

Dokumentasi *Software Testing* Berstandar IEEE 829-2008 untuk *Learning Management System* Fakultas Ilmu Komputer Universitas Subang

Tazkia Salsabila Ardan¹, Denada Fatimah Zahra², Fikri Rahmat Junaedi³, Septian Rheno Widianto⁴

Jurusan Sistem Informasi Bisnis,
Sekolah Tinggi Manajemen Ilmu Komputer LIKMI
Jl. Ir. H. Juanda No.96, Lebakgede, Kecamatan Coblong, Kota Bandung, Indonesia
denada.zahr@gmail.com¹, info@likmi.id^{2,3}, septian.rheno@yahoo.de⁴

Diterima: 16 November 2020. Disetujui: 5 Januari 2021. Dipublikasikan: 6 Januari 2020.

Abstract-- *Learning Management System (LMS) is a new software application that is used to help plan, and implement a learning process at the Faculty of Computer Science, University of Subang. Concerns about the quality of the system that cannot be controlled properly due to the absence of standard documentation on the LMS of the Faculty of Computer Science. And according to DeLone and McLean, the higher the quality of information produced by an information system will further increase user satisfaction, in this case the Faculty of Computer Science as a division of the University of Subang with the service sector, the quality of information is very influential in increasing the satisfaction of students and lecturers at the Faculty of Computer science. Therefore, standardized documentation is needed so that the quality of the LMS system at the Faculty of Computer Science, University of Subang can be maintained and developed properly. System testing is carried out using the Software Testing Learning Center (STLC) methodology which is divided into six testing phases, namely Requirement Analysis, Test Planning, Test Case Development, Text Execution and Closure. The test was conducted with 16 types of tests from 4 aspects, namely aspects of security, data integrity, functional, and recovery. The test results show that the security aspect produces 100% pass from 8 testing stages with 3 different cases. In the aspect of data integrity, it also produces 100% passes from 20 testing stages with 2 different cases. While the functional aspect only produced 82% Passes 15 times from 18 testing stages with 7 different cases. Another aspect is the recovery aspect which results in 100% pass from the testing phase with 2 different cases. After carrying out the testing process for the LMS system, the Faculty of Computer Science is a fairly good standard.*

Keywords: *Learning Management System (LMS, testing documentation, Software Testing Life Cycle (STLC)*

Abstrak-- *Learning Management System (LMS) merupakan sebuah aplikasi software baru yang digunakan untuk membantu merencanakan, dan mengimplementasikan sebuah proses pembelajaran Fakultas Ilmu Komputer, Universitas Subang. Kekhawatiran terhadap kualitas sistem yang tidak dapat dikontrol dengan baik dikarenakan belum adanya pendokumentasian yang berstandar pada LMS Fakultas Ilmu Komputer. Serta menurut DeLone and McLean semakin tinggi kualitas informasi yang dihasilkan suatu sistem informasi, akan semakin meningkatkan kepuasan pengguna, dalam hal ini Fakultas Ilmu Komputer sebagai divisi bagian dari Universitas Subang dengan bidang pelayanan, kualitas informasi sangat berpengaruh kepada meningkatnya kepuasan para mahasiswa dan dosen di Fakultas Ilmu Komputer. Maka dari itu diperlukannya dokumentasi yang berstandar agar kualitas sistem LMS Fakultas Ilmu Komputer, Universitas Subang dapat terjaga dan berkembang dengan baik. Pengujian sistem dilakukan dengan menggunakan metodologi *Software Testing Learning Center (STLC)* yang terbagi kedalam enam fase pengujian yaitu *Analisis Requirement, Test Planning, Test Case Development, Text Execution dan Closure*. Test dilakukan dengan 16 jenis test dari 4 aspek yaitu aspek *security, data integrity, fungsional, dan recovery*. Hasil pengujian memperlihatkan pada aspek *security* menghasilkan 100% *pass* dari 8 kali tahap pengujian dengan 3 *case* yang berbeda. Pada aspek *data integrity* juga menghasilkan 100% *pass* dari 20 kali tahap pengujian dengan 2 *case* yang berbeda. Sementara aspek fungsional hanya menghasilkan 82% *Pass* sebanyak 15 kali dari 18 tahap pengujian dengan 7 *case* yang berbeda. Aspek lainnya yaitu aspek *recovery* menghasilkan 100% *pass* dari tahap pengujian dengan 2 *case* yang berbeda. Setelah melakukan proses pengujian terhadap sistem LMS Fakultas Ilmu Komputer adalah standar yang sudah cukup baik.*

Kata kunci: *Learning Management System (LMS, dokumentasi testing, Software Testing Life Cycle (STLC)*

I. PENDAHULUAN

Pengujian program komputer (*software testing*) adalah merupakan suatu investigasi yang dilakukan

untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji. Pengujian seperti ini akan dapat mengukur kualitas program secara garis besar dari segi ketepatan (*correctness*),

kelengkapan (*completeness*), kebergunaan (*usability*), kinerja (*performance*) dan juga segi fungsional ataupun non-fungsional lainnya. 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 dan pengguna dapat menggunakan aplikasinya serta menghindari hal-hal yang tidak diinginkan dalam proses pengoperasiannya.

Fakultas Ilmu Komputer Universitas Subang merupakan salah satu fakultas yang menggunakan *Learning Management System*. *Learning Management System* (LMS) Fakultas Ilmu Komputer adalah sebuah aplikasi *software* yang dapat membantu yang dapat membantu merencanakan, mengimplementasikan sebuah proses pembelajaran. LMS memungkinkan pemilik atau pembuat *course* untuk *manage*, menyampaikan, dan memonitor para muridnya. LMS memadukan antara kursus tradisional dengan media digital dan alat interaktif meliputi kursus *online*, *virtual live sessions* dan forum diskusi.

Hingga saat ini LMS Fakultas Ilmu Komputer masih belum memiliki dokumentasi *testing* dikarenakan, ini LMS Fakultas Ilmu Komputer sendiri merupakan sistem manajemen pembelajaran baru dan memenuhi kebutuhan pengguna saat ini. Sebagai salah satu aset penting yang dimiliki fakultas, tentunya diperlukan dokumentasi yang terstandar agar kualitas sistem informasi yang dimiliki fakultas dapat terjaga dan berkembang dengan baik. Kekhawatiran terhadap kualitas sistem yang tidak dapat dikontrol dengan baik dikarenakan belum adanya pendokumentasian yang berstandar pada LMS Fakultas Ilmu Komputer. Serta menurut DeLone and McLean semakin tinggi kualitas informasi yang dihasilkan suatu sistem informasi, akan semakin meningkatkan kepuasan pengguna, dalam hal ini Fakultas Ilmu Komputer sebagai divisi bagian dari Universitas Subang dengan bidang pelayanan, kualitas informasi sangat berpengaruh kepada meningkatnya kepuasan para Mahasiswa dan Dosen di Fakultas Ilmu Komputer. 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 *analysis*, *planning*, *test case*, *setup*, dan *closure*.

