.

$$\text{Nilai efficiency} = \frac{\text{conciseness} + \text{execution efficiency} + \text{operability}}{3} \quad (3)$$

- d. *Integrity*: sejauh mana akses ke perangkat lunak dan data oleh pihak yang tidak berhak dapat dikendalikan. Pengukuran *integrity* ditentukan oleh 3 faktor kualitas, yaitu:
- (1). Auditability, penyesuaian/keselarasan terhadap standar yang dapat diperiksa. Auditabilitas merupakan pengujian software dengan cara melihat desain sistemnya.
 - (2). Instrumentation, tingkatan dimana pengawasan program memiliki operasi tersendiri dan mengidentifikasi kesalahan yang terjadi. Sehingga program dapat mengenal, mengidentifikasi dan menangani error yang terjadi.
 - (3). Security, merupakan ketersediaan mekanisme yang mengontrol atau melindungi program atau data.
- Nilai metric *integrity* ini dihitung menggunakan persamaan 4.

$$\text{Nilai integrity} = \frac{\text{auditability} + \text{instrumentation} + \text{security}}{3} \quad (4)$$

- e. *Usability*: usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan input, dan mengartikan output dari perangkat lunak. Pengukuran *usability* ditentukan oleh 2 faktor kualitas, yaitu:
- (1). Operability, kecocokan operasi pengguna system oleh user akhir. Penilaian diambil berdasarkan observasi melalui kuisisioner analisis, perhitungan stastistika.
 - (2). Training, tingkatan dimana software membantu menerapkan system oleh user yang baru.

2. Kemampuan perangkat lunak menjalani perubahan (*Product Revision*).
Suatu perangkat lunak yang dirancang dan dikembangkan dengan baik, akan dengan mudah direvisi apabila dibutuhkan. Tingkatan dimana perangkat lunak dapat diperbaiki adalah hal penting yang harus diperhatikan. Faktor-faktor McCall yang berhubungan dengan kemampuan perangkat lunak untuk menjalani perubahan adalah:
 - a. *Maintainability*: usaha yang dibutuhkan untuk menemukan dan memperbaiki kesalahan dalam perangkat lunak.
 - b. *Flexibility*: usaha yang dibutuhkan untuk melakukan modifikasi terhadap perangkat lunak yang operasional.
 - c. *Testability*: usaha yang dibutuhkan untuk menguji suatu perangkat lunak dengan memastikan apakah perangkat lunak melakukan atau tidak melakukan fungsi yang dikehendaki.
3. Penyesuaian perangkat lunak terhadap lingkungan baru (*Product Transition*).
Setelah integritas perangkat lunak telah diukur menggunakan faktor sistem operasional, faktor terakhir yang harus diperhatikan adalah faktor transisi, yaitu bagaimana perangkat lunak tersebut dapat dijalankan pada beberapa platform atau kerangka sistem yang beragam. Faktor-faktor McCall yang berhubungan dengan tingkat adaptabilitas perangkat lunak terhadap lingkungan baru adalah:
 - a. *Portability*: usaha yang dibutuhkan untuk mentransfer perangkat lunak dari sebuah hardware dan/atau sistem perangkat lunak tertentu supaya mampu melakukan fungsi pada hardware dan/atau sistem perangkat lunak yang lain.
 - b. *Reusability*: tingkat perangkat lunak dapat digunakan ulang pada aplikasi lainnya.
 - c. *Interoperability*: usaha yang dibutuhkan untuk menghubungkan suatu perangkat lunak dengan lainnya.

1.3. Studi Pustaka

Penelitian tentang pengujian perangkat lunak telah banyak dilakukan peneliti sebelumnya. Menurut Zahra (2012) faktor kualitas yang paling diprioritaskan pengguna adalah correctness kemudian usability. Sedangkan untuk hasil uji korelasi faktor kualitas McCall dengan indikatornya diperoleh hasil bahwa hanya faktor correctness dan usability yang mempunyai korelasi. Penelitian ini menggunakan metode kuantitatif, metode pengambilan data memakai survei dan instrumen penelitian berupa kuesioner. Penelitian ini sekaligus melakukan validasi terhadap faktor kualitas McCall aspek operasional terhadap indikator-indikatornya menggunakan uji korelasi Kendall Tau dan Spearman Rank. Penelitian ini juga diperkuat oleh Tresnasari (2012) dengan menggunakan metode software quality assurance (SQA).

