

RANCANG BANGUN APLIKASI SIMULASI TES TOEFL BERBASIS WEBSITE (STUDI KASUS: PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS MATARAM)

(Website-Based TOEFL Test Simulation Application Design (Case Study: Informatics Engineering Department, Mataram University))

Yuli Yana Astuti^{[1]*}, Sri Endang Anjarwani^[1], Budi Irmawati^[1]

^[1]Dept. Informatics Engineering, Mataram University

Jl. Majapahit 62, Mataram, Lombok NTB, INDONESIA

Email: yuliana06082018@gmail.com, [endang, budi-i]@unram.ac.id

Abstract

One measure of a person's English language skills can be seen from the TOEFL test score obtained. TOEFL (Test of English as a Foreign Language) is a test used to measure a person's English language skills. If the TOEFL test score has reached a predetermined standard, it can be used for various selection requirements, such as scholarships and university graduation. Based on the regulation of the Chancellor of the University of Mataram Number 7 of 2021 Article 17 Paragraph 8 Letter g, students who are going to do a thesis trial need to attach a certificate of passing the TOEFL test first as proof that the student has taken the TOEFL test and was declared to have passed with the score determined by their department, including the Informatics Engineering Department. So, in this study, the authors will design and build a website-based TOEFL test simulation application at the Informatics Engineering Department, University of Mataram. This study uses the CodeIgniter (CI) framework and the personal extreme programming (XP) system development method. Testing on this application uses the black box testing method and the Mean Opinion Score (MOS). The result is that the application can run properly according to its respective functionality and get an MOS value of 4.6, showing that the TOEFL test simulation application has good quality to use.

Keywords: TOEFL, CodeIgniter, Personal Extreme Programming, Black Box Testing, Mean Opinion Score

*Corresponding Author

1. PENDAHULUAN

Saat ini, kemampuan berbahasa Inggris sudah mulai diperhitungkan dalam dunia pendidikan maupun dunia kerja. Oleh karena itu, perlu untuk mengikuti kursus agar lebih mahir dalam berbahasa Inggris, yang sesuai dengan penjelasan Pasal 26 Ayat 5 Undang-Undang Nomor 20 Tahun 2003[1]. Salah satu tolak ukur kemampuan berbahasa Inggris seseorang bisa dilihat dari skor tes TOEFL yang diperoleh. Jika skor tes TOEFL sudah mencapai standar yang sudah ditentukan, maka bisa digunakan untuk berbagai syarat seleksi, seperti beasiswa, studi lanjut dalam negeri maupun luar negeri, rekrutmen pegawai, serta sebagai salah satu syarat kelulusan Perguruan Tinggi.

TOEFL adalah suatu tes yang digunakan untuk mengukur kemampuan berbahasa Inggris bagi orang yang menggunakan bahasa Inggris sebagai bahasa keduanya (*non-native*)[2].

Berdasarkan peraturan Rektor Universitas Mataram Nomor 7 Tahun 2021 Pasal 17 Ayat 8 Huruf g,

mahasiswa yang ingin melakukan sidang skripsi perlu menunjukkan atau melampirkan sertifikat kelulusan tes TOEFL terlebih dahulu sebagai bukti bahwa mahasiswa tersebut benar-benar telah melakukan tes TOEFL dan dinyatakan lulus dengan skor yang telah ditentukan oleh program studi atau jurusan masing-masing, termasuk Program Studi Teknik Informatika. Dari hasil wawancara dengan Ketua Program Studi Teknik Informatika Universitas Mataram, saat ini dibutuhkan sebuah aplikasi simulasi tes TOEFL yang diharapkan dapat membantu mahasiswa untuk berlatih agar mendapatkan hasil yang maksimal dalam tes TOEFL yang sebenarnya, mengingat bahwa sertifikat kelulusan tes TOEFL sudah mulai diberlakukan sebagai salah satu syarat mengikuti sidang skripsi.

Oleh karena itu, maka pengembangan sistem aplikasi simulasi tes TOEFL ini menjadi sangat penting. Aplikasi yang dibuat ditujukan kepada seluruh mahasiswa Universitas Mataram sebagai peserta yang ingin melakukan tes TOEFL dan staf tata usaha sebagai

admin dalam pengelolaan data. Aplikasi simulasi tes TOEFL yang akan dibangun memiliki beberapa keunggulan, yaitu melihat dan mengunduh materi, melihat skor tes, serta melihat *report* atau pencapaian berdasarkan beberapa kategori tes TOEFL yang pernah dilakukan. Dengan adanya *report* tersebut, peserta bisa melihat statistik dari hasil tes yang pernah dilakukan dan bisa menilai sendiri bagian mana yang perlu ditingkatkan dan dipelajari kembali untuk mencapai nilai yang diinginkan. Dengan adanya materi juga bisa membantu peserta untuk mempersiapkan diri sebelum melaksanakan tes dan memperbaiki nilai pada kategori yang dirasa kurang. Diharapkan dengan dibangunnya aplikasi simulasi tes TOEFL berbasis *website* di Program Studi Teknik Informatika Universitas Mataram ini dapat meminimalisir waktu dan biaya yang dikeluarkan oleh mahasiswa, juga menambah persiapan diri sebelum mengikuti tes TOEFL yang sesungguhnya.

