

Rancang Bangun Aplikasi Simulasi Tes TOEFL Berbasis *Website* (Studi Kasus: Program Studi Teknik Informatika Universitas Mataram)

(*Website-Based TOEFL Test Simulation Design (Case Study: Informatics Engineering
Departement, 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, [ending, budi-i]@unram.ac.id

Kemampuan berbahasa inggris sudah mulai diperhitungkan dalam dunia pendidikan maupun dunia kerja. Salah satu tolak ukur kemampuan berbahasa inggris seseorang bisa dilihat dari skor tes TOEFL yang diperoleh. TOEFL (*Test of English as a Foreign Language*) adalah tes yang digunakan untuk mengukur kemampuan berbahasa inggris bagi yang menggunakan bahasa inggris sebagai bahasa keduanya. Jika skor tes TOEFL sudah mencapai standar yang sudah ditentukan, maka bisa digunakan untuk berbagai syarat seleksi, seperti beasiswa dan syarat kelulusan perguruan tinggi. Berdasarkan peraturan Rektor Universitas Mataram Nomor 7 Tahun 2021 Pasal 17 Ayat 8 Huruf g, mahasiswa yang ingin melakukan sidang skripsi perlu melampirkan sertifikat kelulusan tes TOEFL terlebih dahulu sebagai bukti bahwa mahasiswa tersebut 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. Sehingga pada penelitian ini penulis akan merancang dan membangun aplikasi simulasi tes TOEFL berbasis *website* pada Program Studi Teknik Informatika Universitas Mataram. Penelitian ini menggunakan *framework CodeIgniter* (CI) dan metode pengembangan sistem *personal extreme programming* (PXP). Pengujian pada aplikasi ini menggunakan metode *black box testing* dan *Mean Opinion Score* (MOS). Hasilnya adalah aplikasi dapat berjalan dengan baik sesuai dengan fungsionalitasnya masing-masing dan mendapatkan nilai MOS sebesar 4,6 dimana nilai tersebut menunjukkan aplikasi

Kata kunci: TOEFL, Program Studi Teknik Informatika, *CodeIgniter*, *Personal Extreme Programming*, *black box testing*.

I. PENDAHULUAN

2.1 Latar Belakang

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 (*Test of English as a Foreign Language*) 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.

Berdasarkan beberapa penjelasan di atas, penulis akan merancang dan membangun sebuah aplikasi simulasi tes TOEFL di Program Studi Teknik Informatika Universitas Mataram berbasis *website*. 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.

2.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dipaparkan, rumusan masalah yang dapat diambil adalah:

1. Bagaimana cara membuat desain mekanisme evaluasi diri bagi user mengenai perkembangan kemampuan berbahasa Inggris melalui aplikasi simulasi tes TOEFL berbasis *website* yang akan dibangun?
2. Bagaimana merancang dan membangun aplikasi simulasi tes TOEFL berbasis *website* di Program Studi Teknik Informatika Universitas Mataram?

2.3 Tujuan

Tujuan yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Merancang sebuah aplikasi simulasi tes TOEFL yang dapat membantu mahasiswa Program Studi Teknik Informatika Universitas Mataram dalam mempersiapkan tes TOEFL yang asli.
2. Membangun aplikasi simulasi tes TOEFL berbasis *website* di Program Studi Teknik Informatika Universitas Mataram yang berisi informasi seputar tes TOEFL dan simulasi tes TOEFL.

II. TINJAUAN PUSTAKA

2.4 Dasar Teori

2.1.1 TOEFL

TOEFL adalah singkatan dari *Test Of English as a Foreign Language*. TOEFL merupakan sebuah *proficiency*

test, yaitu tes untuk menguji kemampuan seseorang dalam penguasaan bahasa Inggris tanpa dikaitkan dengan proses belajar mengajar[3]. TOEFL sendiri ada tiga jenis, yaitu PBT (*Paper Based Test*), CBT (*Computer Based Test*), dan IBT (*Internet Based Test*).

