

DESAIN DAN IMPELEMENTASI APLIKASI SIMULASI TOEFL MENGGUNAKAN ALGORITA RANDOM NUMBER GENERATOR BERBASIS WEB

Muh. Istikmal Husain¹, Gunawan^{*2}, Budi Wijaya R. S.³

^{1,2,3}Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Halu Oleo, Kendari

Email: ¹istikmal.husain@gmail.com, ²gunawan@aho.ac.id, ³budiwijaya@aho.ac.id
^{*}Penulis Korespondensi

Abstrak

Penelitian ini bertujuan untuk menghasilkan sebuah sistem aplikasi simulasi TOEFL berbasis web dengan menggunakan algoritma RNG dalam mengacak soal. Penelitian dilakukan dengan 3 skenario berbeda pada jenis tes berbeda (listening, reading, dan structure). Skenario pertama dilakukan sebanyak 10 kali percobaan untuk setiap jenis tes dengan tujuan menguji konsistensi sekaligus kualitas acak dari algoritma yang digunakan. Skenario pengujian kedua dilakukan dengan tujuan menguji kualitas pengacakan jika terdapat banyak data pada bank soal yang tersimpan di database. Skenario pengujian ketiga dilakukan untuk menguji hasil pengacakan soal jika terdapat 2 user dengan time (waktu akses) yang saman namun Id User berbeda. Proses pengacakan dilakukan melalui aplikasi web yang telah dirancang sebelumnya dan menggunakan persamaan MCG yaitu salah satu metode pada algoritma RNG. Hasil penelitian yang telah dilakukan, menunjukkan bahwa penggunaan algoritma RNG menggunakan metode MCG dapat diterapkan dengan baik pada sistem simulasi TOEFL. Misal pada hasil pengujian skenario pertama menunjukkan bahwa pada setiap percobaan, urutan soal yang dihasilkan berbeda-beda. Sebagai contoh, pada tes structure soal dengan nomor tertentu dapat muncul pada posisi ke-18 pada percobaan pertama, tetapi pada percobaan berikutnya dapat bergeser ke posisi ke-27 atau 32; Hasil pengujian juga didapatkan bahwa meskipun kedua pengguna memiliki waktu akses yang sama, perbedaan nilai Id User menyebabkan nilai seed awal berbeda. Perbedaan kecil pada nilai seed ini menghasilkan urutan soal yang berbeda secara signifikan.

Kata Kunci: TOEFL, algoritma, RNG, tes, sistem simulasi.

Abstract

This research aims to develop a web-based TOEFL simulation application system using the RNG algorithm to randomize questions. The study was conducted using three different scenarios for different test types (listening, reading, and structure). The first scenario involved 10 trials for each test type to test the consistency and randomness of the algorithm used. The second test scenario was conducted to assess the quality of randomization when there was a large amount of data in the question bank stored in the database. The third test scenario was conducted to examine the results of randomization when there were two users with the same access time but different user IDs. The randomization process was carried out through a previously designed web application and used the MCG equation, one of the methods in the RNG algorithm. The results of the research conducted indicate that the use of the RNG algorithm using the MCG method can be applied well to the TOEFL simulation system. For example, the results of the first scenario test showed that in each trial, the resulting question order was different. For example, in the structure test, a question with a certain number may appear in position 18 on the first trial, but in subsequent trials it may shift to position 27 or 32; The test results also found that even though both users had the same access time, differences in User ID values caused different initial seed values. This small difference in seed values resulted in a significantly different question order.

Keywords: TOEFL, algorithm, RNG, test, simulation system.

1. PENDAHULUAN

Kemampuan berbahasa Inggris merupakan keterampilan penting di dunia pendidikan maupun profesional. Test of English as a Foreign Language (TOEFL) adalah salah satu standar yang dipakai guna mengukur keterampilan bahasa Inggris seseorang (Permatasari, 2024). TOEFL merupakan

tes yang dirancang untuk mengevaluasi keterampilan berbahasa Inggris, khususnya untuk individu yang menjadikan bahasa Inggris sebagai bahasa kedua. Tes ini mengukur aspek-aspek kemampuan mendengarkan (listening), struktur bahasa (structure), dan pemahaman bacaan (comprehension). Oleh karena itu, hasil dari TOEFL sering dijadikan sebagai

tolok ukur dalam menilai tingkat penguasaan bahasa Inggris seseorang (Nasser dan Saldriani, 2019).

Universitas Halu Oleo menjadikan skor TOEFL sebagai salah satu syarat untuk mengikuti sidang skripsi. Sebagaimana Peraturan Rektor Nomor 1 Tahun 2019 tentang Peraturan Akademik di Lingkungan Universitas Halu Oleo, Pasal 79 ayat (3) disebutkan bahwa salah satu syarat untuk mengikuti ujian tugas akhir/skripsi/tesis/disertasi adalah melampirkan sertifikat nilai TOEFL atau TOEFL-Like dan atau yang disertakan dengan nilai skor minimal 400, sehingga banyak mahasiswa berupaya mencari metode atau sarana untuk meningkatkan kesiapan mereka dalam menghadapi ujian TOEFL.

Beberapa mahasiswa mengalami kesulitan dalam menghadapi ujian TOEFL karena kurangnya latihan dan evaluasi yang sistematis. Untuk mengatasi permasalahan tersebut, penggunaan aplikasi simulasi TOEFL berbasis web menjadi solusi yang lebih praktis dan efisien dibandingkan mengikuti pelatihan dan simulasi TOEFL secara konvensional dengan menggunakan soal dalam bentuk cetak.