Backhtiar et al (2013) mengidentifikasi komponen penjaminan kualitas perangkat lunak di Universitas Komputer Indonesia (UNIKOM). Selain itu untuk mengukur kesiapan UNIKOM membangun perangkat lunak yang berkualitas. Metode penelitian yang digunakan adalah penelitian analisis (analytical research). Faktor yang diuji adalah faktor kualitas McCall yang terdiri dari faktor *correctness*, *reliability*, *efficiency*, *integrity*, *usability*, *maintainability*, *flexibility*, *testability*, *portability*, *reusability*, *interoperability*. Sedangkan faktor kualitas alternatif terdiri dari *verifiability*, *expandability*, *safety*, *manageability*, dan *survivability*. Hasil penelitian ini menunjukkan bahwa belum ada perhatian khusus terhadap model kualitas perangkat lunak baik metode McCall maupun metode alternative lainnya.

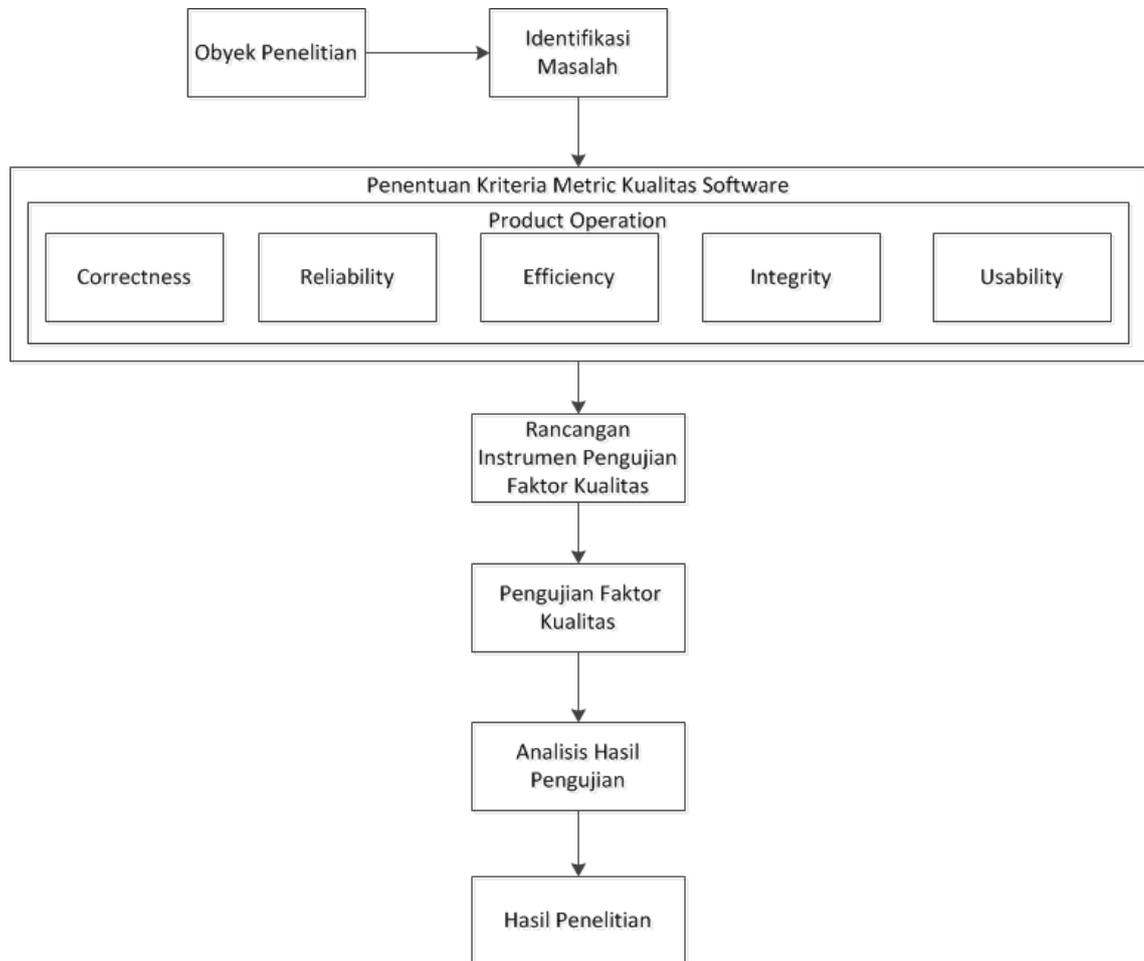
Sulaiman (2009) menggali berapa besar kualitas perangkat lunak pada perusahaan perangkat lunak Indonesia. Penelitian ini menerapkan dan mengaplikasikan penjaminan mutu perangkat lunak pada perusahaan perangkat lunak. Kegiatan penjaminan mutu pada aplikasi perangkat lunak sangat menentukan hasil aplikasi yang dibuat.