Penelitian terdahulu mengenai *software testing* oleh Adi Purnomo (2013) dengan judul “*Software Testing Aplikasi Website PT. Gramedia Menggunakan Metode Blackbox Pada PT WGS Bandung*” [3]. Peneliti mendapatkan klien dari PT. Gramedia Pustaka Utama dengan mengetes aplikasi berbasis website berdasarkan *Functional Test*, *Security Test*, *Front End Performance Test*, *Back End Performance Test* dan *User Interface Test* pada web *staging* perusahaan yaitu www.publishing.gramediana.com. Penelitian lain tentang *software testing* oleh Reza Maulana, Indrawan Liwanto, Christouvel Lucma (2017) dengan judul “*Software Testing pada Aplikasi Website PT Semen Tonasa menggunakan Metode Assessment Vulnerability*” [4]. Peneliti menguji coba aplikasi berbasis website yang dimiliki PT Semen Tonasa berdasarkan *Security Test* pada Web perusahaan yaitu sementonasa.co.id dengan menggunakan Teknik *String Based Sql Injection* untuk mencari tahu kelemahan alamat web tersebut. Penelitian selanjutnya yaitu yang berkaitan dengan *software testing* berstandar IEEE 829-2008 pernah dilakukan sebelumnya oleh Muchamad Adam Harris Anwar dan Yudhi Kurniawan (2019) dengan judul “*Dokumen Software Testing Berstandar IEEE 829-2008 Untuk Sistem Informasi Terintegrasi Universitas*” [5]. Penelitian ini dilakukan karena Ma Chung *Integrate Information System* belum mempunyai pendokumentasian yang terstandar pada yang menyebabkan kualitas sistem tidak dikontrol dengan baik, dengan menggunakan metode SDLC (*Software Testing Life Cycle*). Dari beberapa penelitian yang telah diuraikan, maka terdapat perbedaan dengan penelitian ini seperti *software* yang dilakukan *testing* sangat berbeda, serta tempat penelitian yang berbeda pula.

II. METODE PENELITIAN

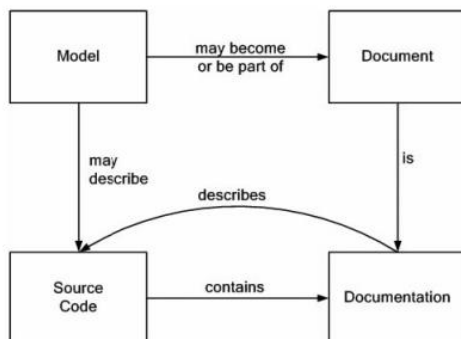
Dokumentasi pengujian perangkat lunak adalah elemen penting yang meningkatkan aktivitas eksperimental ke tingkat pengujian perangkat lunak. Dokumentasi merupakan sebuah artefak yang tujuannya untuk menyampaikan informasi tentang

sistem perangkat lunak yang menyertainya. Selain itu dokumentasi mempunyai fungsi sebagai berikut:

1. Bertindak sebagai media komunikasi antar anggota pengembang tim,
2. Penyimpanan sistem informasi untuk digunakan oleh *maintenance engineers*,
3. Membantu manajer proyek dalam merencanakan, mengatur anggaran, dan penjadwalan dalam proses pembangunan perangkat lunak,
4. Memberi penjelasan kepada pengguna bagaimana cara menggunakan dan mengelola sistem yang dibangun.

Sebagai tempat penyimpanan informasi, dokumen semestinya harus berisi informasi yang lengkap, *valid*, mudah dimengerti, dan *up-to-date*. Tetapi banyak pengembang yang membiarkan dokumen yang dibuat tidak memberikan informasi yang lengkap atau informasi yang tidak diperbaharui (*out-of-date*).

R Shujjar mengklasifikasikan dokumentasi ke dalam empat bagian yaitu dokumen kebutuhan, arsitektur dan desain, dokumen teknis, dokumen *end user*, dan dokumen pemasaran. Dokumen kebutuhan merupakan dokumen yang menjelaskan tentang atribut, kemampuan, karakteristik, atau kualitas dari suatu sistem yang merupakan dasar dari pembuatan suatu perangkat lunak. Dokumen arsitektur dan desain yaitu dokumen yang menjelaskan tentang arsitektur sistem dan prinsip-prinsip konstruksi yang akan digunakan dalam desain komponen perangkat lunak. Dokumen teknis merupakan dokumentasi dari kode, algoritma dan *interface*. Dokumen *end user* merupakan dokumen manual tentang bagaimana perangkat lunak tersebut digunakan. Dokumen pemasaran berisi bagaimana cara pemasaran dari produk dan analisis permintaan pasar.



Gambar 1. Hubungan antara *Source Code*, *Model*, *Document*, dan *Documentation*

Standar Pengujian perangkat lunak memiliki kelebihan dan kekurangan masing-masing. Ringkasan perbedaan masing-masing standar dapat di lihat pada Tabel 1.

TABEL I. PERBANDINGAN STANDAR PENGUJIAN PERANGKAT LUNAK

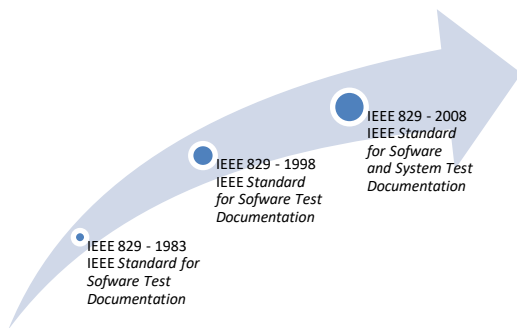
Aspek Pembeda-ing	Nama <i>Standard</i>		
	IEEE-829	IEEE 1008	BS-7925
Deskripsi	Merupakan Standar yang mendefinisikan sekumpulan dokumen untuk perangkat lunak dan pengujian sistem	Merupakan standar yang medefinisikan pendekatan pengujian unit yang menjadi acuan praktik pengujian perangkat lunak	Merupakan Standar yang mendefinisikan proses pengujian menggunakan pendekatan component testing
Proses Pengujian	Proses pengujian disesuaikan oleh penguji	Definisi serangkaian aktivitas pengujian unit	Definisi serangkaian proses pengujian komponen
Dokumenta-si Pengujian	Mendefinisikan format dokumentasi untuk 8 tahapan pengujian, yang kepada penguji diberikan kebebasan untuk mengisi semua atau tidak.	Tidak menjelaskan mengenai dokumentasi atau output yang berhubungan dengan pengajuan unit	Tidak menjelaskan dokumen output standar