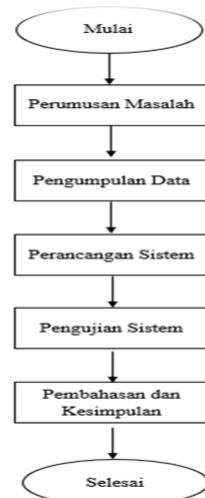
Salah satu tantangan utama dalam pengembangan aplikasi simulasi TOEFL adalah memastikan setiap peserta mendapatkan soal yang bervariasi dan tidak berulang dalam setiap sesi tes. Untuk mengatasi permasalahan tersebut, penerapan algoritma *Random Number Generator* (RNG) menjadi solusi yang dapat digunakan dalam proses pengacakan soal. Algoritma RNG memungkinkan soal yang ditampilkan dalam setiap sesi tes selalu berbeda, sehingga peserta mendapatkan pengalaman mengerjakan soal yang lebih variatif.

Berdasarkan latar belakang tersebut, penulis melakukan sebuah penelitian berjudul "Rancang Bangun Aplikasi Simulasi TOEFL Menggunakan Algoritma Random Number Generator Berbasis Web". Pada penelitian ini akan menggunakan algoritma Multiplicative Congruential Generator (MCG) dalam pengacakan soal. Algoritma ini merupakan salah satu jenis algoritma RNG.

2. METODEOLOGI PENELITIAN

2.1 Prosedur Penelitian

Prosedur penelitian adalah serangkaian langkah-langkah sistematis yang diikuti peneliti sebagai acuan dalam penelitian yang dilakukan. Prosedur yang digunakan pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1 Prosedur Penelitian

2.2 Metode Pengembangan Sistem

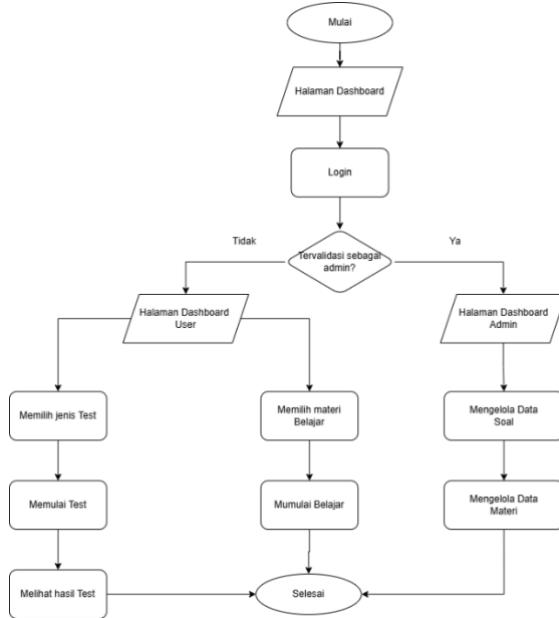
Metode pengembangan sistem yang digunakan dalam pengembangan aplikasi ini adalah R&D (*Research and Development*) yang telah disederhanakan menjadi beberapa langkah utama yaitu Identifikasi masalah, pengumpulan data, desain sistem, pengembangan sistem, dan uji coba sistem.

2.3 Perancangan Sistem

Perancangan sistem mencakup pembuatan *flowchart system*, *use case diagram*, *activity diagram*, serta *sequence diagram* untuk menggambarkan proses yang terjadi di dalam aplikasi. Selain itu, algoritma RNG juga dirancang agar dapat mengacak soal secara efisien tanpa mengurangi tingkat validitas soal TOEFL.

2.3.1 Flowchart

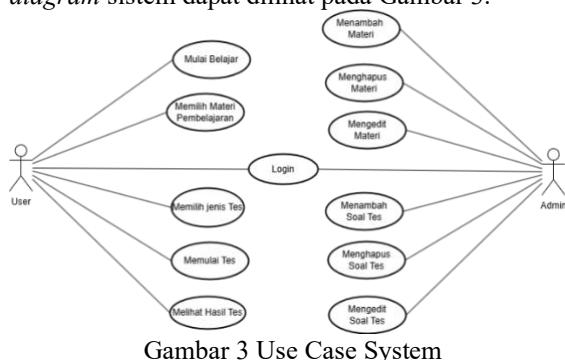
Flowchart membantu dalam proses analisis, perancangan, dan pengkodean dengan memecah masalah menjadi bagian-bagian yang lebih kecil untuk mempermudah pengoperasiannya. *Flowchart* pada sistem ini dapat dilihat pada Gambar 2.



Gambar 2 Flowchart System

2.3.2 Use Case Diagram

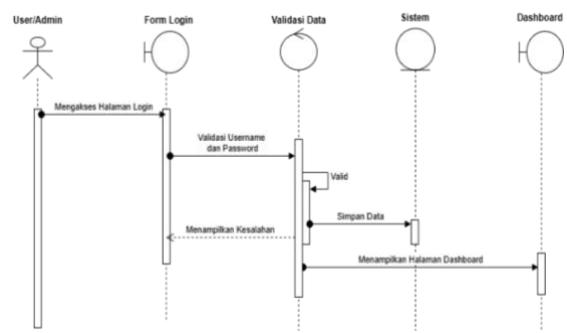
Use case diagram ini akan memvisualisasikan interaksi antar *user* dengan sistem yang dapat membantu dalam memahami bagaimana sistem bekerja dan bagaimana user terlibat dalam sistem tersebut. Dalam sistem ini, terdapat 2 aktor utama yang terlibat yaitu admin (sebagai pengelola sistem) dan *user* (sebagai pengguna sistem). *Use case diagram* sistem dapat dilihat pada Gambar 3.