2. METODE PENELITIAN

2.1. Desain Penelitian

Penelitian ini bersifat deskriptif kualitatif yaitu penelitian yang mendeskripsikan suatu keadaan dengan kerangka kerja yang dibuat oleh McCall. Langkah-langkah penelitian deskriptif adalah

menentukan obyek penelitian, mengidentifikasi masalah, menentukan metrik yang akan diukur secara kuantitatif, merancang instrument pengujian, melakukan pengujian, menganalisis secara kuantitatif dan merumuskan hasil penelitian. Langkah-langkah penelitian ini disajikan dalam gambar 2.



Gambar 2. Langkah-langkah Penelitian.

2.2 Objek Penelitian

Objek penelitian ini adalah Sistem Informasi Akademik UIN Sunan Kalijaga Yogyakarta. Perangkat lunak ini diambil sebagai objek penelitian karena sistem ini merupakan sistem akademik terpusat yang digunakan di UIN Sunan Kalijaga dalam mengelola kegiatan akademiknya. Kelompok pengguna sistem ini yaitu administrator, dosen dan mahasiswa. Namun demikian, penelitian ini fokus pada kelompok pengguna dosen dan dilakukan pada tahun akademik 2015/2016. Kelompok pengguna dosen memiliki tujuh menu utama yaitu: Perkuliahan, Evaluasi Pembelajaran, Kinerja Dosen, Kuliah Kerja Nyata, Kuliah Praktik, Tugas Akhir & Ujian Tugas Akhir, serta Penelitian & Pengabdian. Tampilan menu utama untuk pengguna dosen disajikan pada gambar 3.



Gambar 3. Tampilan Menu Sistem Informasi Akademik.

3. HASIL DAN PEMBAHASAN

Hasil pengujian kelima metrik kualitas software pada aplikasi sistem informasi akademik disajikan dalam bagian berikut.

3.1. Pengujian Correctness

Faktor *correctness* merupakan suatu metrik yang digunakan untuk mengetahui sejauh mana suatu software dapat memenuhi spesifikasi dan *mission objective* dari pengguna. *Correctness* memiliki tiga faktor kualitas yaitu *completeness*, *traceability* dan *consistency*.

Correctness merupakan metrik yang digunakan untuk mengetahui sejauh mana suatu software dapat memenuhi spesifikasi dan *mission objective* dari pengguna. *Correctness* terdiri dari 3 faktor kualitas yaitu *completeness*, *consistency* dan *traceability*.

Faktor *completeness* adalah ketercapaian implementasi fungsi-fungsi yang diperlukan secara penuh. Sistem informasi akademik untuk dosen terdapat 47 fitur yang diuji. Terdapat 44 fitur yang memiliki kesesuaian antara input dan outputnya, sedangkan 3 fitur tidak sesuai antara input dan outputnya. Dengan demikian dapatkan nilai *completeness* adalah 44/47 atau sebesar 93,62%.