Organisasi internasional seperti IEEE dan ISO telah menerbitkan standar untuk dokumentasi pengujian perangkat lunak. IEEE 829-2008, juga dikenal sebagai *829 Standard for Software and System Test Documentation*, adalah standar IEEE yang menetapkan bentuk sekumpulan dokumen untuk digunakan dalam delapan tahap pengujian perangkat lunak dan pengujian sistem, setiap tahap berpotensi menghasilkan sendiri jenis dokumen terpisah. Standar tersebut menetapkan format dokumen-dokumen ini, tetapi tidak menetapkan apakah semuanya harus diproduksi, juga tidak memasukkan kriteria apa pun mengenai konten yang memadai untuk dokumen-dokumen ini. Ini adalah masalah penilaian di luar lingkup standar. Dokumen tersebut adalah:

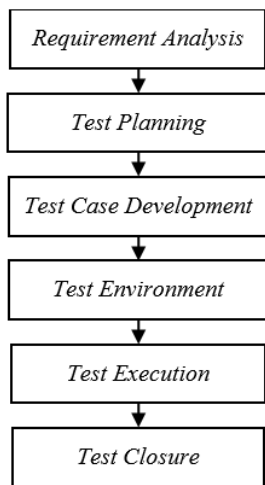
1. *Master Test Plan (MTP)*: Tujuan *Master Test Plan (MTP)* adalah untuk menyediakan

- perencanaan pengujian dan dokumen manajemen pengujian secara keseluruhan untuk berbagai level pengujian (baik dalam satu proyek atau di beberapa proyek).
2. Rencana Uji Tingkat (LTP): Untuk setiap LTP, ruang lingkup, pendekatan, sumber daya, dan jadwal kegiatan pengujian untuk tingkat pengujian yang ditentukan perlu dijelaskan. Item yang diuji, fitur yang akan diuji, tugas pengujian yang akan dilakukan, personel yang bertanggung jawab untuk setiap tugas, dan risiko terkait perlu diidentifikasi.
 3. *Level Test Design* (LTD): Merinci kasus tes dan hasil yang diharapkan serta kriteria lulus tes.
 4. Kasus Uji Tingkat (LTC): Menentukan data uji untuk digunakan dalam menjalankan kasus uji yang diidentifikasi dalam Desain Uji Tingkat.
 5. Prosedur Uji Tingkat (LTPr): Merinci cara menjalankan setiap pengujian, termasuk prasyarat penyiapan dan langkah-langkah yang perlu diikuti.
 6. *Level Test Log* (LTL): Untuk memberikan catatan kronologis detail yang relevan tentang pelaksanaan pengujian, misalnya merekam kasus pengujian mana yang dijalankan, siapa yang menjalankannya, dalam urutan apa, dan apakah setiap pengujian berhasil atau gagal.
 7. *Anomaly Report* (AR): Untuk mendokumentasikan setiap peristiwa yang terjadi selama proses pengujian yang memerlukan investigasi. Ini mungkin disebut masalah, insiden pengujian, cacat, masalah, masalah, anomali, atau laporan kesalahan. Dokumen ini sengaja diberi nama sebagai laporan anomali, dan bukan laporan kesalahan. Alasannya adalah bahwa ketidaksesuaian antara hasil yang diharapkan dan yang sebenarnya dapat terjadi karena sejumlah alasan selain kesalahan dalam sistem. Ini termasuk hasil yang diharapkan salah, pengujian yang dijalankan tidak benar, atau ketidakkonsistenan dalam persyaratan yang berarti bahwa lebih dari satu interpretasi dapat dibuat. Laporan tersebut terdiri dari semua rincian insiden seperti hasil aktual dan yang diharapkan, ketika gagal, dan bukti pendukung yang akan membantu dalam penyelesaiannya. Laporan tersebut juga akan mencakup, jika memungkinkan, penilaian dampak suatu insiden pada pengujian.
 8. Laporan Status Tes Sementara Tingkat (LITSR): Untuk meringkas hasil sementara dari kegiatan pengujian yang ditunjuk dan secara opsional untuk memberikan evaluasi dan rekomendasi berdasarkan hasil untuk tingkat tes tertentu.
 9. Laporan Uji Tingkat (LTR): Meringkas hasil dari kegiatan pengujian yang ditunjuk dan untuk memberikan evaluasi dan rekomendasi berdasarkan hasil setelah pelaksanaan pengujian selesai untuk tingkat pengujian tertentu.
 10. Laporan Uji Induk (MTR): Untuk meringkas hasil dari tingkat kegiatan pengujian yang ditunjuk dan untuk memberikan evaluasi berdasarkan hasil ini. Laporan ini dapat digunakan oleh organisasi mana pun yang menggunakan MTP. Laporan manajemen yang memberikan informasi penting yang ditemukan oleh pengujian yang dilakukan, dan termasuk penilaian kualitas upaya pengujian, kualitas sistem perangkat lunak yang diuji, dan statistik yang diperoleh dari *Anomaly Reports*. Laporan tersebut juga mencatat pengujian apa yang telah dilakukan dan berapa lama waktu yang dibutuhkan untuk meningkatkan perencanaan pengujian di masa mendatang. Dokumen akhir ini digunakan untuk menunjukkan apakah sistem perangkat lunak yang diuji sesuai untuk tujuan menurut memenuhi kriteria penerimaan yang ditentukan oleh pemangku kepentingan proyek atau tidak.
- Standar tersebut merupakan bagian dari silabus pelatihan Yayasan ISEB dan Sertifikat Praktisi dalam Pengujian Perangkat Lunak yang dipromosikan oleh British Computer Computer Society. ISTQB, mengikuti pembentukan silabusnya sendiri berdasarkan silabus ISEB dan ASQF Jerman, juga mengadopsi IEEE 829 sebagai standar referensi untuk perangkat lunak dan dokumentasi pengujian sistem. David Gelperin dan Dr. William C. Hetzel mengembangkan metodologi *Systematic Test and Evaluation Process* (STEP) untuk menerapkan Standar IEEE-829 yang asli untuk Dokumentasi Pengujian Perangkat Lunak. Pada penelitian ini, penulis hanya menggunakan dokumen *Level Test Design* (LTD).
- Dokumen Testing IEEE 829 telah mengalami beberapa kali perubahan versi yang dapat dilihat di Gambar 2.
- A. *Software Testing Life Cycle* (STLC)
- Metode pengujian pada *software* pada studi kasus *Learning System Management*, Fakultas Ilmu Komputer, Universitas Subang 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 yang secara garis besar tergambar pada penyajian Tabel 2. [8]. Penelitian ini menggunakan metode STLC didasari oleh tahapan yang terstruktur yang dimiliki metode tersebut dapat menjelaskan berbagai kegiatan secara spesifik sehingga peneliti mendapatkan kebutuhan spesifik dan terperinci untuk proses pengujian *Learning Management System* seperti pada Gambar 3. Pengujian perangkat lunak adalah area yang sangat luas, yang melibatkan banyak area teknis dan non-teknis lainnya, seperti spesifikasi, desain dan implementasi, pemeliharaan, proses, dan masalah manajemen dalam rekayasa perangkat lunak [6].