Gambar 3 Use Case System

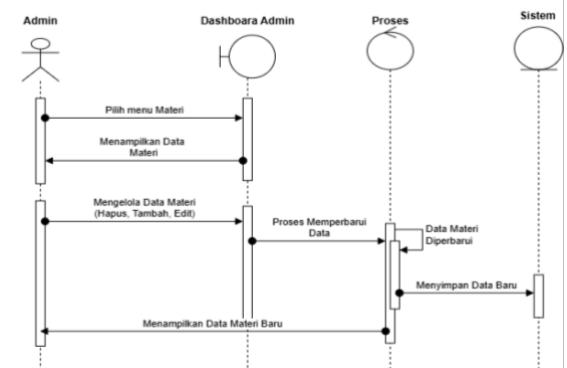
2.3.3 Sequence Diagram

1. *Sequence diagram* *Login* dimulai dengan *user/admin* mengakses halaman *form Login* dan memasukan *username* dan *password*, kemudian sistem akan melakukan validasi data. Jika *username* dan *password* tervalidasi maka sistem akan menyimpan data tersebut dan menampilkan halaman *dashboard*. *sequence diagram login* dapat dilihat pada Gambar 4.



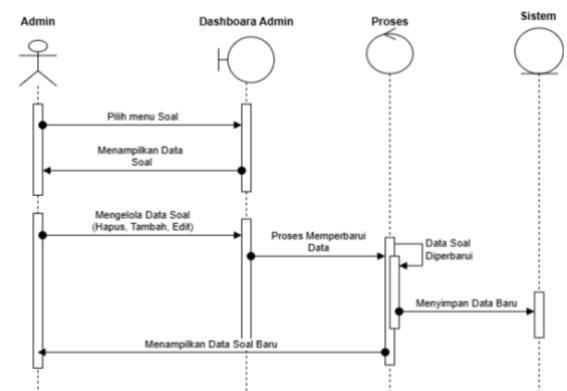
Gambar 4 Sequence Diagram Login

2. *Sequence diagram* materi dimulai dengan admin mengakses halaman materi. Pada halaman ini, admin mempunyai akses untuk mengelola data seperti menambah, menghapus, atau mengedit data materi yang sebelumnya. *Sequence diagram* materi dapat dilihat pada Gambar 5.



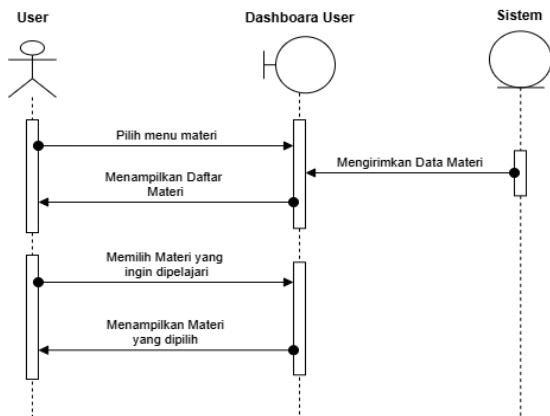
Gambar 5 Sequence Diagram materi

3. *Sequence diagram* soal dimulai dengan admin mengakses halaman admin kemudian memilih menu soal. Pada halaman ini, admin mempunyai akses untuk mengelola data soal dengan memilih aksi seperti menambah, menghapus, atau mengedit soal-soal yang sebelumnya. *Sequence diagram* soal dapat dilihat pada Gambar 6.



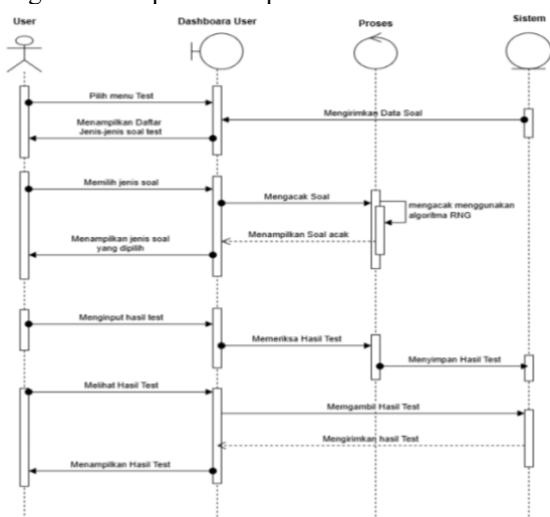
Gambar 6 Sequence Diagram soal

4. *Sequence diagram* belajar dimulai dengan user mengakses halaman belajar. Setelahnya sistem akan menampilkan daftar materi yang dapat dipilih oleh user. *Sequence diagram* belajar dapat dilihat pada Gambar 7.



Gambar 7 Sequence diagram belajar

- Sequence diagram tes dimulai dengan user yang masuk ke menu dashboard dan memilih menu tes. Kemudian sistem akan menampilkan jenis-jenis soal tes yang dapat dipilih oleh user. Sequence diagram tes dapat dilihat pada Gambar 8.



Gambar 8 Sequence diagram tes

2.3.4 Penerapan Algoritma RNG

Pada sistem simulasi TOEFL berbasis web, algoritma MCG (*Multiplicative Congruential Generator*) digunakan untuk mengacak urutan soal yang ditampilkan kepada pengguna secara acak dan dinamis. MRGN merupakan varian dari LCG (*Linear Congruential Generator*) yang disederhanakan, di mana tidak digunakan konstanta penambah (c). Persamaan yang digunakan dari algoritma ini adalah

$$Z_{i+1} = (a \cdot Z_i) \bmod m \quad (2.1)$$

Keterangan:

- Z_i = Bilangan acak ke i (seed)
- a = Bilangan konstanta pengali ($a < m$) bilangan bulat
- m = Bilangan konstanta modulus ($m > 0$) bilangan bulat.

Berikut adalah contoh penerapan algoritma MCG dalam pengacakan soal, dimana akan dilakukan

pengacakan pada 10 soal dengan nilai awal $Z_i = 135$, $a = 20$, dan $m = 148$.