Faktor *consistency* adalah derajat penggunaan teknik-teknik desain dan dokumentasi yang seragam pada seluruh proyek pengembangan software. Fitur dari sistem informasi akademik yang di uji sebanyak 47 fitur. Sebanyak 39 fitur memiliki kesesuaian antara dokumen dan sistem, sedangkan yang 8 fitur tidak sesuai antara dokumen dan sistemnya. Sehingga nilai *consistency* adalah 39/47 atau sebesar 82,98%.

Sedangkan faktor *traceability* adalah kemudahan merujuk balik implementasi atau komponen program kepada kebutuhan pengguna software. Fitur dari sistem informasi akademik yang di uji sebanyak 47 fitur. Terdapat 39 fitur yang memiliki kesesuaian antara sistem dan dokumen kebutuhannya, dan 8 fitur tidak sesuai antara sistem dan dokumen kebutuhannya. Sehingga nilai *consistency* adalah 39/47 atau sebesar 82,98%.

Berdasar persamaan 1, maka diperoleh nilai *correctness*:

$$\text{Nilai correctness} = \frac{93,62\% + 82,98\% + 82,98\%}{3}$$

$$\text{Nilai correctness} = \frac{259,58\%}{3} = 86,53\%$$

3.2. Pengujian Reliability

Reliability merupakan metrik yang digunakan untuk mengetahui sejauh mana suatu software dapat diharapkan untuk melaksanakan fungsinya dengan ketelitian yang diperlukan. Metrik *reliability* ditentukan oleh 5 faktor kualitas yaitu *accuracy*, *consistency*, *error tolerance*, *modularity*, dan *simplicity*.

Accuracy adalah ketelitian komputasi dan kontrol. Penelitian ini menguji 56 fitur dalam sistem informasi akademik. Sebanyak 48 fitur sudah sesuai antara input dan output dari komputasinya. Sedangkan 8 fitur tidak sesuai antara input dan outputnya. Hasil pengujian faktor *accuracy* adalah 48/56 atau sebesar 85,71 %.

Consistency adalah derajat penggunaan teknik-teknik desain dan dokumentasi yang seragam pada seluruh proyek pengembangan software. Pengujian *consistency* pada 47 aktivitas, ditemukan 8 aktivitas yang tidak sesuai dengan dokumen kebutuhan dan 39 aktivitas lainnya sudah sesuai dengan dokumen kebutuhan. Sehingga hasil pengujian faktor *consistency* adalah 39/47 atau sebesar 82,98 %

Error Tolerance adalah toleransi kesalahan yang terjadi apabila program mengalami gagal running. Pengujian ini dilakukan kepada 48 fitur, terdapat 44 fitur yang sesuai. Sehingga hasil pengujian faktor *error tolerance* adalah 44/48 atau sebesar 91,67 %.

Modularity adalah independensi fungsional dari komponen program. Pengujian *modularity* dilakukn kepada 30 fitur dan diperoleh 22 fitur yang independen. Sehingga hasil pengujian faktor modularity sebesar 22/30 atau sebesar 73,33.

Simplicity adalah kemudahan program untuk dimengerti tanpa kesulitan. Semakin independence suatu program maka semakin mudah dimengerti. Total fitur yang diuji sebanyak 30 fitur, terdapat 22 fitur independen. Hasil pengujian faktor simplicity adalah 22/30 atau sebesar 73,33 %.

Dengan demikian metrik *reliability* dapat dihitung menggunakan rumus 2 yaitu:

$$\text{Nilai reliability} = \frac{85,71\% + 91,67\% + 82,98\% + 73,33\% + 73,33\%}{5}$$

$$\text{Nilai reliability} = \frac{407,02\%}{5}$$

$$\text{Nilai reliability} = 81,40\%$$

3.3. Pengujian Efficiency

Efficiency merupakan metrik untuk mengetahui banyaknya sumber daya komputasi dan kode program yang dibutuhkan suatu software untuk melakukan fungsinya. Metrik *efficiency* ditentukan oleh 3 faktor kualitas yaitu *conciseness*, *execution efficiency* dan *operability*.

