

## DOKUMENTASI SOFTWARE TESTING BERSTANDAR IEEE 829-2008 UNTUK SISTEM INFORMASI TERINTEGRASI UNIVERSITAS

Mochamad Adam Harris Anwar<sup>1)</sup> Yudhi Kurniawan<sup>2)</sup>

<sup>1)2)</sup> Sistem Informasi Universitas Machung, Villa Puncak Tidar N-1 Malang  
email : 321510006@student.machung.ac.id<sup>1)</sup>, yudhi.kurniawan@machung.com<sup>2)</sup>

### Abstraksi

Perkembangan teknologi telah memasuki tahap dimana penggunaan aplikasi atau perangkat lunak sebagai core dari sebuah bisnis. Hal ini dilihat dari munculnya berbagai sistem informasi yang mulai dijadikan aset penting dalam mengembangkan sebuah bisnis usaha, salah satunya dalam bidang akademik. Seiring dengan bertambah dan berkembangnya kebutuhan pengguna maka sistem/aplikasi yang digunakan akan semakin membesar. Melalui tahapan testing kualitas sistem yang digunakan akan dapat terjaga dan terdokumentasi. Ma Chung Integrated Information System (macIS) merupakan sistem terintegrasi yang digunakan universitas Ma Chung sebagai pengolah data-data dan informasi akademik/non akademik mahasiswa, dosen, dan staff. Hingga saat ini macIS masih belum memiliki dokumentasi testing, maka dari itu diperlukan dokumentasi yang terstandar agar kualitas sistem informasi yang dimiliki universitas dapat terjaga dan berkembang dengan baik.

Berdasarkan latar belakang, terdapat dua masalah utama dalam sistem macIS. Pertama, macIS belum memiliki dokumentasi testing yang menyebabkan kualitas sistem tidak dapat dikontrol dengan baik. Kedua, tidak ada pendokumentasian yang terstandar pada sistem informasi universitas. Sedangkan Pengujian sistem dilakukan dengan menggunakan metodologi STLC (Software Testing Life Cycle) yang membagi proses pengujian sistem menjadi enam bagian mulai dari fase analisis, perencanaan, desain, spesifikasi kebutuhan perangkat, eksekusi, hingga ditutup dengan hasil luaran berupa dokumentasi pengujian perangkat lunak. Hasil akhir yang didapat adalah dokumentasi berupa skenario uji berdasarkan 4 aspek pada sistem dan lampiran dokumen MTP (Master Test Plan) berstandar IEEE-829.

### Kata Kunci :

Software testing, IEEE-829, sistem akademik, software testing life cycle

### Abstract

Technological developments have entered the stage where the use of applications or software as the core of a business. This is seen from the emergence of various information systems that began to be made as important assets in developing a business in various fields, one of which is in the academic field. Along with the growing of users needs, the used of system or application will be even bigger. Through the stages of quality testing the system will be maintained and documented. Ma Chung Integrated Information System (MacIS) is an integrated system used by Ma Chung University to process academic / non-academic data and information for students, lecturers, and staff. Until now, MacIS still does not have standardized testing documentation, therefore standardized documentation is needed so that the quality of the information system owned by the university can be maintained and developed properly.

Based on the study case, there are two main problems in the MacIS. First, the MacIS does not have testing documentation that causes the quality of the system cant controlled properly. Second, there is no standardized documentation on university information systems. System testing is carried out using the STLC methodology (Software Testing Life Cycle) which divides the system testing process into six parts starting from the analysis, planning, design, device requirements, execution, to the closing with the results of software testing documentation. The final results obtained are documentation in the form of a test scenario based on 4 aspects on the system and the MTP (Master Test Plan) document attachment with the IEEE-829 standard.

**Keywords :**

*Software testing, IEEE-829, academic system, software life cycle testing*

**1. PENDAHULUAN**

*Software Testing* merupakan suatu tahapan dimana pengujian dilakukan pada beberapa aspek dalam suatu aplikasi seperti kualitas, fitur/fungsi, keamanan, serta kinerja dari aplikasi tersebut agar memenuhi syarat dan kebutuhan dari pengguna. Definisi *testing* sendiri adalah proses menganalisa suatu entitas *software* untuk mendeteksi perbedaan antara kondisi yang ada dengan kondisi yang diinginkan (*defect/errors/bugs*) dan mengevaluasi fitur-fitur dari entitas *software* (standar ANSI/IEEE 1059). Dengan menggunakan tahapan *testing* maka kualitas sistem yang digunakan akan dapat terjaga dan terdokumentasi, karena *testing* berperan penting untuk mengukur seberapa baik kualitas aplikasi dimana pengguna bisa menggunakan aplikasinya serta menghindari hal-hal yang tidak diinginkan dalam proses pengoperasiannya. Universitas Ma Chung merupakan salah satu universitas swasta yang telah menggunakan sistem informasi akademik untuk memenuhi kebutuhan baik layanan maupun informasi yang dibutuhkan bagi seluruh civitas akademika. *Ma Chung Integrated Information System* (macIS) merupakan sistem terintegrasi yang digunakan universitas Ma Chung sebagai pengolah data-data dan informasi akademik/non akademik mahasiswa, dosen, dan staff. Banyak proses yang dicakup oleh sistem ini, mulai dari proses akademik seperti penerimaan mahasiswa baru, perkuliahan, dan wisuda hingga proses non akademik seperti keuangan dan perawatan aset.