Soal nomor 1

$$\begin{aligned} Zi + 1 &= (a \cdot Zi) \bmod m \\ &= (20 \cdot 135) \bmod 148 \\ &= 2700 \bmod 148 \\ &= 36 \end{aligned}$$

$$\begin{aligned} Ri &= Zi + 1 / m \\ &= 36 / 148 \\ &= 0,2432 \end{aligned}$$

Soal nomor 2

$$\begin{aligned} Zi + 1 &= (a \cdot Zi) \bmod m \\ &= (20 \cdot 36) \bmod 148 \\ &= 720 \bmod 148 \\ &= 128 \\ Ri &= Zi + 1 / m \\ &= 128 / 148 \\ &= 0,8649 \end{aligned}$$

Soal nomor 3

$$\begin{aligned} Zi + 1 &= (a \cdot Zi) \bmod m \\ &= (20 \cdot 128) \bmod 148 \\ &= 2560 \bmod 148 \\ &= 44 \\ Ri &= Zi + 1 / m \\ &= 44 / 148 \\ &= 0,2973 \end{aligned}$$

Soal nomor 4

$$\begin{aligned} Zi + 1 &= (a \cdot Zi) \bmod m \\ &= (20 \cdot 44) \bmod 148 \\ &= 880 \bmod 148 \\ &= 140 \\ Ri &= Zi + 1 / m \\ &= 140 / 148 \\ &= 0,9459 \end{aligned}$$

Soal nomor 5

$$\begin{aligned} Zi + 1 &= (a \cdot Zi) \bmod m \\ &= (20 \cdot 140) \bmod 148 \\ &= 2800 \bmod 148 \\ &= 136 \\ Ri &= Zi + 1 / m \\ &= 136 / 148 \\ &= 0,9189 \end{aligned}$$

Soal nomor 6

$$\begin{aligned} Zi + 1 &= (a \cdot Zi) \bmod m \\ &= (20 \cdot 136) \bmod 148 \\ &= 2720 \bmod 148 \\ &= 56 \\ Ri &= Zi + 1 / m \\ &= 56 / 148 \\ &= 0,3784 \end{aligned}$$

Soal nomor 7

$$\begin{aligned} Zi + 1 &= (a \cdot Zi) \bmod m \\ &= (20 \cdot 56) \bmod 148 \\ &= 1120 \bmod 148 \\ &= 84 \\ Ri &= Zi + 1 / m \\ &= 84 / 148 \\ &= 0,5676 \end{aligned}$$

Soal nomor 8

$$\begin{aligned} Z_i + 1 &= (a \cdot Z_i) \bmod m \\ &= (20 \cdot 84) \bmod 148 \\ &= 1680 \bmod 148 \\ &= 52 \end{aligned}$$

$$\begin{aligned} R_i &= Z_i + 1 / m \\ &= 52 / 148 \\ &= 0,3514 \end{aligned}$$

Soal nomor 9

$$\begin{aligned} Z_i + 1 &= (a \cdot Z_i) \bmod m \\ &= (20 \cdot 52) \bmod 148 \\ &= 1040 \bmod 148 \\ &= 4 \end{aligned}$$

$$\begin{aligned} R_i &= Z_i + 1 / m \\ &= 4 / 148 \\ &= 0,0270 \end{aligned}$$

Soal nomor 10

$$\begin{aligned} Z_i + 1 &= (a \cdot Z_i) \bmod m \\ &= (20 \cdot 4) \bmod 148 \\ &= 80 \bmod 148 \\ &= 80 \end{aligned}$$

$$\begin{aligned} R_i &= Z_i + 1 / m \\ &= 80 / 148 \\ &= 0,5405 \end{aligned}$$

Untuk dapat mendapatkan hasil pengacakan soal, dilakukan pengurutan nomor soal berdasarkan nilai R . Penomoran diberikan berdasarkan nilai R terkecil ke nilai R terbesar dengan menggunakan persamaan random variate.

$$Random Variate : R_i = Z_{i+1} / m \quad (2.2)$$

Maka diperoleh bilangan acak yang dapat dilihat pada Tabel 2.1.

Tabel 3.1 Hasil Bilangan Acak Algoritma MRNG

i	a	Z_i	$a \cdot Z_i$	Mod m	Z_{i+1}	R	Urutan Soal
1	20	135	2700	148	36	0,2432	Ke-2
2	20	36	720	148	128	0,8649	Ke-8
3	20	128	2560	148	44	0,2973	Ke-3
4	20	44	880	148	140	0,9459	Ke-10
5	20	140	2800	148	136	0,9189	Ke-9
6	20	136	2720	148	56	0,3784	Ke-5
7	20	56	1120	148	84	0,5676	Ke-7
8	20	84	1680	148	52	0,3514	Ke-4
9	20	52	1040	148	4	0,0270	Ke-1
10	20	4	80	148	80	0,5405	Ke-6

Dari hasil penerapan algoritma MRNG tersebut, maka didapatkan urutan soal yang acak yaitu sebagai berikut:

Soal nomor 1 adalah urutan ke-2

Soal nomor 2 adalah urutan ke-8

Soal nomor 3 adalah urutan ke-3

Soal nomor 4 adalah urutan ke-10

Soal nomor 5 adalah urutan ke-9

Soal nomor 6 adalah urutan ke-5

Soal nomor 7 adalah urutan ke-7

Soal nomor 8 adalah urutan ke-4

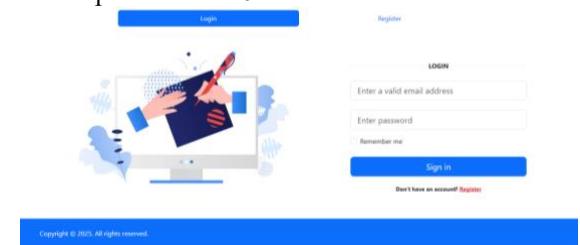
Soal nomor 9 adalah urutan ke-1

Soal nomor 10 adalah urutan ke-6

3. IMPLEMENTASI SISTEM

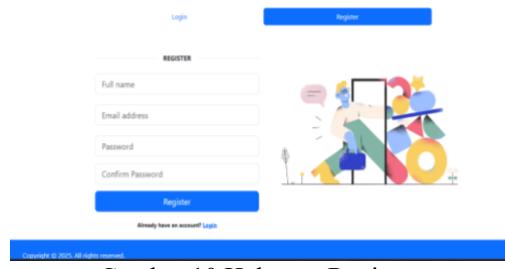
1. Implementasi halaman login dan register