Gambar 2. Sejarah Versi Dokumen Testing IEEE 829



Gambar 3 Tahapan STLC

TABEL II. SOFTWARE TESTING LIFE CYCLE

Fase	Kegiatan	Output
Analisis Requirement	Wawancara Analisis Kebutuhan Sistem Memetakan Requirement menggunakan case	Hasil wawancara Dokumen Requirement Matrix RTM

Fase	Kegiatan	Output
Test Planning	Persiapan rencana pengujian	Rencana proses uji
Test Case Development	Membuat scenario testing	Dokumen Test Case
Test Case Development	Melihat spesifikasi perangkat yang dibutuhkan	Dokumentasi Test Case/Skenario Testing
Text Execution	Merancang tabel case dengan dilengkapi ekspektasi outcome	Spesifikasi hardware dan tools pendukung
Closure	Meringkas hasil pengujian Evaluasi dan dokumentasi	Dokumentasi software testing

1. Analisis Kebutuhan

Fase pertama yang merupakan fase Analisis merupakan fase dasar dari proses pengujian perangkat lunak. Fase ini mencakup analisis kebutuhan fungsional dan *non-fungsional* misalnya. persyaratan bisnis, dokumen spesifikasi fungsional dan dokumen spesifikasi teknis, dan lain-lain. Pengumpulan persyaratan dan harus dilakukan untuk penjelasan dengan pelanggan untuk mengidentifikasi hasil aktual dan yang diharapkan dari pengujian seperti Identifikasi persyaratan dan celah, yang pada dasarnya merupakan persyaratan non-fungsional seperti kegunaan, skalabilitas, kemampuan pengujian, pemeliharaan, kinerja, dan keamanan. Semua persyaratan yang tidak dapat diuji karena kendala sistem dan lingkungan pengujian harus dikomunikasikan kepada tim bisnis. Selama fase ini, tim penguji meninjau dan menganalisis persyaratan dan mengidentifikasi pengujian, yang akan dilakukan dan menetapkan prioritas untuk anggota tim penguji [7].

Tahap ini dilakukan dengan melakukan beberapa kegiatan guna menghasilkan dokumen kebutuhan pengguna. Tahapan dalam tahap ini diawali dengan kegiatan pengumpulan data melalui wawancara dengan beberapa narasumber di Fakultas Ilmu Komputer Universitas Subang mengenai *software Learning Management System*. Tahap kedua menganalisis hasil wawancara dan menghasilkan kesimpulan tentang kebutuhan pengguna dari beberapa aspek yang dapat diuraikan dalam *Requirement Traceability Matrix* yang tercantum pada Tabel 3. Tabel tersebut digunakan untuk menentukan persyaratan pengujian dan memverifikasi persyaratan yang ada di IEEE 829-2008 terpenuhi. Melalui hasil tersebut dapat terlihat pemetaan setiap fase dalam pengujian *software*.

TABEL III. REQUIREMENT TRACEABILITY MATRIX

Requirement Traceability	Business Requirement							
	BR1	BR2	BR3	BR4	BR5	BR6	BR7	BR8
TEST CASE	TC1							
	TC2							
	TC3							
	TC4							
	TC5							
	TC6							
	TC7							
	TC8							
	TC9							
	TC10							
	TC11							
	TC12							
	TC13							
	TC14							
	TC15							

2. Test Planning

Tahap persiapan ujian meliputi penyusunan rencana ujian, kasus ujian, data ujian dan persiapan lingkungan ujian. Pada tahap perencanaan pengujian, akan dimulai perancangan metode pengujian yang paling sesuai dan perencanaan upaya pengujian mulai dari penentuan waktu pengujian hingga sistem yang akan diuji. Rencana pengujian adalah dokumen pertama yang harus disiapkan, yang menguraikan ruang lingkup, tujuan, fitur yang akan diuji, fitur yang tidak akan diuji, jenis pengujian yang akan dilakukan, peran dan tanggung jawab tim pengujian, kriteria dan asumsi masuk dan keluar [9]. Pada penelitian ini dilakukan uji rencana dengan menggunakan metode STLC sesuai dengan hasil tinjauan pustaka. Sistem yang akan diujicobakan sudah ditentukan di awal penelitian. Pengujian akan dilakukan pada Sistem Manajemen Pembelajaran Fakultas Ilmu Komputer Universitas Subang.

3. Test Case Development

Tahap ini akan memulai merancang skenario pengujian yang akan dilakukan untuk menguji fungsi sistem sesuai RTM dan rencana yang telah dibuat pada tahap perencanaan pengujian. Secara bersamaan, tim penguji mulai mempersiapkan kasus uji dan data uji. Kasus uji adalah dokumen, yang menguraikan langkah-langkah yang diperlukan untuk menguji fungsionalitas apa pun dengan hasil yang diharapkan dan aktual. Jika hasil sebenarnya tidak sesuai dengan hasil yang diharapkan, bug dibuka. Untuk setiap persyaratan, kasus uji positif dan negatif disiapkan, yang dikonfirmasi oleh matriks ketertelusuran persyaratan (RTM). RTM adalah dokumen yang memetakan persyaratan dengan kasus uji untuk memastikan pengujian 100% dilakukan. Semua kumpulan data pengujian yang valid dan tidak valid harus disiapkan untuk setiap kasus pengujian dan dokumen data pengujian harus disiapkan. Data uji juga dihasilkan berdasarkan beberapa algoritma dan alat

[10]. Penyusunan kasus uji [11] memiliki berbagai tahapan yang dimulai dengan pembuatan kasus uji [12], pemilihan kasus uji [13], evaluasi, dan prioritas kasus uji [14] [15]. Ada berbagai algoritma yang digunakan untuk menghasilkan dan mengoptimalkan kasus uji [16] [17] [18].

Swain et.al mengusulkan teknik untuk menghasilkan kasus uji menggunakan diagram urutan yang tepat serta menentukan batas di seluruh artefak yang ditentukan. Pada saat yang sama, teknik pembuatan kasus uji berguna untuk mendeteksi sinkronisasi dan ketergantungan kasus penggunaan dan pesan, interaksi objek, dan kesalahan operasi [19]. Langkah yang dilakukan pada tahap ini adalah membuat tabel deskripsi awal yang diperlukan. Tabel seperti yang terlihat pada Tabel 4. berisi test *id* dan deskripsi *test* sebagai referensi yang nantinya akan dirinci kembali dalam bentuk skenario.