Hingga saat ini macIS masih belum memiliki dokumentasi *testing* dikarenakan, macIS sendiri merupakan sistem akademik baru yang mana pengembangan dari sistem akademik sebelumnya dimana fitur/fungsi yang diimplementasikan lebih banyak dan memenuhi kebutuhan pengguna saat ini. Sebagai salah satu aset penting yang dimiliki universitas, tentunya diperlukan dokumentasi yang terstandar agar kualitas sistem informasi yang dimiliki universitas dapat terjaga dan berkembang dengan baik. Standar yang akan digunakan dalam pengujian perangkat lunak kali ini adalah IEEE 829-2008 *Software Testing* yang merupakan sebuah standar yang digunakan dalam proses menganalisis sebuah perangkat lunak untuk mendeteksi perbedaan kondisi yang ada dengan kondisi yang diperlukan atau yang sering disebut *bug* dan untuk mengevaluasi fitur dari perangkat lunak[1]. Pengerjaan dokumen *testing* ini menggunakan metode *Software Testing Life Cycle* (STLC) yang didefinisikan sebagai serangkaian kegiatan yang dilakukan untuk melakukan pengujian perangkat lunak

yang mengacu pada proses pengujian dengan langkah-langkah spesifik yang harus dilakukan dalam urutan tertentu untuk memastikan bahwa sasaran mutu terpenuhi[2], metode ini dibagi menjadi 6 fase, yaitu *analisis, planning, test case, setup, dan closure*.

## 2. METODE / ALGORITMA

Metode pengujian perangkat lunak untuk sistem informasi terintegrasi pada studi kasus Universitas Ma Chung ini akan menggunakan metodologi STLC (*Software Testing Life Cycle*) yang membagi proses pengujian sistem menjadi enam bagian mulai dari fase analisis, perencanaan, desain, spesifikasi kebutuhan perangkat, eksekusi, hingga ditutup dengan hasil luaran berupa dokumentasi pengujian perangkat lunak. Metode ini sangat cocok digunakan karena memiliki tahapan yang terstruktur dengan berbagai kegiatan yang akan dijelaskan secara spesifik, sehingga dapat didapatkan segala kebutuhan yang spesifik dan terinci untuk proses pengujian pada sistem ini. Berikut adalah rincian dari kegiatan yang dapat dilakukan dengan metode STLC (*Software Testing Life Cycle*).

### 2.1 Analisis Requirement

Pada fase ini akan dilakukan beberapa aktivitas yang akan dilakukan untuk menghasilkan dokumen kebutuhan pengguna. Diawali dengan kegiatan mengumpulkan data dengan metode wawancara/konsultasi kepada beberapa narasumber terkait sistem informasi terintegrasi universitas, dilanjutkan dengan menganalisis hasil dari wawancara/konsultasi, mendefinisikan siapa saja pengguna pada sistem yang akan diuji, merangkum list kebutuhan pengguna dari beberapa aspek, dan diakhiri dengan membuat *Requirement Traceability Matrix (Tabel 2)*. *Requirement Traceability Matrix* adalah tabel yang digunakan untuk menelusuri kebutuhan untuk tes yang diperlukan untuk memverifikasi bahwa persyaratan tersebut terpenuhi. Sebuah *Traceability Matrix* yang baik akan memberikan kemudahan dalam melakukan penelusuran kedepan atau kebelakang, yaitu kebutuhan dapat ditelusuri mulai dari fase requirement sampai fase testing atau sebaliknya yakni dari fase testing ke fase requirement[3].