Pada halaman login, pengguna yang sudah memiliki akun dapat masuk ke dalam aplikasi dengan memasukkan username dan password yang telah terdaftar. Informasi ini akan di proses oleh sistem dengan melakukan validasi role pengguna. Tiap role akan memiliki tampilan yang berbeda. Tampilan halaman login dapat dilihat pada Gambar 9.



Gambar 9 Halaman Login

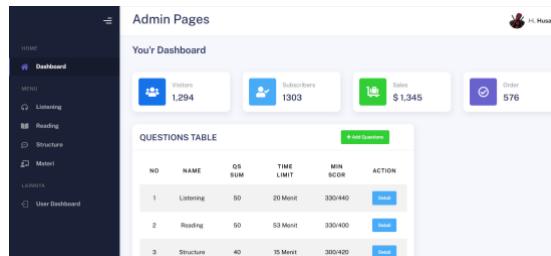
Jika belum terdaftar, pengguna baru dapat melakukan registrasi dengan mengisi data identitas yang diperlukan berupa nama, email, dan *password* pada halaman *register*. Secara otomatis, pengguna yang pertama kali melakukan registrasi akan memiliki *role* sebagai *admin*, sedangkan pengguna berikutnya akan terdaftar sebagai *user*. Tampilan halaman register dapat dilihat pada Gambar 10.



Gambar 10 Halaman Register

2. Halaman Dashboard Admin

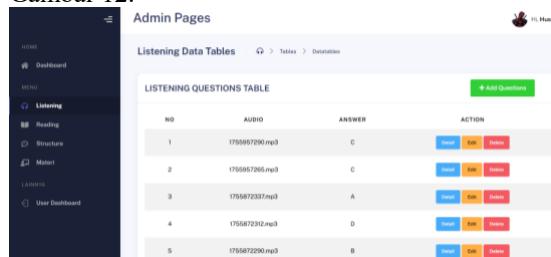
Pada halaman ini, admin dapat melihat ringkasan informasi berupa jumlah pengunjung, jumlah pengguna terdaftar, serta data tambahan lainnya yang ditampilkan dalam bentuk kartu informasi di bagian dashboard. Halaman Dashboard admin dapat dilihat pada Gambar 11.



Gambar 11 Halaman Dashboard Admin

3. Halaman Questions

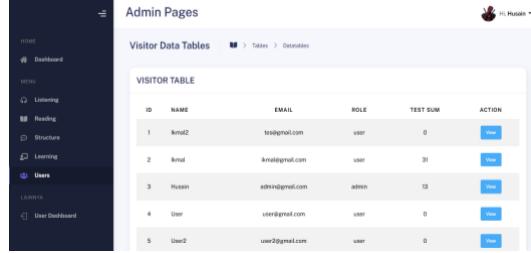
Halaman questions merupakan bagian dari dashboard admin yang digunakan untuk menambahkan, melihat, mengedit, dan menghapus soal soal TOEFL (listening, reading, dan structure). Pada halaman ini, daftar soal-soal ditampilkan dalam bentuk tabel yang memuat informasi nomor urut, file audio, jawaban benar, serta menu aksi. Tampilan halaman *listening questions* dapat dilihat pada Gambar 12.



Gambar 12 Halaman Listening Questions

4. Halaman Lihat List User

Pada halaman ini, admin dapat melihat list user yang sudah registrasi. Tampilan halaman lihat hasil user dapat dilihat pada Gambar 13.



Gambar 13 Halaman Lihat List User

5. Halaman Dashboard User

Halaman dashboard user atau beranda merupakan tampilan awal yang ditujukan untuk pengguna dengan role user setelah berhasil masuk ke dalam aplikasi simulasi TOEFL. Pada halaman ini ditampilkan informasi utama mengenai fungsi dan tujuan dari aplikasi, yaitu sebagai media pembelajaran dan latihan untuk menghadapi tes TOEFL. Tampilan halaman dashboard user dapat dilihat pada Gambar 14.



Gambar 14 Halaman Dashboard User

6. Halaman Kelas User

Halaman kelas menampilkan pilihan materi pembelajaran yang terbagi menjadi tiga, yaitu Listening, Reading, dan Structure, yang dapat diakses pengguna sesuai kebutuhan pembelajaran TOEFL. Tampilan halaman kelas dapat dilihat pada Gambar 15.



Gambar 15 Halaman Kelas User

7. Halaman Tes User

Halaman tes user menampilkan pilihan jenis ujian TOEFL yang terdiri dari Listening, Reading, dan Structure. Setiap jenis tes dilengkapi dengan informasi jumlah soal, durasi waktu, serta skor minimum, sehingga pengguna dapat mempersiapkan diri sebelum memulai simulasi. Saat user memilih jenis tes, sistem akan mengambil soal dari database sesuai kategori yang dipilih. Untuk menjaga variasi, soal dapat dipanggil secara acak dengan bantuan algoritma RNG. Tampilan halaman tes user dapat dilihat pada Gambar 16.