Conciseness adalah keringkasan program dalam ukuran *line of commands (LOC)*. Perhitungan *conciseness* adalah sama dengan rata-rata total classes x 100%. Penelitian ini menguji sebanyak 107 classes dari 4236 LOC. Nilai rata-rata classes dapat dihitung menggunakan persamaan:

$$\text{Classes X} = \frac{\sum \text{deklarasi classes X}}{\text{Total LOC}}$$

$$\text{Classes X} = \frac{107 \text{ classes}}{4236} = 0,025259$$

Diperoleh nilai standar deviasi sebesar 0,014875. Sehingga nilai concieness adalah 2,53% ± 1,49%.

Execution efficiency adalah kinerja run-time dari program. *Execution Efficiency* dapat digunakan untuk menentukan efisiensi suatu sitem yang memiliki hubungan dengan penggunaan memori saat sudah dipasang sampai penggunaan memori ketika sistem tersebut berjalan pada *device* yang dipakai. Pengujian yang dilakukan dalam penelitian menggunakan dua *browser* yaitu mozilla fiefox dan google chrome. Perhitungan *execution efficiency* adalah :

$$\text{Execution efficiency} = \frac{\text{Rata-rata memory (MB)}}{\text{Total RAM (MB)}} \times 100 \%$$

Hasil perhitungan *execution efficiency* untuk *mozilla firefox* sebesar 1,77% ± 0,32%, sedangkan untuk *google chrome* sebesar 0,63% ± 0,07%. Mengacu pada literatur sebelumnya, yaitu semakin kecil penggunaan memori RAM maka semakin efisien sistem untuk dijalankan. Berdasarkan perbandingan kedua *browser* tersebut, maka nilai yang paling efisien adalah *google chrome* sebesar 0,63%± 0,07%.

Operability adalah faktor kemudahan mengoperasikan program. Pengujian faktor operability untuk mengukur kemudahan pengguna mengoperasikan sistem informasi akademik. Alat ukur yang digunakan adalah kuisisioner yang diisi oleh pengguna. Pengguna sistem ini adalah dosen. Pengukuran menggunakan skala likert dengan 5 skala dan 10 pernyataan. Kelima skala diberi bobot dari 5 sampai dengan 1 secara berurutan dengan dinyatakan dalam “sangat setuju (SS)”, “setuju (S)”, “netral (SS)”, “tidak setuju (TS)” dan “sangat tidak setuju (STS)”. Hasil pengukuran operability disajikan dalam tabel 1.

Tabel 1. Hasil Pengukuran Operability.

NO	PERNYATAAN	SS	S	N	TS	STS	JML
1	Aplikasi akademik UIN Sunan Kalijaga mudah diakses dari semua <i>device</i> .	18	23	7	1	1	50
2	Menu-menu yang ada di aplikasi akademik UIN Sunan Kalijaga mudah dipahami.	4	19	18	9	0	50
3	Menu isi presensi MK dosen di akademik UIN Sunan Kalijaga mudah digunakan.	16	29	3	2	0	50
4	Menu isi nilai MK dosen di akademik UIN Sunan Kalijaga mudah digunakan.	10	19	13	8	0	50
5	Proses cetak laporan rekapitulasi presensi perkuliahan di menu laporan mata kuliah dosen mudah dilakukan.	10	19	14	7	0	50
6	Proses cetak laporan jurnal realisasi perkuliahan di menu laporan mata kuliah dosen mudah dilakukan.	3	13	24	8	2	50
7	Proses cetak laporan rekap nilai mata kuliah di menu laporan mata kuliah dosen mudah dilakukan.	4	19	25	1	1	50
8	Proses input data di aplikasi akademik UIN Sunan Kalijaga mudah dilakukan.	2	22	13	8	5	50
9	Berkas yang dibutuhkan untuk pengisian nilai mudah di <i>download</i> .	13	16	19	1	1	50
10	Aplikasi akademik UIN Sunan Kalijaga memiliki tata letak tampilan/ <i>layout</i> yang mudah dilihat.	13	20	14	3	0	50
	JUMLAH	93	199	150	48	10	500

Berdasar perhitungan tabel 1, kemudian dilakukan pengukuran dari setiap kategori (skala sikap). Hasil perhitungan kategori ini disajikan dalam tabel 2.

Tabel 2. Hasil Perhitungan Kategori.