Tabel 1 Metode Pengujian

Fase	Detil Kegiatan	Output
<b>Analisis Requirement</b>	Gathering data/wawancara.	Lembar hasil wawancara.
	Analisis kebutuhan sistem.	Dokumen requirement sistem.
	Memetakan requirement pengguna dengan fungsi dalam sistem menggunakan case.	Matrix RTM.
<b>Test Planning</b>	Persiapan rencana pengujian: Alat Uji, Upaya pengujian, perencanaan sumber daya.	Rencana/Strategi untuk proses uji.
<b>Test Case Development</b>	Membuat skenario testing berdasarkan matrix dan strategi uji yang telah dibuat.	Dokumen Test Case/Skenario Testing.
<b>Environment Setup</b>	Melihat spesifikasi perangkat yang dibutuhkan untuk menjalankan pengujian.	Spesifikasi hardware dan tools pendukung.
<b>Test Execution</b>	Merancang tabel case dengan dilengkapi ekspektasi outcome dan hasil pengujian	Hasil uji skenario testing
<b>Closure</b>	Meringkas hasil pengujian.	Dokumen software testing.
	Evaluasi dan dokumentasi	

Tabel 2 Requirement Traceability Matrix Sebelum Pengujian

Requirement Traceability	Business Requirement									
	BR1	BR2	BR3	BR4	BR5	BR6	BR7	BR8	BR9	BR10
	Login sistem	Dashboard sistem	Mastering	Keuangan Mahasiswa	Rekap prodi	Rekap mahasiswa	Status KRS	Generate NIM	Pengumuman	Setting
Test Case	TC1									
	TC2									
	TC3									
	TC4									
	TC5									
	TC6									
	TC7									
	TC8									
	TC9									
	TC10									
	TC11									
	TC12									
	TC13									
	TC14									
	TC15									

## 2.2 Test Planning

Pada fase ini akan mulai dirancang bagaimana metode pengujian yang paling tepat dan perencanaan upaya pengujian mulai dari menentukan waktu serta menentukan bagian sistem yang akan diuji. Pembuatan upaya pengujian atau rencana proses pengujian mulai disusun dari penggabungan beberapa aktivitas yang akan dilakukan secara urut dan terstruktur.

## 2.3 Test Case Development

Pada fase ini akan mulai dirancang berbagai skenario test yang akan dilakukan untuk menguji fungsi pada sistem sesuai dengan RTM dan *planning* yang sudah dibuat. Langkah awal yang diperlukan yaitu membuat tabel deskripsi awal yang berisikan test id dan deskripsi test sebagai acuan yang nantinya akan didetilkan lagi berupa sebuah tabel skenario.

## 2.4 Environment Setup

Pada fase ini akan dilihat bagaimana arsitektur yang diperlukan, lingkungan dan membuat daftar spesifikasi hardware dan software yang diperlukan untuk melakukan pengujian.

## 2.5 Test Execution

Pada fase ini akan dilakukan proses pengujian berdasarkan test skenario (**Tabel 3**) dan jadwal uji yang sudah dibuat. Pada fase ini juga dibuat ekspektasi outcome dari sistem yang diuji dan hasil dari proses pengujian itu sendiri.

Tabel 3 Test Case Description

Test Case ID	Test Case Description
TC01	<i>Login menggunakan kombinasi username dan password salah atau sebaliknya</i>
TC02	<i>Login menggunakan username yang tidak terdaftar</i>
TC03	<i>Login menggunakan password dengan menggunakan spasi</i>
TC04	<i>Input data melebihi kapasitas/length</i>
TC05	<i>Input data tidak pada tempatnya dengan menggunakan special character/symbol/number</i>
TC06	<i>Input blank data pada field</i>
TC07	<i>View function berdasarkan hak akses</i>
TC08	<i>View data inputan</i>
TC09	<i>View informasi berdasarkan hak akses</i>
TC010	<i>Input code html</i>
TC011	<i>Edit data tidak pada tempatnya dengan menggunakan special character/symbol/number</i>
TC012	<i>Edit dengan menggunakan blank data pada field</i>
TC013	<i>Cek fitur lupa password</i>
TC014	<i>Delete data master</i>
TC015	<i>Mengganti password lama dengan data baru</i>

Tabel 4 Test Execution

Test skenario 31				
Test case 1				
No.	Action	Input	Expected Outcome	Result
1.	Login menggunakan kombinasi username dan password salah atau sebaliknya	User : adam	User / password salah	Pass
		Pass : demo	Actual Outcome you have entered wrong Username or Password!	
		User : demo	Expected Outcome User / password salah	
2.	Login menggunakan kombinasi username dan password salah atau sebaliknya	Pass : 1234	Actual Outcome you have entered wrong Username or Password!	Pass
		User : demo	Expected Outcome User / password salah	
		Pass : 1234	Actual Outcome you have entered wrong Username or Password!	

### 2.6 Closure

Pada fase ini akan didiskusikan hasil dari siklus pengujian dan dianalisis bagaimana cara untuk meningkatkan strategi tes yang digunakan, menghilangkan kemacetan proses untuk siklus pengujian, dan berbagi cara/metode terbaik untuk proyek serupa di masa depan.