2. TINJAUAN PUSTAKA

Pada penelitian yang berjudul “Perancangan Dan Pembuatan Aplikasi Simulasi Tes Toefl Berbasis Desktop” yang dilakukan oleh R. Nasser dan P. Saldriani dihasilkan sebuah aplikasi simulasi tes TOEFL berbasis desktop menggunakan metode *Research and Develop* (R&D). Hasil pengujian dari tampilan menu *login*, tes *listening*, tes *reading*, tes *structure*, dan halaman skor tes TOEFL keseluruhannya berjalan sesuai dengan hasil yang diharapkan (*valid*) dan layak untuk digunakan[2]. Pada penelitian yang berjudul “Rekayasa Perangkat Lunak Sistem Ujian TOEFL Menggunakan PHP & MySQL Berbasis *Framework Codeigniter*” yang dilakukan oleh Yustiar Alan Dwiwana dan Achmad Bisri dihasilkan kesimpulan bahwa perbandingan waktu yang dibutuhkan jika menggunakan sistem TOEFL lebih singkat dari cara manual, dan persentase waktu yang dihemat sekitar 92,5% atau 3.440 menit[3].

Pada penelitian yang berjudul “Pembangunan Aplikasi Simulasi Tes TOEFL Berbasis Web” yang dilakukan oleh Zohwan Diraja menghasilkan sebuah aplikasi simulasi tes TOEFL berbasis web yang dapat melaporkan hasil tes secara periodik sesuai dengan kebutuhan *user*[4]. Fitur tersebut dapat membantu *user* mengetahui seberapa besar kemajuan dan kemampuannya dalam berbahasa inggris.

Pada penelitian yang berjudul “Aplikasi TOEFL *Preparation* Berbasis IBT (*Internet Based Test*)” yang dilakukan oleh Bobby, Atiqah, dan Deddy menghasilkan aplikasi TOEFL *preparation* yang dapat membantu mahasiswa untuk melakukan persiapan tes TOEFL dengan mengerjakan soal dan mempelajari

materi-materi yang telah disediakan. Namun, materi yang disediakan perlu ditambah lagi, baik materi dari *e-book* atau dari buku[5].

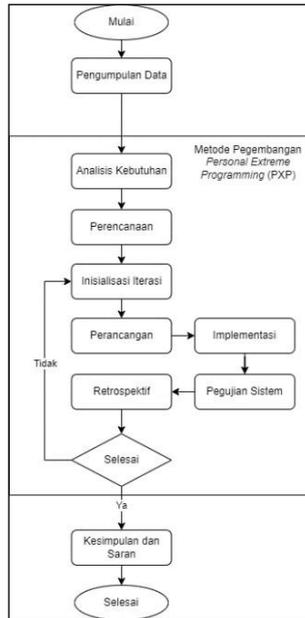
Pada penelitian yang berjudul “Rancang Bangun Aplikasi Tes TOEFL Menggunakan Algoritma *Quick Sort* Berbasis Komputer” yang dilakukan oleh Nuri David dan Yulia Darnita menghasilkan sebuah aplikasi tes TOEFL berbasis komputer yang dapat menampilkan keseluruhan hasil tes yang dilakukan oleh pengguna setelah nilai-nilai tersebut diurutkan. Kekurangan dari aplikasi yang dibuat adalah hanya bisa diakses menggunakan sistem operasi *Windows* dan tidak dapat diakses dari luar jaringan, sehingga perlu dikembangkan untuk dapat dibuat berbasis *web* ataupun *mobile*[6].

Pada penelitian yang berjudul “Pembuatan Aplikasi TOEFL Sebagai Media Pelatihan Bahasa Inggris Berbasis Web” yang dilakukan oleh Yongki, R. Rizal, dan Kurniawan menghasilkan sebuah aplikasi TOEFL berbasis web yang mirip dengan tes TOEFL PBT (*Paper Based Test*). Aplikasi tersebut tidak bisa digunakan sebagai penilaian akhir hasil tes TOEFL dan tidak untuk menggantikan tes yang sebenarnya[7].