Kategori	Bobot	Skor kategori	Total Bobot (Bobot X skor kategori)
Sangat Setuju	5	93	465
Setuju	4	199	796
Netral	3	150	450
Tidak Setuju	2	48	96
Sangat Tidak Setuju	1	10	10
Jumlah Total Bobot			1817
Rata-rata Total Bobot			363
Standar Deviasi			317

Berdasar perhitungan kategori pada tabel 2, total bobot yang memilih Sangat Setuju adalah 465, Setuju sebesar 796, Netral sebesar 450, Tidak Setuju sebesar 96 dan Sangat Tidak Setuju sebesar 10. Jumlah total bobot tersebut adalah 1817.

$$\text{Operability} = \frac{\text{Rata-rata skor}}{\text{Jumlah Opsi}} = \frac{363}{5} = 72,68 \text{ dengan Standar Deviasi} = \frac{317}{5} = 63,35$$

Dengan demikian nilai operability adalah $72,68 \pm 63,35$. Nilai ini menunjukkan bahwa pengguna setuju akan kemudahan operabilitas sistem informasi akademik.

3.4. Pengujian Integrity

Integrity merupakan metrik yang digunakan untuk mengetahui sejauh mana akses ke software dan data oleh pihak yang tidak berhak dapat dikendalikan. Nilai integrity ditentukan oleh faktor auditability, instrumentation, dan security.

Auditability adalah faktor kemudahan untuk memeriksa apakah software memenuhi standar yang telah ditentukan. Fitur SIA yang diuji sebanyak 47 fitur, terdapat 39 fitur yang sesuai dengan standar yang ditentukan. Sehingga nilai auditability adalah:

$$\text{Auditability} = \frac{\text{Jumlah fitur yang sesuai}}{\text{Jumlah fitur yang diuji}} \times 100\% = \frac{39}{47} \times 100\% = 82,98\%$$

Instrumentation adalah pengawasan dan identifikasi kesalahan operasi dari program. Fitur SIA yang diuji sebanyak 48 fitur, terdapat 44 fitur yang memiliki identifikasi kesalahan operasi dari program.

$$\text{Instrumentation} = \frac{\text{Jumlah fitur yang sesuai}}{\text{Jumlah fitur yang diuji}} \times 100\% = \frac{44}{48} \times 100\% = 91,67\%$$

Security adalah ketersediaan mekanisme untuk mengontrol serta melindungi program dan data terhadap akses dari pihak yang tidak berhak. Total aktivitas yang diuji sebanyak 5 fitur, terdapat 3 fitur yang memiliki security. Hasil pengujian security disajikan dalam tabel 3.

Tabel 3. Hasil Pengujian Security.

No	Activity	Product	Doc. System	Remark	Keterangan
1	Login	v	v	v	Sesuai
2	Session logout	v	v	v	Sesuai
3	Level user	v	v	v	Sesuai
4	Enskripsi password	v	x	x	Tidak Sesuai
5	Enskripsi url	v	x	x	Tidak Sesuai

Berdasar tabel 3, diperoleh nilai security:

$$\text{Security} = \frac{\text{Jumlah fitur yang sesuai}}{\text{Jumlah fitur yang diuji}} \times 100\% = \frac{3}{5} \times 100\% = 60\%$$

Dengan demikian metrik integrity dapat dihitung menggunakan rumus (4), yaitu:

$$\text{Nilai integrity} = \frac{\text{auditability} + \text{instrumentation} + \text{security}}{3}$$

$$\text{Nilai integrity} = \frac{82,98\% + 91,67\% + 60\%}{3} = \frac{234,65\%}{3} = 78,23\%$$

3.5. Pengujian Usability

Usability adalah usaha yang diperlukan untuk mempelajari, mengoperasikan, menyiapkan input, dan mengartikan output dari software. Usability memiliki dua faktor yaitu operability dan training. Operability adalah kemudahan mengoperasikan program. Sebagaimana yang sudah dibahas di bagian 3.3, faktor operability memiliki nilai sebesar $72,68 \pm 63,35$

Sedangkan training adalah sejauh mana software membantu pengguna baru untuk memahami sistem. Training telah dicoba sebanyak 30 kali, dalam sekali training terdapat 5 tugas yang harus dilakukan. Berdasarkan hasil pengujian training sistem informasi akademik memiliki rata-rata waktu pemahaman pemakaian sebesar $24,55 \text{ detik} \pm 3,0202 \text{ detik}$.