2.1.2 CodeIgniter

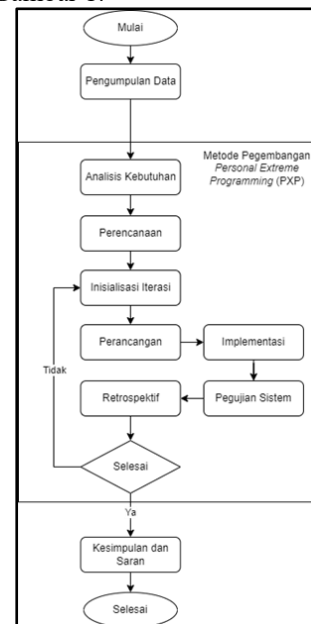
CodeIgniter adalah *framework web* untuk bahasa pemrograman PHP yang dirancang oleh Rick Ellis pada tahun 2006. *CodeIgniter* menganut pola desain atau arsitektur *Model-View-Controller* (MVC) yang memisahkan bagian kode untuk penanganan proses bisnis dengan bagian kode untuk keperluan presentasi (tampilan). Pola ini memungkinkan para *developer web* untuk mengerjakan aplikasi berbasis *web* secara bersamaan (*teamwork*) dan lebih berfokus pada bagiannya masing-masing tanpa mengganggu bagian yang lain[4].

2.1.3 Personal Extreme Programming (XP)

Metode *Personal Extreme Programming* (XP) merupakan metode hasil pengembangan dari metode *Extreme Programming* (XP). Ketika *software developer* hanya satu orang saja, maka metode XP tidak dapat digunakan. Metode XP memudahkan *software developer* dalam melakukan pengembangan perangkat lunak, karena waktu siklus hidup yang cepat serta fase-fase yang tidak terlalu banyak[5].

III. METODOLOGI

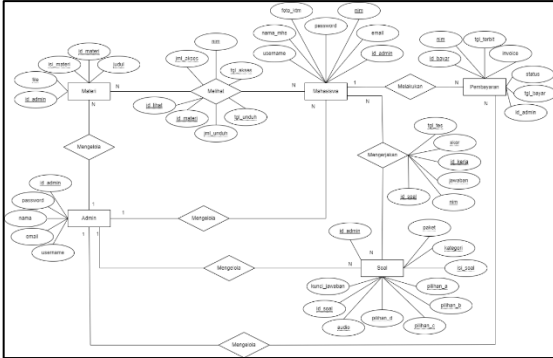
Pada penelitian Tugas Akhir ini menggunakan metode pengembangan sistem *Personal Extreme Programming* (XP). Tahapan pada metode XP dapat dilihat pada Gambar 1:



Gambar 1. Diagram alir penelitian

3.5.3 Entity Relationship Diagram

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*

2.10 Implementasi

Tahap selanjutnya adalah melakukan implementasi rancangan sistem ke dalam bentuk sistem aplikasi. Proses *coding* pembuatan sistem dilakukan oleh penulis 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.

2.11 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.

2.12 Retrospektif

Tahap *retrospective* merupakan tahap akhir dari iterasi yang dilakukan. Pada tahap ini, penulis sebagai pengembang akan melakukan verifikasi apakah waktu yang dibutuhkan untuk menyelesaikan 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.

IV. HASIL DAN PEMBAHASAN

4.1 Implementasi Database

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 K18	-
exam_assets	Insert Empty Drop	158	InnoDB	utf8mb4_general_ci	46.9 K18	-
exam_data	Insert Empty Drop	58	InnoDB	utf8mb4_general_ci	16.0 K18	-
exam_history	Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 K18	-
exam_package	Insert Empty Drop	288	InnoDB	utf8mb4_general_ci	112.0 K18	-
learn	Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 K18	-
student	Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 K18	-
student_token	Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 K18	-
9 tables	Sum	474	InnoDB	utf8mb4_general_ci	272.0 K18	0.8

Gambar 6. Implementasi *database*

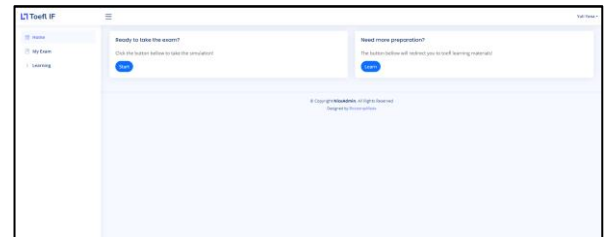
Gambar 6 merupakan implementasi *database* pada aplikasi simulasi tes TOEFL berbasis *website*, dimana terdapat 8 tabel, yaitu admin, exam_assets, exam_data, exam_history, exam_package, exam_questions, learn, dan student.