Berdasarkan tinjauan pustaka yang telah dipaparkan, peneliti akan merancang dan membangun aplikasi simulasi tes TOEFL berbasis *website* pada Program Studi Teknik Informatika Universitas Mataram yang bertujuan untuk memudahkan mahasiswa program studi Teknik Informatika Universitas Mataram dalam mempersiapkan diri dan melakukan simulasi tes TOEFL sebagai salah satu syarat kelulusan secara gratis dengan menyediakan simulasi tes TOEFL, *history* tes yang pernah dilakukan mahasiswa yang bersangkutan, serta materi pendukung yang dapat membantu mahasiswa mengetahui tips dan trik dalam menjawab soal serta beberapa contoh soal dan penjelasannya.

3. METODE PENELITIAN

Metode yang digunakan dalam pembuatan Aplikasi Simulasi Tes TOEFL Berbasis *Website* ini adalah metode PXP. Metode PXP mempunyai keunggulan dari sisi kecepatan karena fase-fase yang tidak terlalu banyak. Keunggulan lainnya adalah dapat menangani perubahan *requirement* tanpa harus mengulang program. Namun, hal tersebut membuat *developer* harus selalu siap untuk mengerjakan perubahan *requirement* yang terjadi. Pada Gambar 1 berikut menampilkan tahap demi tahap dalam pembuatan aplikasi menggunakan metode PXP.



Gambar 1. Diagram alir penelitian metode PXP

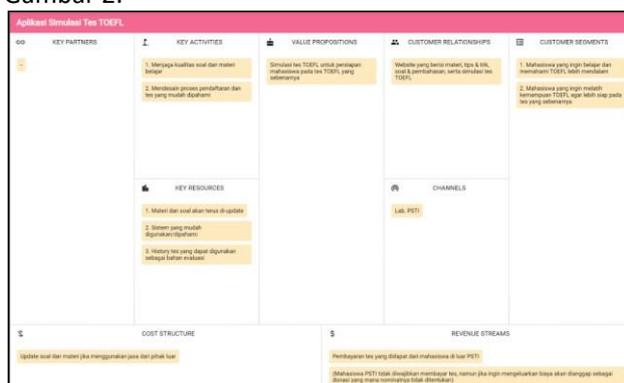
Berikut penjelasan dari setiap tahap yang ada dalam metode PXP pada Gambar 1:

3.1. Pengumpulan Data

Pada tahap ini dilakukan wawancara dengan beberapa pihak yang bersangkutan. Wawancara pertama dilakukan dengan ketua Program Studi Teknik Informatika Universitas Mataram selaku *client* dari sistem yang akan dibuat. Wawancara kedua dilakukan dengan sekretaris Pusat Bahasa dan staff Laboratorium Bahasa Fakultas Keguruan dan Ilmu Pendidikan Universitas Mataram.

3.2. Analisis Kebutuhan

Dari hasil data yang didapatkan, kemudian akan digambarkan ke dalam sebuah proses bisnis. Berikut rancangan *business model canvas* dari aplikasi simulasi tes TOEFL yang akan dibuat yang dapat dilihat pada Gambar 2.



Gambar 2. *Business model canvas* aplikasi simulasi tes TOEFL

Berdasarkan rancangan *business model canvas* pada Gambar 2, target utama dari aplikasi simulasi TOEFL yang akan dibuat adalah mahasiswa Universitas Mataram, terutama mahasiswa program studi Teknik Informatika.

3.3. Perencanaan

Pada tahap ini, kebutuhan *client* yang telah dikumpulkan akan diimplementasikan ke dalam bentuk *user story*. *User story card* yang baik menerapkan model INVEST yaitu *Independent, Negotiable, Valuable, Estimable, Small, dan Testable*. *Independent* berarti tidak ada ketergantungan antar *user story*, *Negotiable* berarti seluruh detail diperoleh melalui kolaborasi, *Valuable* berarti memiliki nilai bagi pengguna, *Estimable* berarti dapat diperkirakan baik dalam segi waktu dan biaya, *Small* berarti ruang lingkupnya tidak kompleks untuk dikembangkan dan diuji, dan *Testable* berarti dapat ditentukan *acceptance criteria*-nya. Sebuah *user story* dapat dikatakan selesai jika *acceptance criteria user story* tersebut telah terpenuhi[8].