TABEL IV. DESKRIPSI PENGUJIAN

Test ID	Deskripsi Test
TC01	Login menggunakan kombinasi <i>username</i> dan <i>password</i> yang salah.
TC02	Login menggunakan <i>username</i> yang tidak terdaftar.
TC03	Login menggunakan <i>password</i> dengan menggunakan spasi.
TC04	Input data melebihi kapasitas.
TC05	Input data tidak pada tempatnya dengan menggunakan <i>special character</i> .
TC06	Input blank pada <i>filed</i> .
TC07	View function berdasarkan hak akses.
TC08	View data inputan.
TC09	View Informasi berdasarkan hak akses.
TC10	Input Code html.
TC11	Edit data tidak pada tempatnya dengan menggunakan <i>special character</i> .
TC12	Edit dengan menggunakan data blank pada <i>field</i> .
TC13	Cek fitur lupa <i>password</i> .
TC14	Delete data <i>master</i> .
TC15	Mengganti <i>Password</i> lama dengan data baru.

4. Environment Setup

Tahap ini akan melihat bagaimana arsitektur yang dibutuhkan, lingkungan dan daftar spesifikasi perangkat keras dan lunak yang dibutuhkan untuk melakukan pengujian. Persiapan lingkungan pengujian adalah salah satu fase terpenting yang biasanya

disiapkan oleh lingkungan penanganan tim terpisah. Setelah menyelesaikan bagian pengkodean, kode tersebut diperiksa oleh alat manajemen konfigurasi dan kemudian pembuatan pengujian disiapkan di mana pengujian harus memulai pelaksanaan pengujian [8].

5. Test Execution

Pada tahap ini proses pengujian dilakukan berdasarkan skenario pengujian. dan jadwal pengujian yang telah dibuat. Pada tahap ini juga dibuat hasil yang diharapkan dari sistem yang diuji dan hasil dari proses pengujian itu sendiri. Dalam fase ini pengujian menjalankan perangkat lunak sesuai dengan kasus pengujian. Jika hasil aktual dan yang diharapkan tidak cocok, pengujian membuka *bug* dan menetapkan hal yang sama kepada pengembang. *Logging* dan pelacakan *bug* [12] mengikuti siklus hidup bug lengkap. Banyak pekerjaan yang telah dilakukan di masa lalu yang berfokus pada langkah-langkah utama yang harus dilakukan untuk melaporkan kesalahan yang valid. Laporan rutin dapat didiskusikan setiap minggu / setiap hari seiring kemajuan proyek dalam pelaksanaan proyek, penerimaan dan persetujuan dipantau untuk menganalisis proyek percontohan.

6. Closure

Pada tahap ini akan membahas hasil dari pengujian siklus dan menganalisis cara memperbaiki strategi pengujian yang digunakan, memperbaiki masalah proses pada siklus pengujian, dan mencari metode alternatif yang dapat digunakan dalam penelitian selanjutnya. Penutupan Uji adalah fase kritis yang mencakup semua laporan pengujian yang memastikan bahwa semua sistem, integrasi, pengujian penerimaan pengguna lolos, dan keputusan diambil apakah semua persyaratan telah diuji dan tidak ada *critical bug* yang menunggu untuk diperbaiki atau diverifikasi. Peninjauan semua artefak uji dilakukan oleh Manajer. Setelah semua artefak ditinjau dan disetujui, rilis perangkat lunak selesai. Analisis akar penyebab lebih lanjut sedang dilakukan untuk bertukar pikiran tentang apa yang berjalan dengan baik, apa yang tidak berhasil, dan area perbaikan. Ada berbagai alat dan metode analisis akar penyebab yang tersedia di mana banyak penelitian telah dilakukan di masa lalu.

Fase-fase tersebut digambarkan dalam tabel demografi pada Tabel 5. dibawah ini. Partisipan dalam pembuatan dokumen *software testing* dibagi kedalam tiga peran yaitu *Test Manager*, *Test Analyst*, dan *Tester*. *Test Manager* memiliki tanggung jawab memastikan kualitas seluruh pengujian yang terjadi pada semua fase dan menentukan standar pengujian. Sementara *Test Analyst* memiliki tanggung jawab

dalam proses pengujian termasuk merancang kasus uji dan mengembangkan proses pengujian. *Tester* bertanggung jawab dalam membuat dokumentasi yang sesuai dengan kasus pengujian, lingkup pengujian dan kebutuhan tes. Ketiga peran tersebut dijalankan oleh peneliti dan pihak Universitas Subang.

TABEL V. DEMOGRAFI PENELITIAN

N O	FASE	PARTISIPAN	DURASI PENELITIAN	INSTRUMEN PENELITIAN
1	<i>Requirement Analysis</i>	<i>Test Analyst</i>	1 minggu	<i>Requirement Traceability Matrix</i>
2	<i>Test Planning</i>	<i>Test Analyst</i>	1 minggu	Rencana Proses Uji
3	<i>Test Case Development</i>	<i>Test Manager</i>	1 minggu	<i>Test case</i> dan <i>Skenario Testing</i>
4	<i>Environment Setup</i>	<i>Test Manager</i>	1 minggu	Daftar spesifikasi <i>Hardware</i> dan <i>Software</i>
5	<i>Test Execution</i>	<i>Tester</i>	3 minggu	<i>Expectation outcome</i> dan <i>Test Execution</i>
6	<i>Closure</i>	<i>Tester</i>	1 minggu	Dokumen <i>Software Testing</i> .

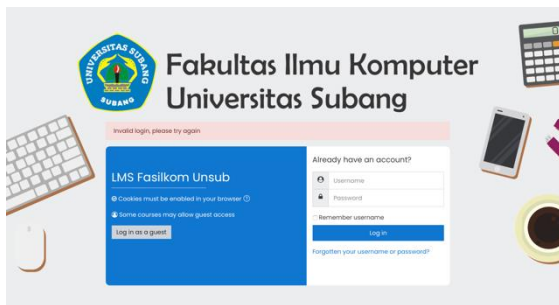
III. HASIL DAN PEMBAHASAN

Pengujian dilakukan berdasarkan *test scenario* yang telah dibuat pada Tabel 3. Dengan *test case id* TC01 terlihat *test execution* pada Tabel 6. dengan aksi *Login* menggunakan kombinasi *username* atau *password* yang sama pada *business requirement* B01 yaitu *User Access*. Pada pengujian sistem TC01 ekpektasi outcome adalah *username/password* salah dan *actual outcome* seperti yang terlihat pada Gambar 4. yaitu *invalid Login please try again*.