4. KESIMPULAN

4.1. Simpulan

Berdasarkan pengujian yang telah dilakukan terhadap sistem informasi akademik UIN Sunan Kalijaga, maka dapat disimpulkan bahwa penelitian ini telah berhasil menguji sistem informasi akademik UIN Sunan Kalijaga menggunakan kerangka teori McCall quality attributes dengan pendekatan black box dan white box testing. Hasil pengujian yang telah dilakukan maka dapat diambil kesimpulan yaitu :

1. Nilai Correctness sebesar 86,53 %.
2. Nilai Reliability sebesar 81,40 %.
3. Nilai Efficiency untuk sub faktor consiness sebesar $2,91\% \pm 1,49\%$, execution effisinsi sebesar $0,63\% \pm 0,07\%$ dan operability sebesar $72,68 \pm 63,35$.
4. Nilai Integrity sebesar 78,23 %.
5. Nilai Usability untuk faktor operability sebesar $72,68 \pm 63,35$ dan training sebesar $24,55 \text{ detik} \pm 3,0202 \text{ detik}$.

4.2. Saran

Penelitian ini masih memiliki kekurangan dan kelemahan. Saran untuk penelitian selanjutnya, perlu diperhatikan beberapa hal berikut:

1. Nilai masing-masing sub faktor dapat di tingkatkan dengan cara memperbaiki fitur yang ada di sistem informasi akademik yang belum berjalan sesuai dengan fungsinya.
2. Dapat melakukan pengujian dengan framework yang lain seperti ISO, IEEE atau yang lainnya.

DAFTAR PUSTAKA

Bachtiar, M. A., Dharmayanti, D. & Sabariyah, M. K., 2013. Analisis. *Majalah Ilmiah UNIKOM*, 11(2), pp. 224-233.

Pressman, R. S., 2002. *Rekayasa Perangkat Lunak Pendekatan Praktis (Buku Satu)*. Yogyakarta: Andi.

Sulaiman, D. Z., 2009. *Penggalian, Penerapan, Penjaminan Kualitas Perangkat Lunak Pada Perusahaan Perangkat Lunak (Study kasus di seluruh Indonesia)*, Jakarta: Tesis Uinversitas Indonesia.

Tresnasari, N. A., 2012. *Implementasi Model Penjaminan Mutu pada Organisasi Pengembangan Perangkat Lunak Skala Kecil*, Yogyakarta: Skripsi UIN Sunan Kalijaga.

Zahra, A., 2012. *Perancangan Model Rujukan Penjaminan Mutu pada Organisasi Pengembangan Perangkat Lunak Skala Kecil*, 2012: Skripsi UIN Sunan Kalijaga .

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) UIN Sunan Kalijaga Yogyakarta yang telah memberi dukungan dana terhadap penelitian ini. Penelitian ini dilaksanakan berdasar Keputusan Rektor UIN Sunan Kalijaga Nomor 163.7 Tahun 2015 Tentang Penetapan Personalia Penelitian dan Judul Penelitian Individu, Kelompok, Profesor dan Potensi HAKI UIN Sunan Kalijaga Yogyakarta Tahun Anggaran 2015 tertanggal 1 September 2015. Penelitian yang dibiaya LPPM UIN Sunan Kalijaga ini berjudul "Pemodelan Software Testing Menggunakan McCall's Quality Framework pada Aplikasi Sistem Informasi di Universitas Islam Negeri Sunan Kalijaga". Selain itu, penulis juga mengucapkan terima kasih kepada Agung Fatwanto, Ph.D yang telah memberi kesempatan untuk melakukan penelitian di Pusat Teknologi Informasi dan Pangkalan Data UIN Sunan Kalijaga dan Yessi Yunitasari, S.Kom yang telah membantu pelaksanaan penelitian ini.