3.4. Inisialisasi Iterasi

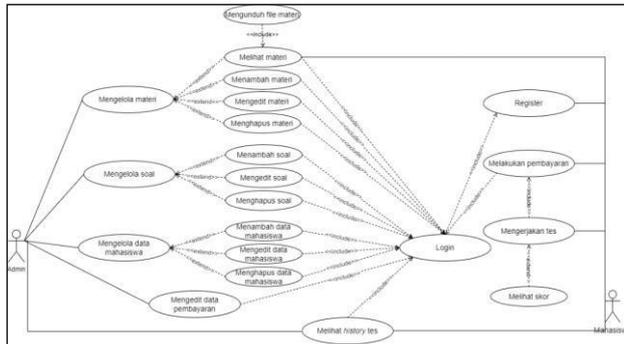
Pada tahap ini ditentukan *user stories* mana yang akan diimplementasikan pada tiap iterasi. Tahap ini meliputi perencanaan estimasi *story* dan pembagian iterasi untuk setiap *story*. *Story point* tiap *user story* dijumlahkan hingga sama dengan nilai *velocity*. Nilai *velocity* dapat diperoleh dari dokumen proyek serupa yang pernah dilakukan, jika tidak ada maka nilai *velocity* bisa ditentukan sendiri oleh pihak pengembang[9]. Pada penelitian Tugas Akhir ini terdapat 24 *story points* dan nilai *velocity* bernilai 12, maka dapat diketahui jumlah iterasinya yaitu $24/12 = 2$ iterasi. Setiap iterasi dalam pembuatan sistem pada penelitian Tugas Akhir ini dapat diselesaikan dalam kurun waktu 12 hari.

3.5. Perancangan

Pada tahap ini dilakukan perancangan atau penggambaran tentang bagaimana sistem yang dikembangkan akan bekerja.

3.5.1. Use case diagram

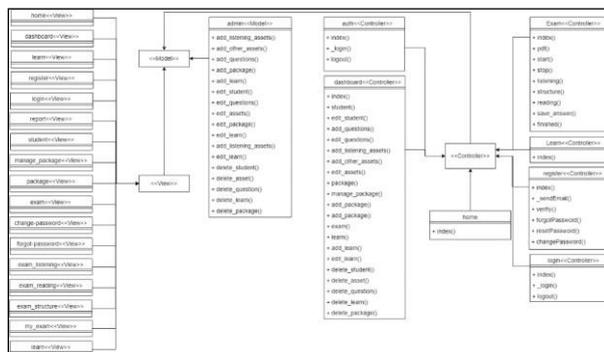
Berikut merupakan *use case diagram* dari aplikasi simulasi tes TOEFL berbasis *website* yang dapat dilihat pada Gambar 3:



Gambar 3. Use case diagram

Pada Gambar 3 dapat dilihat bahwa terdapat dua user yang akan menggunakan aplikasi simulasi tes TOEFL ini, yaitu admin sebagai pengelola sistem dan mahasiswa yang akan melakukan tes TOEFL.

3.5.2. Class diagram



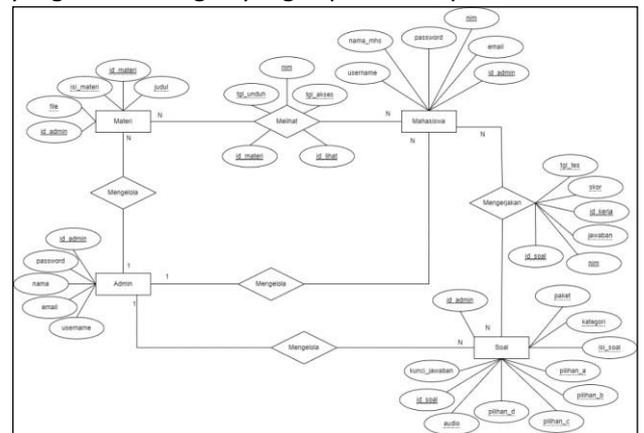
Gambar 4. Class diagram

Pada Gambar 4 dapat dilihat bahwa kerangka kerja yang digunakan dalam pembangunan aplikasi simulasi tes TOEFL berbasis website ini adalah arsitektur Model – View – Controller (MVC). MVC menjadi sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya (Controller).

3.5.3. Entity relationship diagram

Pada rancangan ERD berikut, alur sistem dimulai saat mahasiswa mendaftarkan akun. Saat mendaftar, data akan dimasukkan ke dalam tabel mahasiswa. Tabel mahasiswa memiliki relasi dengan tabel tes, tabel materi, dan tabel admin. Hal tersebut menandakan bahwa jika akun sudah terdaftar, maka mahasiswa bisa mengerjakan tes, melihat materi, dan dikelola juga oleh admin. Admin tidak perlu mendaftarkan akun. Data admin ditambahkan secara manual ke database sehingga admin hanya perlu login untuk masuk ke sistem. Tabel admin memiliki relasi dengan tabel mahasiswa, tes, dan materi. Admin bisa melakukan pengelolaan terhadap mahasiswa, tes, dan materi.

Berikut merupakan Entity Relationship Diagram (ERD) dari aplikasi simulasi tes TOEFL berbasis website yang akan dibangun yang dapat dilihat pada Gambar 5:



Gambar 5. Entity relationship diagram

3.6. Implementasi

Tahap selanjutnya adalah melakukan implementasi rancangan sistem ke dalam bentuk sistem aplikasi. Proses coding pembuatan sistem dilakukan oleh peneliti sendiri sebagai pengembang tunggal dan aplikasi yang akan dibangun berbasis website. Pembuatan aplikasi yang akan dibangun menggunakan tools berupa framework CodeIgniter 3 dan Bootstrap. Untuk database yang digunakan yaitu MySQL.