TABEL VI. TEST EXECUTION TABEL CASE 1

Test Case 1				
No	Action	Input	Expected Outcome	Result
1	<i>Login</i> menggunakan kombinasi dan <i>password</i> yang salah atau sebaliknya	<i>User: Tazkia</i> <i>Pass: testing</i>	<i>Username/ Password</i> salah <i>Actual Outcome</i> <i>Invalid Login Please Try Again</i>	<i>Pass</i>

Test Case 1				
No	Action	Input	Expected Outcome	Result
2		User: Denada Pass:1234	Expected Outcome	Pass
	Username/ Password Salah			
	Actual Outcome			
	Invalid Login Please Try Again			

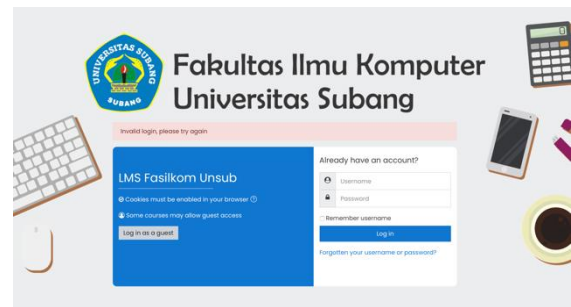


Gambar 4. Actual Outcome TC01

Pengujian selanjutnya dilakukan *test case id* TC02 terlihat *test execution* pada Tabel 7. dengan aksi menggunakan *username* yang tidak terdaftar pada LMS Fakultas Ilmu Komputer, Universitas Subang. Pengujian dilakukan pada *business requirement* B01 yaitu *User Access*. Pada pengujian sistem TC02 ekpektasi *outcome* adalah *username* tidak terdaftar dan *actual outcome* seperti yang terlihat pada Gambar 5. yaitu *invalid Login please try again*.

TABEL VII. TEST EXECUTION TABEL CASE 2

Test Case 2				
No	Action	Input	Expected Outcome	Result
1	Login menggunakan <i>username</i> yang tidak terdaftar	User: Fikri Pass: 1234	Username tidak terdaftar	Pass
			Actual Outcome	
			Invalid Login Please Try Again	



Gambar 5. Actual Outcome TC02

Pengujian TC03 dilakukan terlihat *test execution* pada Tabel 8. dengan aksi *Login* menggunakan *password* dengan menggunakan spasi pada LMS Fakultas Ilmu Komputer, Universitas Subang. Pengujian dilakukan pada *business requirement* B01 yaitu *User Access*. Pada pengujian sistem TC03 ekpektasi *outcome* adalah *username/password* salah dan *actual outcome* seperti yang terlihat pada Gambar 6. yaitu *invalid Login please try again*.

TABEL VIII. TEST EXECUTION TABEL CASE 3

Test Case 3				
No	Action	Input	Expected Outcome	Result
1	Login dengan menggunakan <i>password</i> dengan menggunakan spasi	User; Tazkia Pass: 1234	Username/ Password salah	Pass
			Actual Outcome	
			Invalid Login Please Try Again	



Gambar 6. Actual Outcome TC03

Pengujian selanjutnya dilakukan untuk TC04 yaitu dengan aksi input data melebihi kapasitas. Dalam pengujian ini peneliti mencoba menuliskan abjad tanpa henti. *Expected income* dari pengujian ini adalah tidak


dapat input lagi begitupun dengan *expected income* tidak dapat input lagi. Dalam pengujian TC03 dilakukan pada *business requirement* yaitu *General User Management*.

TABEL IX. TEST EXECUTION TABEL CASE 4

Test Case 4				
No	Action	Input	Expected Outcome	Result
1	Input data melebihi kapasitas	Menuliskan seluruh abjad tanpa henti	Tidak dapat input lagi	Pass
			Actual Outcome	
			Tidak dapat input lagi	

Pengujian pada *Test Case* kelima dengan test id TC05 dilakukan pada *business requirement* yang sama dengan TC04 yaitu *General User Management*. Aksi pada pengujian ini adalah input data tidak pada tempatnya menggunakan *special character*. Pada pengujian ini peneliti menuliskan email dengan "ccccccc" dengan *expected outcome* salah email dan *actual outcome* adalah 414leee82 seperti yang tertera pada Tabel 10.

TABEL X. TEST EXECUTION TABEL CASE 5

Test Case 5				
No	Action	Input	Expected Outcome	Result
1	Input data tidak pada tempatnya dengan menggunakan special karakter	Email: eeeee	Salah email	Pass
			Actual Outcome	
				

Sementara itu pengujian untuk *Test Case 6* dengan id test TC06 menggunakan aksi *input blank* pada *filed* dengan tidak menginput apapun pada kolom. *Expected outcome* pada pengujian ini adalah harap diisi sementara *actual outcome* adalah missing given name seperti yang terlihat pada Tabel 11. Pengujian ini dilakukan pada *business requirement* yang sama dengan TC04 dan TC05 yaitu *General User Management*.

TABEL XI. TEST EXECUTION TABEL CASE 6

Test Case 6				
No	Action	Input	Expected Outcome	Result
1	Input blank pada fileld	Tidak input apapun	Harap diisi	Pass
			Actual Outcome	
			- Missing given nome	

Pengujian *test case* ketujuh dengan id test TC07 dilakukan pada beberapa *business requirement* yaitu *Dashboard System*, *General User Management*, *Learner/ Student Management*, *Learning Content Management Capabilities*, *Classroom Management*, *Online Learning Portal*, *Learning Management Report*, dan *Plugin Management*. Aksi pada pengujian ini terbagi dua yaitu hak akses Admin dan hak akses Mahasiswa dengan keduanya memiliki aksi *view function* berdasarkan hak akses. Seperti terlihat pada Tabel 12. bahwa *expected outcome* untuk kedua hak akses baik mahasiswa maupun admin adalah dapat menggunakan semua fungsi sementara untuk hak akses. Sementara *actual outcome* berbeda, untuk hak akses Admin dapat menggunakan semua fungsi sementara untuk mahasiswa tidak dapat menggunakan semua fungsi.

TABEL XII. TEST EXECUTION TABEL CASE 7

Test Case 7				
No	Action	Input	Expected Outcome	Result
1	View function berdasarkan hak akses	Hak akses: Admin	Data menggunakan semua fungsi	Pass
			Actual Outcome	
			Dapat menggunakan semua fungsi	
2	View function berdasarkan hak akses	Hak akses: Mahasiswa	Expected Outcome	Pass
			Tidak dapat menggunakan semua fungsi	
			Actual Outcome	
	Tidak dapat menggunakan semua fungsi			

Pengujian selanjutnya dilakukan pada *test case 8* dengan id TC08 yang dilakukan pada *business requirement General User Management*. Aksi pada

pengujian ini adalah *view data inputan* dengan input melihat data. *Expected income* pada pengujian ini seperti terlihat pada Tabel 13. Sementara *actual income* pada pengujian ini adalah dapat dilihat.