### 3. HASIL DAN PEMBAHASAN

Berikut merupakan salah satu hasil dari sistem yang diuji. Setelah dilakukan tahap pengujian didapatkan hasil RTM sebagai berikut.

Berdasarkan hasil uji Tabel 5, ditemukan hasil sebagai berikut :

- Aspek Security : 100% Pass dari 6 kali tahap pengujian dengan 3 case yang berbeda.
- Aspek Data Integrity : 100% Pass dari 22 kali tahap pengujian dengan 2 case yang berbeda
- Aspek Functional : 61% Pass sebanyak 11 kali dari 18 kali tahap pengujian dengan 7 case yang berbeda.
- Aspek Recovery : 100% Pass dari tahap pengujian dengan 2 case yang berbeda.

Tabel 5 Requirement Traceability Matrix Setelah Pengujian

Requirement Traceability		Business Requirement									
		BR1	BR2	BR3	BR4	BR5	BR6	BR7	BR8	BR9	BR10
		Login sistem	Dashboard sistem	Mastering	Keuangan Mahasiswa	Rekap prodi	Rekap mahasiswa	Status KRS	Generate NIM	Pengumuman	Setting
Test Case	TC1	✓									
	TC2	✓									
	TC3	✓									
	TC4			✓							
	TC5			✓							
	TC6			✓							
	TC7		✓	✓	✓	✓	✓	✓	✓	✓	
	TC8			✓							
	TC9		✓	✓	✓	✓	✓	✓	✓		
	TC10			✓							
	TC11			✓							
	TC12			✓							
	TC13										✓
	TC14										
	TC15										✓

Kode:

- SSTD-MacIS Keuangan Tabel RTM Security
- SSTD-MacIS Keuangan Tabel RTM Data Integrity
- SSTD-MacIS Keuangan Tabel RTM Functional
- SSTD-MacIS Keuangan Tabel RTM Recovery

#### 4. KESIMPULAN

Penerapan standar pengujian dengan menggunakan metode STLC pada proses uji dapat dikatakan cukup baik, karena selain proses pengerjaan yang urut dan tertata didapatkan hasil proses pengujian yang baik. Hampir seluruh sistem memiliki hasil *pass* lebih dari 50% yang dipetakan secara jelas dengan menggunakan matrix RTM. Pada bagian pembuatan case/skenario test dirancang berdasarkan requirement dan ditambahi beberapa case acak yang berguna untuk mencari kesalahan pada sistem.

Kesimpulan yang didapat setelah melakukan proses pengujian terhadap sistem MacIS adalah standar yang digunakan masih jauh dari hasil yang diharapkan. Ditemukan beberapa hasil *fail* dari proses uji pada semua sistem namun hasil *fail* pada proses uji bukan berarti sistem gagal memenuhi standar pengujian. Standar case/skenario test yang dibuat memang melebihi

batasan scope uji agar sistem tersebut dapat berkembang dan terarah dalam proses pengembangan kedepan. Untuk functional tiap sistem sendiri sudah berjalan sesuai dengan dokumen requirement yang sudah dibuat.

Untuk pengerjaan dokumentasi selanjutnya diharapkan case/skenario test yang dibuat lebih spesifik dan detil untuk tiap sistem, karena tiap sistem memiliki requirement yang berbeda. Selain itu jumlah case/skenario yang digunakan juga mengikuti kebutuhan sistem, semakin besar sistem yang diuji maka semakin besar juga scope case/skenario test yang harus dibuat.

## 5. REFERENSI

- [1] The Institute of Electrical and Electronics Engineers, 8 Juli 2019. IEEE Standard for Software and System Test Documentation. 3 Park Avenue. New York. <https://techterms.com/definition/ieee>.
- [2] Systeme Evolutif Limited, 8 Juli 2019, IEEE Test Plan Template, <https://jmpovedar.files.wordpress.com/2014/03/ieee-829.pdf>.
- [3] Hoongrad, Wim. 8 Juli 2019. STLC - Pengujian Perangkat Lunak Siklus Hidup, <https://www.itpedia.nl/id/2018/07/20/stlc-software-testing-life-cycle/>.
- [4] Daniel, Galin., 2018, Harlow: Software Quality Assurance from Theory to Implementation, Pearson Addison Wesley.
- [5] Erawati, Erna., 2018, Usability Testing Dengan ISO/IEC 9126-4 Sistem Informasi Akademik Universitas Pendidikan Ganesha Ditinjau Dari Pengguna Dosen.
- [6] Astuti, Puji., 2018, Penggunaan Metode *Black Box Testing (Boundary Value Analysis)* Pada Sistem Akademik (SMA/SMK).
- [7] Fadel Majid, Ali., 2018, Implementasi Metode *Cause Effect Graphing* Pada Pengujian Aplikasi Sistem Informasi Alumni FSM UNDIP.