3.7. Pengujian Sistem

Tahap selanjutnya adalah melakukan pengujian sistem. Metode yang digunakan adalah black box testing atau pengujian fungsionalitas untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program. Selain menggunakan black box testing, pengujian juga dilakukan kepada pengguna yang terlibat yaitu mahasiswa sebagai pengguna utama menggunakan metode MOS (Mean Opinion Score) untuk melakukan penilaian apakah sistem sudah layak digunakan atau belum.

3.8. Retrospektif

Tahap retrospektif merupakan tahap akhir dari iterasi yang dilakukan. Pada tahap ini, dilakukan verifikasi apakah waktu pengerjaan sistem sesuai atau tidak dengan waktu yang telah diestimasikan. Jika waktu lebih dari estimasi, maka perlu dilakukan identifikasi mengenai masalah dan solusi untuk pengerjaan iterasi selanjutnya. Untuk menandai akhir dari penelitian, maka semua requirements sistem harus sudah terpenuhi.

4. HASIL DAN PEMBAHASAN

4.1. Implementasi Database

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 K B	-
exam_assets	Browse Structure Search Insert Empty Drop	130	InnoDB	utf8mb4_general_ci	42.0 K B	-
exam_data	Browse Structure Search Insert Empty Drop	50	InnoDB	utf8mb4_general_ci	16.0 K B	-
exam_history	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 K B	-
exam_package	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 K B	-
exam_questions	Browse Structure Search Insert Empty Drop	280	InnoDB	utf8mb4_general_ci	112.0 K B	-
learn	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 K B	-
student	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 K B	-
student_token	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 K B	-
9 tables	Sum	474	InnoDB	utf8mb4_general_ci	272.0 K B	0 B

Gambar 6. Implementasi database

Gambar 6 merupakan implementasi database pada aplikasi simulasi tes TOEFL berbasis website, dimana terdapat 8 tabel, yaitu admin, exam_assets, exam_data, exam_history, exam_package, exam_questions, learn, dan student.

4.2. Implementasi Class

Berdasarkan perancangan class diagram yang telah dilakukan sebelumnya, implementasi class yang dibuat sudah sesuai dengan perancangan tersebut. Class pada sistem dibangun menggunakan bahasa pemrograman PHP dengan framework CodeIgniter. Class tersebut menghasilkan interface yang akan berinteraksi langsung dengan pengguna dan representasi database.

4.3. Implementasi Interface

Interface sistem dibangun menggunakan template bootstrap dan memanfaatkan berbagai fitur dari bootstrap.

4.3.1. Interface mahasiswa

Setelah berhasil melakukan login, mahasiswa akan diarahkan ke halaman utama. Pada halaman utama mahasiswa dapat memilih menu untuk memulai tes, melihat history tes, atau melihat daftar materi pembelajaran. Gambar 7 berikut menunjukkan tampilan halaman utama.



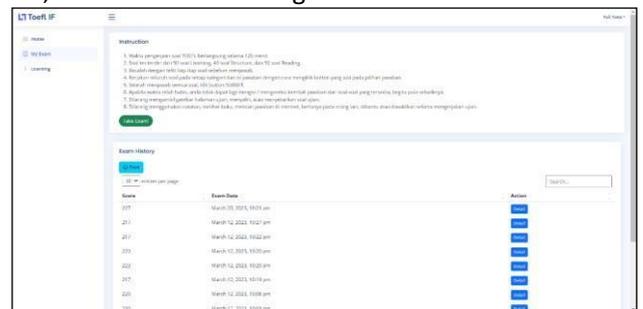
Gambar 7. Tampilan halaman utama

Gambar 8 berikut menunjukkan implementasi halaman tes. Pada halaman ini mahasiswa dapat mengerjakan soal-soal yang tersedia dan mengklik button Submit setelah semua soal terjawab.



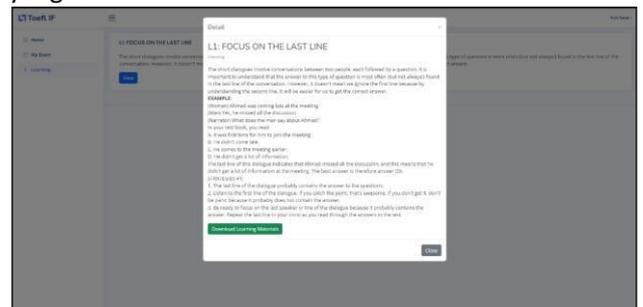
Gambar 8. Implementasi halaman tes

Gambar 9 berikut menunjukkan implementasi halaman history tes, yang berisi waktu dan skor yang diperoleh. Mahasiswa bisa mengklik button Detail untuk melihat detail skor. Untuk mengunduh history tes, mahasiswa bisa mengklik button Print.