TABEL XIII. TEST EXECUTION TABEL CASE 8

Test Case 8				
No	Action	Input	Expected Outcome	Result
1	View data inputan	Melihat Data	Dapat dilihat	Pass
			Actual Outcome	
			Dapat dilihat	

Pengujian *test case* kesembilan dengan *id test* TC09 dilakukan pada beberapa *business requirement* yaitu *Dashboard System, General User Management, Learner/ Student Management, Learning Content Management Capabilities, Classroom Management, Online Learning Portal, Learning Management Report, dan Plugin Management*. Aksi pada pengujian ini terbagi dua yaitu hak akses Admin dan hak akses Mahasiswa dengan keduanya memiliki aksi *view* informasi berdasarkan hak akses. Seperti terlihat pada Tabel 14. bahwa *expected outcome* untuk hak akses admin adalah dapat melihat seluruh informasi sementara untuk hak akses mahasiswa semester 3 adalah hanya dapat melihat informasi semester 3. *Actual outcome* untuk kedua akses sama dengan *expected outcome* masing-masing akses tersebut.


TABEL XIV. TEST EXECUTION TABEL CASE 9

Test Case 9				
No	Action	Input	Expected Outcome	Result
1	View Informasi berdasarkan hak akses	Hak akses: Admin	Dapat melihat seluruh informasi	Pass
			Actual Outcome	
			Dapat melihat seluruh informasi	
2	View informasi berdasarkan hak akses	Hak Akses: Mahasiswa Semester 3	Expected Outcome	Pass
			Hanya dapat melihat informasi semester 3	
			Actual Outcome	

Test Case 9				
No	Action	Input	Expected Outcome	Result
			Hanya dapat melihat informasi semester 3	

Pengujian selanjutnya dilakukan pada *test case* 11 dengan id TC11 yang dilakukan pada *business requirement General User Management*. Aksi pada pengujian ini adalah edit data tidak pada tempatnya dengan menggunakan *special character*. Pada pengujian ini peneliti memasukan angka 1234 pada kolom email. *Expected income* pada pengujian ini seperti terlihat pada Tabel 15 yaitu email salah. Sementara *actual income* pada pengujian 414 Ieee82.

TABEL XV. TEST EXECUTION TABEL CASE 11

Test Case 11				
No	Action	Input	Expected Outcome	Result
1	Edit data tidak pada tempatnya dengan menggunakan special character	Email: 1234	Email salah	Pass
			Actual Outcome	
				

Pengujian selanjutnya dilakukan pada *test case* 12 dengan nomor test id TC14 dengan aksi edit dengan menggunakan *data blank* pada *field*. Peneliti tidak mengisikan apapun pada kolom. Terlihat pada Tabel 16. bahwa *expected outcome* adalah harap diisi sementara *actual outcome* adalah *missing given name*. Pengujian TC12 dilakukan pada *business requirement General User Management*.

TABEL XVI. TEST EXECUTION TABEL CASE 12

Test Case 12				
No	Action	Input	Expected Outcome	Result
1	Edit dengan menggunakan data blank pada field		Harap diisi	Pass
			Actual Outcome	
			- Missing given name	

Pengujian *test case* 13 dengan *test id* TC13 dilakukan pada *business requirement BR10* yaitu *Administrative Function*. Pengujian ini untuk mengecek fitur lupa *password* dengan pengujian klik *forgotten username or password username*. *Expected*

outcome pada pengujian ini adalah new password seperti terlihat pada Tabel 17.

TABEL XVII. TEST EXECUTION TABEL CASE 13

Test Case 13				
No	Action	Input	Expected Outcome	Result
1	Cek fitur lupa password	Klik forgotten username or password by username	New password	Pass
			Actual Outcome	

Pengujian test case 14 dengan test id TC14 dilakukan pada business requirement BR10 yaitu Administrative Function. Pengujian ini dilakukan untuk delete data master dengan input delete seperti terlihat pada Tabel 18. Expected outcome pada pengujian ini adalah terhapus dan actual outcome adalah data terhapus.

TABEL XVIII. TEST EXECUTION TABEL CASE 14

Test Case 14				
No	Action	Input	Expected Outcome	Result
1	Delete data master	Delete	Terhapus	Pass
			Actual Outcome	
			Data terhapus	

Pengujian test case 15 dengan test id TC15 dilakukan pada business requirement BR10 yaitu Administrative Function. Pengujian ini untuk mengganti password lama dengan data baru dengan input ganti password. Terlihat pada Tabel 19. untuk expexted outcome adalah password berhasil diganti dan actual outcome adalah password berhasil diganti.

TABEL XIX. TEST EXECUTION TABEL CASE 15

Test Case 15				
No	Action	Input	Expected Outcome	Result
1	Mengganti password lama dengan data baru	Ganti Password	Password berhasil diganti	Pass
			Actual Outcome	
			Password berhasil diganti	

Setelah melakukan pengujian sistem TC1 hingga TC15 pada 10 business requirement pada 4

aspek yaitu security, data integrity, functional, dan recovery maka didapatkan hasil sebagai berikut:

TABEL XX. HASIL PENGUJIAN DALAM RTM

		Business Requirement									
		BR1	BR2	BR3	BR4	BR5	BR6	BR7	BR8	BR9	BR10
TEST CASE	Requirement Traceability	User Access Management	Dashboard System	General User Management	Learner/Student Management	Learning Content Management	Classroom Management	Online Learning Portal	Learning Management Report	Plugins Management	Administrative Functions
	TC1	✓									
	TC2	✓									
	TC3	✓									
	TC4			✓							
	TC5			✓							
	TC6			✓							
	TC7		✓	✓	✓	✓	✓	✓	✓	✓	✓
	TC8			✓							
	TC9		✓	✓	✓	✓	✓	✓	✓	✓	✓
	TC10			✓							
	TC11			✓							
TC12			✓								

	Business Requirement									
	BR1	BR2	BR3	BR4	BR5	BR6	BR7	BR8	BR9	BR10
Requirement Traceability	User Access Management	Dashboard System	General User Management	Learner/Student Management	Learning Content Management	Classroom Management	Online Learning Portal	Learning Management Report	Plugins Management	Administrative Functions
TC13										✓
TC14										✓
TC15										✓

Kode :

- SSTD – LMS Tabel RTM Security
- SSTD – LMS Tabel RTM Data Integrity
- SSTD – LMS Tabel RTM Functional
- SSTD – LMS Tabel RTM Recovery

Berdasarkan hasil uji Tabel 18, didapatkan Hasil *requirement traceability* pada aspek *security* menghasilkan 100% *pass* dari 8 kali tahap pengujian dengan 3 *case* yang berbeda. Pada aspek *data integrity* juga menghasilkan 100% *pass* dari 20 kali tahap pengujian dengan 2 *case* yang berbeda. Sementara aspek fungsional hanya mengasilkan 82% *Pass* sebanyak 15 kali dari 18 tahap pengujian dengan 7 *case* yang berbeda. Aspek lainnya yaitu aspek *recovery* menghasilkan 100% *pass* dari tahap pengujian dengan 2 *case* yang berbeda.