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*.

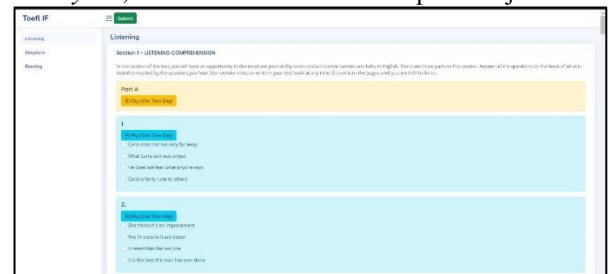
4.3 Implementasi Interface

Berikut merupakan implementasi *interface* dari aplikasi simulasi tes TOEFL berbasis *website* yang telah dibuat:



Gambar 7. Halaman utama

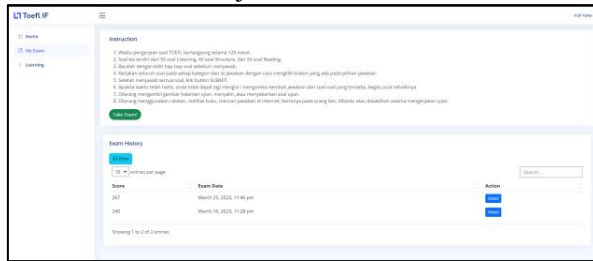
Pada Gambar 7 terdapat implementasi halaman utama bagi mahasiswa. Pada halaman ini mahasiswa dapat memilih menu untuk memulai tes, melihat *history* tes, atau melihat daftar materi pembelajaran.



Gambar 8. Halaman tes

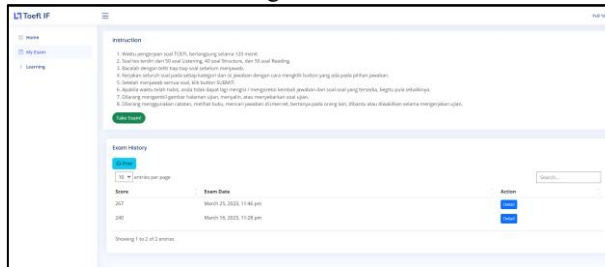
Pada Gambar 8 terdapat implementasi halaman tes. Pada halaman ini mahasiswa dapat mengerjakan

soal-soal yang tersedia dan mengklik *button* Submit setelah semua soal terjawab.



Gambar 9. Halaman *history* tes

Pada Gambar 9 terdapat 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 10. Halaman detail materi

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



Gambar 11. Halaman *dashboard*

Pada Gambar 11 terdapat implementasi halaman *dashboard* admin. 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.

No	Nama	NIM	Email	Status Aktif	Rijalita Ujian	Aksi
1	Mahasiswa 1	12345678	ah@ip.com	Normal	Belum melakukan uji coba	Detail Edit Hapus
2	Darya Pasha	12345679	h@ip.com	Normal	Belum melakukan uji coba	Detail Edit Hapus
3	Ujiya Pasha	12345680	u@ip.com	Normal	Belum melakukan uji coba	Detail Edit Hapus
4	Yul Yana	12345681	y@ip.com	Normal	Rijalita ujian ipa	Detail Edit Hapus

Gambar 12. Halaman kelola mahasiswa

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

No	Aspek ID	Category	Kategori	Part	Mark	Aksi
1	package listening DIRECTOULA	listening	package DIRECTOULA	Part A	10	Detail Edit Hapus
2	package listening DIRECTOULA	listening	package DIRECTOULA	Part B	10	Detail Edit Hapus
3	package listening DIRECTOULA	listening	package DIRECTOULA	Part C	10	Detail Edit Hapus
4	package listening G1	listening	package G1	Part A	10	Detail Edit Hapus
5	package listening G1	listening	package G1	Part B	10	Detail Edit Hapus

Gambar 13. Halaman kelola soal

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

No	Materi	Detail	Status Aktif	Rijalita Ujian	Aksi
1	Materi 1	Detail materi 1	Normal	Belum melakukan uji coba	Detail Edit Hapus
2	Materi 2	Detail materi 2	Normal	Belum melakukan uji coba	Detail Edit Hapus