Gambar 9. Implementasi halaman history tes

Gambar 10 berikut menunjukkan implementasi halaman detail materi. Pada halaman ini mahasiswa dapat melihat isi materi serta mengunduh file materi yang tersedia.



Gambar 10. Implementasi halaman detail materi

4.3.2. Interface admin

Setelah berhasil melakukan login, admin akan diarahkan ke halaman dashboard. Pada halaman ini admin dapat melihat info mengenai total mahasiswa terdaftar, total paket soal, total tes, total materi, dan diagram rata-rata skor tes TOEFL dari setiap kategori. Gambar 11 berikut menunjukkan implementasi halaman dashboard.



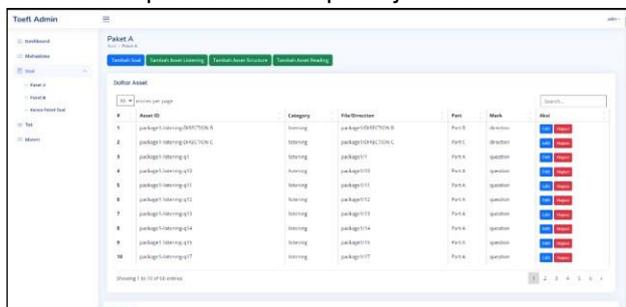
Gambar 11. Implementasi halaman *dashboard*

Gambar 12 berikut menunjukkan implementasi halaman kelola data mahasiswa. Pada halaman ini admin dapat mengubah dan menghapus data mahasiswa.



Gambar 12. Implementasi halaman kelola mahasiswa

Gambar 13 berikut menunjukkan implementasi halaman kelola soal. Pada halaman ini admin dapat menambah, mengubah, dan menghapus soal serta asset soal seperti audio dan petunjuk soal.



Gambar 13. Implementasi halaman kelola soal

Gambar 14 berikut menunjukkan implementasi halaman kelola materi. Pada halaman ini admin dapat menambah, mengubah, dan menghapus data materi.



Gambar 14. Implementasi halaman kelola materi

4.4. Refactoring

Pada pembuatan aplikasi simulasi tes TOEFL berbasis *website* ini, proses *refactoring* dilakukan dengan memperbaiki struktur kode program. Struktur kode yang dimaksud yaitu memisahkan kode

berdasarkan fungsionalitasnya masing-masing serta menambahkan dokumentasi atau komentar pada beberapa fungsi agar lebih mudah dimengerti oleh pengembang selanjutnya.

4.5. Pengujian Sistem

Proses pengujian sistem dilakukan dengan menggunakan metode *black box testing* dan *Mean Opinion Score (MOS)*.

4.5.1. Black box testing

Pengujian menggunakan metode *black box testing* dilakukan dengan cara menguji fungsionalitas yang sudah diimplementasikan pada aplikasi yang sudah dibuat. Hal ini bertujuan untuk menguji apakah hasil *input* dan *output* aplikasi sudah sesuai dengan apa yang diharapkan atau belum. Dari 7 fitur yang telah diuji, semua skenario mengembalikan hasil "Diterima". Pengujian dilakukan dengan Bapak I Gede Pasek Suta Wijaya, selaku ketua Program Studi Teknik Informatika Universitas Mataram. Dapat disimpulkan bahwa fungsionalitasnya telah berjalan dengan baik tanpa adanya kesalahan.

4.5.2. Mean Opinion Score (MOS)

Tahapan pengujian selanjutnya adalah metode *Mean Opinion Score (MOS)*, dimana 25 responden diminta untuk mengisi kuesioner menggunakan *google form* untuk menilai fungsionalitas dari setiap fitur yang ada pada aplikasi sistem simulasi tes TOEFL berbasis *website* yang telah dibangun. Hasil penilaian tersebut kemudian akan dikumpulkan dan dihitung untuk mendapatkan kesimpulan seberapa layak aplikasi dapat digunakan. Kuesioner yang diberikan terdiri dari 7 pertanyaan sebagai berikut:

- Apakah tampilan aplikasi simulasi tes TOEFL yang dibangun mudah untuk dipahami?
- Apakah fitur ujian (tes) mudah untuk dioperasikan?
- Apakah fitur materi mudah untuk dioperasikan?
- Apakah informasi detail skor pada *exam history* dapat lebih memudahkan pengguna dalam memahami bagian-bagian tes TOEFL?
- Apakah informasi yang disampaikan pada aplikasi sudah cukup jelas?
- Secara keseluruhan, apakah aplikasi simulasi tes TOEFL yang dibangun mudah untuk digunakan?
- Apakah aplikasi yang dibangun sudah sesuai dengan kebutuhan pengguna?