IV. KESIMPULAN DAN SARAN

Penerapan standar pengujian dengan metode STLC pada proses uji dengan proses pengerjaan yang urut dan tertata didapatkan hasil proses pengujian yang baik. Hampir keseluruhan sistem mempunyai hasil lulus 100%, dengan dipetakan menggunakan matrix RTM. Pada pembuatan *case/skenario test* dirancang berdasarkan *requirement* dan ditambahi beberapa *case*

acak yang berguna untuk mencari kesalahan pada sistem.

Dengan menggunakan Metode pengujian pada *software* pada studi kasus *Learning Management System*, Fakultas Ilmu Komputer, Universitas Subang ini menggunakan metodologi STLC (*Software Testing Life Cycle*) hasilnya kesimpulan yang di dapatkan setelah melakukan proses pengujian terhadap sistem *Learning Management System* Fakultas Ilmu Komputer adalah standar yang dimiliki sudah cukup baik. *Functional* sistem yang ada sudah berjalan sesuai dengan *document requirement* yang telah dibuat. Serta *system* yang dibagi berdasarkan hak akses pun, sudah cukup baik dalam proses pelaksanaannya.

Pengerjaan dokumentasi selanjutnya diharapkan *scenario test* dapat dibuat lebih spesifik untuk seitan sistem atau fungsi yang terdapat di LMS Fakultas Ilmu Komputer, Universitas Subang, baik untuk Mahasiswa maupun pihak Akademik. Hal tersebut termasuk dalam pengembangan skenario yang disesuaikan dengan kebutuhan sistem. Jika LMS berkembang maka semakin besar *scope* pengujian yang harus dilakukan serta dokumentasi yang harus dibuat. Lalu disarankan pula untuk di lakukan pengujian dengan metode lain untuk memastikan keamanan di *Learning Management System*, Fakultas Ilmu Komputer, Universitas Subang. Semakin berkembangnya *system* pula semakin rentan terhadap kewanaran. Jadi *Learning Management System*, Fakultas Ilmu Komputer, Universitas Subang agar selalu melakukan *update* dalam hal keamanan sistem.

Fakultas Ilmu Komputer maupun Universitas Subang secara keseluruhan dan peneliti selanjutnya dapat melakukan pengujian menggunakan standar pengujian lain yang terdapat pada Tabel 1. Penggunaan standar lain dapat memperlihatkan hasil *dokumen software testing* yang berbeda sehingga akan menjaga dan meningkatkan kualitas LMS. Selain itu juga dapat melakukan pembuatan *document testing* yang lebih luas yang mencakup dokumen kebutuhan, dokumen arsitektur dan desain, dokumen teknis, dokumen *end user*, dan dokumen pemasaran. Pengujian juga dapat diperluas dengan melibatkan *user* dalam hal ini adalah dosen dan mahasiswa.

REFERENSI

- [1] The Institute of Electrical and Electronics Engineers, IEEE Standard for Software and System Test Documentation. 3 Park Avenue. New York, 2019.
- [2] Systeme Evolutif Limited, IEEE Test Plan Template, 2019.
- [3] P., Adi, "Jurnal Software Testing Aplikasi Website PT. Gramedia Menggunakan Metode Blackbox Pada PT WGS Bandung," *E-Journal Universitas Dianapura*, 2013.

- [4] R. Maulana, et al, "Software Testing pada Aplikasi Website PT Semen Tonasa menggunakan Metode Assessment Vulnerability," *Jurnal INSYPRO (Information System and Processing)*, 2017.
- [5] M. A. H. Anwar, Y. Kurniawan, "Dokumentasi Software Testing Berstandar Ieee 829-2008 Untuk Sistem Informasi Terintegrasi Universitas. Kurawal", *Jurnal Teknologi, Informasi dan Industri*, 2(2), 118-125, 2019.
- [6] B. R. Marbun. Perancangan Proses Pengujian Regresi Terotomatisasi Menggunakan Software Testing Standard ISO/IEC 29119 Pada Proses Pengembangan Perangkat Lunak: Studi Kaus PT XYZ.Universitas Indonesia. 2016
- [7] Dwanoko, Y. S. (2016). Implementasi Software Development Life Cycle (SDLC) Dalam Penerapan Pembangunan Aplikasi Perangkat Lunak. *Jurnal Teknologi Informasi: Teori, Konsep, dan Implementasi*, 7(2), 143003.
- [8] I. Hooda et al., "Software Test Process, Testing Type, and Techniques." *International Journal of Computer Application*, Vol.111-No.13, 2015.
- [9] Vasanthapriyan, S., Tian, J., & Xiang, J. (2017, November). An ontology-based knowledge framework for software testing. In *International Symposium on Knowledge and Systems Sciences* (pp. 212-226). Springer, Singapore.
- [10] Park, J. K. (2018). A Development Method of Safety Critical Software for Nuclear Facilities. In *Proceedings of the Korea Information Processing Society Conference* (pp. 229-231). Korea Information Processing Society.
- [11] Dewi, E. K., & Azhari, S. N. (2012). Analisis Keamanan Sistem Perangkat Lunak. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.
- [12] B. Harsh, "Black Box Testing based on Requirement Analysis and Design Specifications," *International Journal of Computer Applications* (0975 – 8887) .18,18, 2014.
- [13] W. Kumar, "Test Case Generation Based on Use case and Sequence Diagram." *International Journal of Software Engineering (IJSE)*, 2010.
- [14] B. Alkhales, "Effective Bug Tracking Systems.Theories and Implementation," *IOSR Journal of Computer Engineering* ISSN:2278-0661 Volume 4,Issue 6 pp 31-36, 2012.
- [15] B.Beizer, "Software Testing Techniques" *Van Nostrand Reinhold*, 1990.ISBN.0-442-20672- 0, 2006.
- [16] G.McGraw, "Software Security Testing," *IEEE Security and Privacy* 2.2," 2004.