Gambar 16 Halaman Dashboard User

8. Halaman Hasil Tes User

Halaman hasil tes user menampilkan skor terbaru pengguna lengkap dengan jenis tes, tanggal, raw score (jumlah benar), dan scaled score (jumlah skor). Hasil perhitungan berupa raw score dan scaled score kemudian disimpan ke dalam tabel user_tests bersama informasi user, jenis tes, serta tanggal pelaksanaan. Selain itu, terdapat tabel Riwayat Tes yang menyimpan catatan hasil ujian sebelumnya agar user dapat

memantau perkembangan kemampuannya. Tampilan halaman hasil tes user dapat dilihat pada Gambar 17.

Hasil Test Anda				
Hasil Test : Husain		Tanggal Tes:	Raw Score	Scaled Score
Jenis Tes	Listening	03 Sep 2025	3 / 10	27
Request Test				
#	Jenis Tes	Tanggal Tes	Raw Score	Scaled Score
1	Listening	03/09/2025	3/10	27
2	Listening	23/08/2025	4/10	23
3	Structure	23/08/2025	9/20	31

Gambar 17 Halaman Hasil Tes User

4. PENGUJIAN METODE MCG

4.1. Pengujian Skenario Ke-1

Pengujian metode MCG pada skenario satu dilakukan pada tiga jenis tes, yaitu Listening, Reading, dan Structure. Masing-masing tes terdiri atas 50 soal untuk Listening dan Reading, serta 40 soal untuk Structure. Proses pengacakan dilakukan sebanyak 10 kali percobaan untuk setiap jenis tes dengan tujuan menguji konsistensi sekaligus kualitas acak dari algoritma yang digunakan. Proses pengacakan dilakukan melalui aplikasi web yang telah dirancang sebelumnya. Hasil pengujian tersebut dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Pengujian MCG Skenario 1

No	Soal	Urutan Soal									
		1	2	3	4	5	6	7	8	9	10
<i>Structure Questions</i>											
1.	Str 1	18	17	27	24	30	32	30	33	28	32
2.	Str 2	36	9	4	19	23	13	36	38	24	36
3.	Str 3	21	8	12	35	2	3	40	12	10	28
4.	Str 4	40	24	20	14	32	14	39	7	17	39
5.	Str 5	24	23	8	16	21	29	15	4	14	15
<i>Listening Questions</i>											
6.	Lst 1	29	33	35	28	37	9	18	25	21	20
7.	Lst 2	23	22	7	14	14	6	43	28	27	36
8.	Lst 3	10	5	44	41	44	16	48	17	48	37
9.	Lst 4	20	45	5	30	41	10	29	46	16	46
10.	Lst 5	19	12	42	6	9	17	21	37	46	19
<i>Reading Questions</i>											
11.	Rd 1	3	9	8	7	9	6	7	8	7	6
12.	Rd 2	2	6	3	1	4	8	1	7	10	5
13.	Rd 3	8	8	7	6	1	5	2	2	3	9
14.	Rd 4	4	2	4	8	6	1	9	4	5	3
15.	Rd 5	7	1	1	4	8	2	5	10	2	1

Hasil pengujian menunjukkan bahwa pada setiap percobaan, urutan soal yang dihasilkan berbeda-beda. Sebagai contoh, pada tes structure soal dengan nomor tertentu dapat muncul pada posisi ke-18 pada percobaan pertama, tetapi pada percobaan berikutnya dapat bergeser ke posisi ke-27 atau 32. Hal serupa juga terjadi pada soal listening maupun reading, di mana setiap soal menempati posisi yang bervariasi pada setiap percobaan. Perubahan urutan ini membuktikan bahwa algoritma MCG berhasil menghasilkan distribusi soal yang acak sehingga tidak terjadi pola pengulangan yang tetap.

4.2. Pengujian Skenario Ke-2

Skenario pengujian kedua dilakukan pada jenis tes structure, yang dimana soal yang akan ditampilkan adalah 40 soal acak dari total 100 soal yang terdapat di database. Proses pengujian dilakukan sebanyak 10 kali percobaan dengan tujuan menguji kualitas pengacakan jika terdapat banyak data pada bank soal yang tersimpan di database. Hasil pengujian tersebut dapat dilihat pada Tabel 4.2

Tabel 4.2 Hasil Pengujian MCG Skenario 2

No	Percobaan Ke-									
	1	2	3	4	5	6	7	8	9	10
Urutan Soal										
1.	Str 28	Str 4	Str 5	Str 13	Str 80	Str 74	Str 84	Str 28	Str 87	Str 45
2.	Str 30	Str 6	Str 43	Str 68	Str 69	Str 10	Str 83	Str 82	Str 79	Str 44
3.	Str 36	Str 46	Str 88	Str 34	Str 31	Str 40	Str 51	Str 53	Str 51	Str 36
4.	Str 25	Str 60	Str 20	Str 77	Str 21	Str 19	Str 74	Str 40	Str 60	Str 14
5.	Str 74	Str 41	Str 70	Str 14	Str 61	Str 73	Str 8	Str 90	Stl 00	Str 33
6.	Str 71	Str 88	Str 92	Str 21	Str 56	Str 44	Str 78	Str 25	Str 68	Str 78
7.	Str 1	Str 99	Str 15	Str 50	Str 62	Str 41	Str 21	Str 23	Str 33	Str 25
8.	Str 62	Str 39	Str 79	Str 78	Str 16	Str 55	Str 72	Str 24	Str 99	Str 67
9.	Str 64	Str 15	Str 72	Str 92	Str 92	Str 62	Str 75	Str 8	Str 76	Str 21
10.	Str 81	Str 55	Str 24	Str 39	Str 89	Str 24	Str 23	Str 2	Str 14	Str 61

Hasil pengujian diatas memperlihatkan bahwa soal yang ditampilkan bervariasi pada setiap percobaan. Misalnya, pada percobaan pertama muncul soal Str28, Str4, Str5, dan seterusnya, sedangkan pada percobaan kedua susunan soalnya berubah menjadi Str30, Str6, Str43, dan seterusnya. Secara keseluruhan hasil pengujian membuktikan bahwa algoritma MCG berhasil memilih 40 soal dari 100 soal dengan distribusi acak yang berbeda pada setiap percobaan, sehingga tidak ada pola pengulangan yang sama.