Responden akan diminta untuk menjawab kuesioner dengan pilihan jawaban sebagai berikut:

A: Sangat Mudah/Sangat Setuju/Sangat Jelas

B: Mudah/Setuju/Jelas

C: Cukup

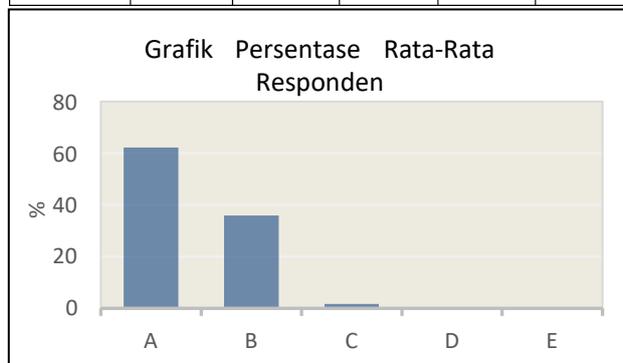
D: Tidak Mudah/Tidak Setuju/Tidak Jelas

E: Sangat Tidak Mudah/Sangat Tidak Setuju/Sangat Tidak Jelas

Pada TABEL I terdapat hasil pengujian dengan menyebarkan kuisisioner kepada 25 pengguna (mahasiswa). Hasil rata-rata yang didapatkan untuk jawaban A adalah 62,28%, jawaban B 36%, dan jawaban C 1,71%. Hasil pengujian tersebut juga divisualisasikan ke dalam bentuk diagram yang dapat dilihat pada Gambar 15.

TABEL I. HASIL PENGUJIAN KUISISIONER KEPADA PENGGUNA (MAHASISWA)

Pertanyaan	A		B		C		D		E	
	Total	%	Total	%	Total	%	Total	%	Total	%
Pertanyaan 1	15	60	10	40	0	0	0	0	0	0
Pertanyaan 2	14	56	10	40	1	4	0	0	0	0
Pertanyaan 3	16	64	9	36	0	0	0	0	0	0
Pertanyaan 4	19	76	6	24	0	0	0	0	0	0
Pertanyaan 5	14	56	9	36	2	8	0	0	0	0
Pertanyaan 6	17	68	8	32	0	0	0	0	0	0
Pertanyaan 7	14	56	11	44	0	0	0	0	0	0
Rata-rata	62.28		36		1.71		0		0	



Gambar 15. Grafik persentase rata-rata responden pengguna aplikasi

Setelah mendapatkan hasil persentase untuk keseluruhan pertanyaan, kemudian menentukan skala yang akan digunakan untuk penentuan nilai MOS.

Berikut adalah tabel keterangan skala yang akan digunakan:

TABEL II. SKALA MOS

MOS	Keterangan	Bobot Nilai	Kualitas
A	Sangat Mudah/Sangat Setuju/Sangat Jelas	5	Excellent
B	Mudah/Setuju/Jelas	4	Good
C	Cukup	3	Fair
D	Tidak Mudah/Tidak Setuju/Tidak Jelas	2	Poor
E	Sangat Tidak Mudah/Sangat Tidak Setuju/Sangat Tidak Jelas	1	Bad

TABEL III. HASIL PERHITUNGAN MOS

Pertanyaan	A (5)	B (4)	C (3)	D (2)	E (1)	Mean
Pertanyaan 1	15	10	0	0	0	4,6
Pertanyaan 2	14	10	1	0	0	4,52
Pertanyaan 3	16	9	0	0	0	4,64
Pertanyaan 4	19	6	0	0	0	4,76
Pertanyaan 5	14	9	2	0	0	4,48
Pertanyaan 6	17	8	0	0	0	4,68
Pertanyaan 7	14	11	0	0	0	4,56
MOS (Mean Opinion Score)						4,6

Berdasarkan hasil perhitungan MOS pada TABEL III yang dilakukan kepada 25 responden, didapatkan nilai MOS sebesar 4,6. Berdasarkan skala yang telah ditentukan pada TABEL II, nilai yang didapatkan menunjukkan bahwa aplikasi simulasi tes TOEFL berbasis *website* yang dibangun memiliki kualitas yang baik untuk digunakan (hasil uji MOS > 4,0).