Gambar 14. Halaman kelola materi

Pada Gambar 14 terdapat implementasi halaman kelola materi. Pada halaman ini admin dapat menambah, mengubah, dan menghapus data 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 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 diuj, semua scenario 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:

1. Apakah tampilan aplikasi simulasi tes TOEFL yang dibangun mudah untuk dipahami?
2. Apakah fitur ujian mudah untuk dioperasikan?
3. Apakah fitur materi mudah untuk dioperasikan?
4. Apakah informasi detail skor pada *exam history* dapat lebih memudahkan pengguna dalam memahami bagian-bagian tes TOEFL?
5. Apakah informasi yang disampaikan pada aplikasi sudah cukup jelas?
6. Secara keseluruhan, apakah aplikasi simulasi tes TOEFL yang dibangun mudah digunakan?
7. 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

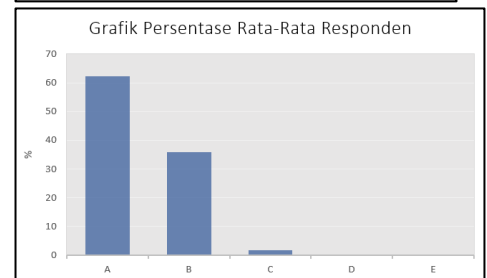
E: Sangat Tidak Mudah/Sangat Tidak Setuju/Sangat Tidak Jelas

Pada Tabel 1 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 1. 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 2. Skala *Mean Opinion Score*

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

Tabel 3. Hasil perhitungan *Mean Opinion Score*

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 (<i>Mean Opinion Score</i>)						4.6

Berdasarkan hasil perhitungan MOS pada Tabel 3 yang dilakukan kepada 25 responden, didapatkan nilai MOS sebesar 4,6. Berdasarkan skala yang telah ditentukan pada Tabel 2, 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 waktu pengerjaan pada iterasi pertama dan kedua lebih lama dari waktu estimasi yang telah ditentukan pada tahap perencanaan, terutama pada halaman admin dimana pada halaman admin terdapat banyak pengelolaan data yang dilakukan. Untuk mengatasi hal tersebut, pengembang dapat meningkatkan *skill programming* dan lebih mendalami pengetahuan mengenai *framework* yang digunakan untuk memudahkan dalam membuat halaman *website* yang akan dibuat, terutama halaman admin.

V. KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

1. Waktu pembuatan aplikasi simulasi tes TOEFL berbasis *website* atau waktu pelaksanaan iterasi pada implementasinya ternyata tidak sesuai dengan estimasi (*iteration planning*) dikarenakan terdapat beberapa perubahan tampilan dan tambahan entitas juga atribut yang membuat pengerjaan menjadi lebih lama, terutama pada bagian admin. Perbedaan antara *iteration planning* dengan pelaksanaan merupakan kewajaran pada metode pengembangan perangkat lunak *personal extreme programming* yang merupakan bagian dari *Agile Method*.
2. 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%.

3. 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:

1. Untuk pengembangan dan penelitian selanjutnya, peneliti bisa mengoptimalkan UI/UX sesuai dengan perkembangan teknologi sehingga sistem lebih ramah terhadap pengguna.
2. 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] A. L. Washliyah and M. Toefl, "Analisis kemampuan bahasa Inggris mahasiswa program pascasarjana universitas muslim nusantara al washliyah melalui," 2022.
- [4] M. F. Setiadi, "Tutorial CodeIgniter Untuk Pemula," *Angew. Chemie Int. Ed.* 6(11), 951–952., p. 63, 1967.
- [5] T. Sandika *et al.*, "Information System Design of River Water Quality in Lampung Province through Personal Extreme Programming Method Rancang Bangun Sistem Informasi Kualitas Air Wilayah Sungai di Provinsi Lampung dengan Metode Personal Extreme Programming," *J. Ilm. ESAI*, vol. 8, no. 2, 2014.

- [6] 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.
- [7] Fabiana Meijon Fadul, "PERSONAL EXTREME PROGRAMMING DENGAN PENENTUAN PRIORITAS MOSCOW," pp. 15–29, 2019, [Online]. Available: <https://eprints.umm.ac.id/44555/>.