4.3. Pengujian Skenario Ke-3

Skenario pengujian ketiga dilakukan pada jenis soal structure dengan membandingkan hasil pengacakan soal jika terdapat 2 user dengan time (waktu akses) yang saman namun id user berbeda yang dimana time dan id user tersebut akan menjadi nilai dari variabel Zi atau seed awal ($Z_i = \text{time} + \text{Id User}$).

Soal yang akan ditampilkan adalah 40 soal acak dari total 100 soal, dengan nilai time = 251005221346 (tahun, bulan, tanggal, jam, menit, dan detik saat user memulai tes), dan id user 1 = 9915704, id user 2 = 2108677. Pengujian ini dilakukan untuk memastikan bahwa algoritma MCG mampu menghasilkan hasil pengacakan yang berbeda walaupun dua pengguna

memulai tes pada waktu yang sama. Hasil pengujian tersebut dapat dilihat pada Tabel 4.3

Tabel 4.3 Hasil Pengujian MCG Skenario 3

User	Urutan Soal									
	1	2	3	4	5	6	7	8	9	10
User 1	Str 10	Str 46	Str 73	Str 92	S1 00	Str 65	Str 20	Str 22	Str 27	Str 51
User 2	Str 37	Str 35	Str 82	Str 20	Str 62	Str 42	Str 4	Str 71	Str 3	Str 95

Dari hasil pengujian di atas terlihat bahwa meskipun kedua pengguna memiliki waktu akses yang sama, perbedaan nilai Id User menyebabkan nilai seed awal berbeda. Perbedaan kecil pada nilai seed ini menghasilkan urutan soal yang berbeda secara signifikan. Meskipun terdapat beberapa soal yang muncul berulang pada posisi yang sama, hal tersebut masih wajar karena bilangan acak yang dihasilkan bersifat pseudo-random, bukan benar-benar acak sempurna. Namun demikian, pola pengulangan tersebut tidak memengaruhi tujuan utama dari pengacakan, yaitu mencegah peserta tes mendapatkan urutan soal yang sama. Hal ini mendukung tujuan sistem untuk mengurangi kemungkinan peserta menghafal urutan soal pada tes berikutnya

5. PENUTUP

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa penggunaan algoritma RNG dapat diterapkan dengan baik pada sistem simulasi TOEFL. Dari hasil pengujian terlihat bahwa proses pengacakan soal menggunakan metode MCG mampu menghasilkan urutan soal yang berbeda pada setiap percobaan. Setiap soal memperoleh posisi yang bervariasi sehingga distribusinya relatif merata dan tidak terfokus pada posisi tertentu saja. Sebagai contoh, hasil pengujian skenario pertama pada tes structure soal dengan nomor tertentu dapat muncul pada posisi ke-18 pada percobaan pertama, tetapi pada percobaan berikutnya dapat bergeser ke posisi ke-27 atau 32; Hasil pengujian juga didapatkan bahwa meskipun kedua pengguna memiliki waktu akses yang sama, perbedaan nilai Id User menyebabkan nilai seed awal berbeda. Perbedaan kecil pada nilai seed ini menghasilkan urutan soal yang berbeda secara signifikan. Hal ini mendukung tujuan sistem untuk mengurangi kemungkinan peserta menghafal urutan soal pada tes berikutnya.

6. DAFTAR PUSTAKA

- [1] Astuti, Y. Y., Anjarwani, S. E., & Irmawati, B. (2021). Rancang Bangun Aplikasi Simulasi Tes TOEFL Berbasis Website (Studi Kasus : Program Studi Teknik Informatika Universitas Mataram). *Jurnal Teknologi Informasi*, Komputer Dan Aplikasinya (JTIKA), 6(2), 409–416.
- [2] Budikusuma, I., & Susanto, E. S. (2022). Pengembangan Aplikasi Toefl Practice Exam Berbasis Website Pada Universitas Teknologi Sumbawa. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 6(1), 75–78.
- [3] Chen, X., Ji, Z., Fan, Y., & Zhan, Y. (2017). Restful API Architecture Based on Laravel Framework. *Journal of Physics: Conference Series*, 910(1–6).
- [4] Endra, R. Y., Aprilinda, Y., Dharmawan, Y. Y., & Ramadhani, W. (2021). Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website. *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, 11(1), 48–55.
- [5] Gus, Nofriansyah, D., & Prayudha, J. (2022). E-Try Out Toefl Online Pada ILTC Menggunakan Metode Multi Random Number Generator. *Jurnal Cyber Tech*, 4(3), 1–9.
- [6] Guzmaliza, D. (2019). Perangkat Lunak Bantu Administrasi Keuangan Sekolah Tinggi Teknologi Pagar Alam Dengan Php Dan Mysql. *Jurnal Ilmiah Betrik*, 10(01), 28–37.
- [7] Hidayah, W. N. (2021). Konsep Solusi Terhadap Problem Keterampilan (Skills) Berbahasa Inggris Di Lembaga Pendidikan Indonesia. *Jurnal Pendidikan Indonesia (Japendi)*, 2(10), 1824–1834.
- [8] Lubis, J. , Fitri, N. Z. N., & Ridwan, S. C. (2024). Pentingnya Menguasai Bahasa Inggris dan Faktor Yang Mempengaruhi Kemampuan Berbahasa Inggris. *Karimah Tauhid*, 3(3), 3599–3605.
- [9] Malabay. (2016). Pemanfaatan Flowchart Untuk Kebutuhan Deskripsi Proses Bisnis. *Jurnal Ilmu Komputer*, 12(1), 21–26.
- [10] Mulyapradana, A., Anjarini, A. D., Elshifa, A., Gulam, H., & Winarsih. (2023). Pengenalan dan Latihan Soal Tes TOEFL Guna Meningkatkan Kemampuan bahasa Inggris pada Mahasiswa. *BERDAYA: Jurnal Pendidikan Dan Pengabdian Kepada Masyarakat*, 5(1), 51–60.
- [11] Nasser, R., & Saldriani, P. (2019). Perancangan Dan Pembuatan Aplikasi Simulasi Toefl Berbasis Desktop. *Prosiding Semantik*, 2(2011), 184–191.
- [12] Nawassarif, Yunawari, W., & Shalihin, A. (2021). Rancang Bangun Aplikasi Percetakan Tiga Bersaudara Berbasis Web Dengan Metode Waterfall. *JINTEKS (Jurnal Informatika Teknologi Dan Sains)*, 3(2), 354–361.
- [13] Nurjani, Y. (2020). Website Kantor Kelurahan Lingkar Selatan. *Fortech (Journal of Information Technology)*, 3(2), 53–59.
- [14] Pasaribu, P., & Susanti, W. (2021). Sistem Informasi Pengajaran Rancangan Usulan Penelitian Menggunakan PHP Native dan Bot