4.6. Retrospektif

Tahap terakhir dalam pembuatan sistem simulasi tes TOEFL berbasis *website* adalah retrospektif. Setelah pengujian selesai dan diterima oleh pihak program studi Teknik Informatika Universitas Mataram, kemudian dilakukan analisis terhadap jalannya setiap fase pengembangan topik *user story* serta kesesuaian estimasi waktu pengerjaan pada tiap iterasi. Hasilnya adalah pengerjaan *user story* pada iterasi pertama dan kedua berhasil diselesaikan sesuai dengan estimasi yang ditentukan, yaitu 24 hari dengan pengerjaan masing-masing iterasi adalah 12 hari.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian dan pembuatan sistem yang dilakukan, dapat ditarik beberapa kesimpulan antara lain sebagai berikut:

- Pembuatan aplikasi simulasi tes TOEFL berbasis *website* menggunakan metode Personal Extreme Programming untuk mengejar waktu

- rilis dalam jangka waktu pendek dapat berjalan dengan efektif hingga selesai sesuai dengan estimasi waktu yang direncanakan yaitu 24 hari karena adanya perencanaan iterasi dan retrospektif pada setiap akhir iterasi yang dilakukan sesuai dengan tahapan metode PXP.
- b. Perancangan dan pembuatan aplikasi simulasi tes TOEFL berbasis website dalam penelitian ini dapat membantu mahasiswa untuk mempersiapkan diri sebelum mengerjakan tes TOEFL dan sudah sesuai dengan kebutuhan pengguna, dimana terbukti dari hasil kuesioner yang telah dilakukan kepada 25 pengguna/mahasiswa dengan hasil rata-rata yang menjawab sangat mudah/sangat jelas/sangat setuju adalah 62,28%, mudah/jelas/setuju adalah 36%, dan cukup adalah 1,71%.
- c. Berdasarkan hasil kuesioner yang dilakukan kepada 25 pengguna, didapatkan nilai MOS sebesar 4,6 yang menunjukkan bahwa aplikasi simulasi tes TOEFL berbasis website yang dibangun memiliki kualitas yang baik untuk digunakan (hasil uji MOS > 4,0).

5.2. Saran

Setelah melakukan penelitian dan pengembangan sistem, terdapat beberapa saran yang peneliti dapat sampaikan untuk pengembangan selanjutnya antara lain sebagai berikut:

- a. Untuk pengembangan dan penelitian selanjutnya, peneliti bisa mengoptimalkan UI/UX sesuai dengan perkembangan teknologi sehingga sistem lebih ramah terhadap pengguna.
- b. Untuk pengembangan dan penelitian selanjutnya, peneliti bisa menambahkan fitur pembayaran dan verifikasi secara online agar memudahkan pengguna dan mengoptimalkan penggunaan sistem.

DAFTAR PUSTAKA

- [1] R. Reksi, "PERANCANGAN PEMBUATAN APLIKASI SIMULASI UJIAN TOEFL BERBASIS ANDROID," vol. 3, no. 2017, pp. 54–67, 2020, [Online]. Available: <http://repositorio.unan.edu.ni/2986/1/5624.pdf>.
- [2] R. Nasser and P. Saldriani, "Perancangan Dan Pembuatan Aplikasi Simulasi Toefl Berbasis Desktop," *Pros. Semant.*, vol. 2, no. 2011, pp. 184–191, 2019, [Online]. Available: <http://journal.uncp.ac.id/index.php/semantik/article/view/1514>.
- [3] Y. A. Dwihana and A. Bisri, "Rekayasa Perangkat Lunak Sistem Ujian TOEFL Menggunakan PHP & MySQL Berbasis Framework CodeIgniter," *J. CoreIT J. Has. Penelit. Ilmu Komput. dan Teknol. Inf.*, vol. 4, no. 1, p. 14, 2018, doi: 10.24014/coreit.v4i1.5911.
- [4] Zowhan Diraja, "Pembangunan aplikasi simulasi tes toefl berbasis web," no. 9, 2014.
- [5] B. P. Rizky, A. N. Asri, and D. K. Purwoko Aji, "Aplikasi Toefl Preparation Berbasis Ibt (Internet Based Test)," *J. Inform. Polinema*, vol. 1, no. 3, p. 13, 2017, doi: 10.33795/jip.v1i3.107.
- [6] N. D. M. Veronica and Y. Darnita, "Rancang Bangun Aplikasi Tes Toefl Menggunakan Algoritma Quick Sort Berbasis Komputer," *Pseudocode*, vol. 2, no. 2, pp. 89–97, 2015, doi: 10.33369/pseudocode.2.2.89-97.
- [7] Y. Y. Marbun, R. R. Isnanto, and K. T. Martono, "Pembuatan Aplikasi TOEFL sebagai Media Pelatihan Bahasa Inggris Berbasis Web," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 1, p. 83, 2016, doi: 10.14710/jtsiskom.4.1.2016.83-92.
- [8] R. A. Azdy and A. Rini, "Penerapan Extreme Programming dalam Membangun Aplikasi Pengaduan Layanan Pelanggan (PaLaPa) pada Perguruan Tinggi," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 2, p. 197, 2018, doi: 10.25126/jtiik.201852658.
- [9] Fabiana Meijon Fadul, "PERSONAL EXTREME PROGRAMMING DENGAN PENENTUAN PRIORITAS MOSCOW," pp. 15–29, 2019, [Online]. Available: <https://eprints.umm.ac.id/44555/>.