- Telegram. *Jurnal Mahasiswa Aplikasi Teknologi Komputer Dan Informasi*, 3(1), 29–38.
- [15] Pramudita, G., & Nuryana, I. K. . (2023). Implementasi Algoritma Multiplicative Congruential (RNG) pada Aplikasi Ujian Sekolah Berbasis Komputer Berbasis Framework Laravel. *Journal of Informatics and Computer Science (JINACS)*, 5(02), 244–252.
- [16] Prasetyianto, M., Ruminar, H., Maharddhika, R., & Prasetyaningrum, D. P. (2024). Pelatihan Persiapan Dan Simulasi TOEFL Untuk Mendukung Pemetaan Kompetensi Bahasa Inggris Akademik. *JAPI (Jurnal Akses Pengabdian Indonesia)*, 9(1), 43–52.
- [17] Prasetyadi, R., Nugroho, N. B., & Azlan. (2020). Implementasi Metode Multiplicative Random Number Generator (MRNG) Pada Aplikasi Ujian Sekolah Berbasis Komputer. *Jurnal CyberTech*, 3(2), 224–229.
- [18] Pratama, S. D., Lasimin., & Dadaprawira, M. N. (2023). Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence Dan Boundary Value. *J-SISKO TECH (Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD)*, 6(2), 560–569.
- [19] Rachmadtullah, M. R. W., & Prihanto, A. (2022). Implementasi Algoritma Random Number Generator (RNG) pada Aplikasi Try Out Berbasis Framework Laravel. *Journal of Informatics and Computer Science (JINACS)*, 3(03), 300–307.
- [20] Renaldy, & Rustam, A. (2022). Perancangan Sistem Informasi Inventory Berbasis Web Pada Gudang Di Pt. Spin Warriors. *Aisyah Journal of Informatics and Electrical Engineering*, 4(1), 27–32.
- [21] Saputri, S. S. (2020). Pengenalan Flashcard sebagai Media untuk Meningkatkan Kemampuan Bahasa Inggris. *ABDIKARYA: Jurnal Pengabdian Dan Pemberdayaan Masyarakat*, 2(1), 56–61.
- [22] Sulfia, E., Dewi, S., Munawwaroh, K., Melati, E., & Hafrida, L. (2022). Pelatihan Toefl Preparation Course Bagi Mahasiswa Program Studi Pendidikan Matematika Fakultas Fkip Universitas Batanghari. *Jurnal Abdimas Adpi Sosial Dan Humaniora*, 2(4), 195–200.
- [23] Siregar, H., & Sari, N. (2018). Rancang Bangun Aplikasi Simpan Pinjam Uang Mahasiswa Fakultas Teknik Universitas Asahan Berbasis Web. *Jurnal Teknologi Informasi*, 2(1), 53–59.
- [24] Sonata, F.-. (2019). Pemanfaatan UML (Unified Modeling Language) Dalam Perancangan Sistem Informasi E-Commerce Jenis Customer-To-Customer. *Jurnal Komunika : Jurnal Komunikasi, Media Dan Informatika*, 8(1), 22. <https://doi.org/10.31504/komunika.v8i1.1832>
- [25] Taufan, M. A., Rusdianto, D. R., & Ananta, M. T. (2022). Pengembangan Sistem Otomatisasi Use Case Diagram berdasarkan Skenario Sistem menggunakan Metode POS Tagger Stanford NLP. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 6(8), 3733–3740.
- [26] Thariq, P. A., Husna, A., Aulia, E., Djusfi, A. R., Lestari, R., Fahrimal, Y., & Jhoanda, R. (2021). Sosialisasi Pentingnya Menguasai Bahasa Inggris Bagi Mahasiswa. *Jurnal Pengabdian Masyarakat: Darma Bakti Teuku Umar*, 2(2), 316–325.
- [27] Usada, E., Yuniarsaha, Y., & Rifani, N. (2021). Rancang Bangun Sistem Informasi Jadwal Perkuliahan Berbasis Jquery Mobile Dengan Menggunakan Php Dan Mysql. *Jurnal Infotel*, 4(November), 40–51.
- [28] Zainal, M., & Kasmawati, S. (2021). Optimalisasi Google Site sebagai Media Pembelajaran Berbasis Website pada Pembelajaran Jarak Jauh. *Seminar Nasional Pendidikan LPPM IKIP PGRI Bojonegoro*, 3(April), 42–51.