

Rancang Bangun Aplikasi Quiz Simulasi TOEFL Memanfaatkan Algoritma Linear Congruential Generator (LCG) Berbasis Android

Design and build a TOEFL simulation quiz application utilizing the Android-based Linear Congruential Generator (LCG) algorithm

Tengku Mhd. Zulfikar^a, Lili Tanti^b

^{a,b}Potensi Utama, Jl. K.L Yos Sudarso KM. 6,5 No. 3A Tj. Mulia, Medan 20214, Indonesia
zulfikartengku3117@gmail.com^a, lilitanti82@gmail.com^b

ABSTRAK

TOEFL (Test of English as a Foreign Language) merupakan tes yang digunakan untuk mengetahui tingkat kemampuan seseorang dalam berbahasa Inggris. Tes TOEFL yang biasanya dilakukan oleh mahasiswa adalah tes TOEFL PBT (Paper Based Test) yang meliputi tes listening comprehension, structure and written expression, dan reading comprehension. Setiap mahasiswa seharusnya diberikan soal yang berbeda dan tidak sama dengan mahasiswa lainnya. Namun masih sering terjadi kesalahan (human error) yang menyebabkan mahasiswa memiliki soal yang sama atau tidak teracak. Oleh karena itu dibutuhkan sistem yang dapat mengacak soal dalam TOEFL secara efektif dan efisien. Penerapan algoritma linear congruential generator (LCG) pada aplikasi quiz bertujuan menampilkan soal yang berbeda atau acak sehingga mahasiswa tidak mendapatkan soal yang sama setiap membuka aplikasi. Dengan menerapkan metode ini diharapkan membuat mahasiswa terbiasa dengan soal-soal yang ada dalam TOEFL dan mendapatkan nilai yang tinggi saat tes TOEFL yang sebenarnya. Aplikasi ini bertujuan untuk menampilkan soal-soal quiz simulasi TOEFL yang telah diacak menggunakan metode linear congruential generator (LCG). Aplikasi ini dibuat menggunakan software android studio dan bahasa pemrograman dart dengan database aplikasinya yaitu firebase. Hasil Pengujian menunjukkan sistem dapat berjalan dengan baik pada android maupun laptop, sistem dapat memperbarui data pengguna secara realtime menggunakan firebase dan tampilan antarmuka yang sederhana akan mudah dipahami bagi pengguna baru yang melakukan simulasi quiz TOEFL. Setelah dilakukan pengujian maka dapat disimpulkan bahwa aplikasi quiz simulasi TOEFL berbasis android memanfaatkan algoritma Linear Congruential Generator (LCG) dengan online system yang dapat menghasilkan soal-soal yang acak sehingga mahasiswa tidak dapat bekerja sama pada saat mengerjakan quiz TOEFL.

Kata Kunci : TOEFL, LCG, Dart, Android Studio, Firebase

ABSTRACT

TOEFL (Test of English as a Foreign Language) is a test used to assess an individual's proficiency in the English language. The TOEFL test commonly taken by students is the TOEFL PBT (Paper Based Test), which includes sections on listening comprehension, structure and written expression, and reading comprehension. Each student should be provided with different questions to prevent duplication or a lack of randomness. However, human errors often occur, leading to students receiving the same or unrandomized questions. Therefore, a system is needed that can effectively and efficiently randomize TOEFL questions. The implementation of the Linear Congruential Generator (LCG) algorithm in the quiz application aims to present different or randomized questions to students each time they access the application. By using this method, it is hoped that students can become accustomed to the types of questions found in the TOEFL and achieve high scores on the actual TOEFL test. This application's purpose is to display randomized TOEFL quiz questions using the Linear Congruential Generator (LCG) method. The application was created using Android Studio software and the Dart programming language, with Firebase as the application's database. The testing results show that the system functions well on both Android devices and laptops. The system can update user data in real-time using Firebase, and its simple user interface is easy for new users to understand when participating in the TOEFL quiz simulation. In conclusion, after testing, it can be affirmed that the Android-based TOEFL quiz simulation application utilizes the Linear Congruential Generator (LCG) algorithm with an online system to generate randomized questions, ensuring that students cannot collaborate during the TOEFL quiz.

Keywords : TOEFL, LCG, Dart, Android Studio, Firebase

Info Artikel :

Disubmit: 1 Oktober 2023

Direview: 6 November 2023

Diterima : 11 Desember 2023

Copyright © 2024 - Journal UPU. All rights reserved.

1. PENDAHULUAN

Perkembangan teknologi informasi yang semakin maju dan berkembang pada era digital memberikan banyak kemudahan pada kehidupan sehari-hari. Teknologi informasi juga memberikan banyak perkembangan baru pada berbagai bidang terutama bidang pendidikan dan edukasi. Sistem konvensional yang sebelumnya digunakan telah berubah menjadi sistem komputerisasi, beberapa layanan pendidikan telah menggunakan sistem komputerisasi dengan cara *online*, salah satu diantaranya adalah ujian secara *online*. [1]

TOEFL (*Test of English as a Foreign Language*) merupakan tes yang digunakan untuk mengetahui tingkat kemampuan seseorang dalam berbahasa Inggris[2]. Salah satu faktor yang mempengaruhi keberhasilan tes TOEFL adalah dengan banyaknya latihan, penggunaan media yang tepat akan mempermudah proses pembelajaran dan latihan[3]. Sehingga diperlukan suatu basis aplikasi yang dapat digunakan sebagai simulasi dan latihan sebelum mengikuti tes TOEFL.

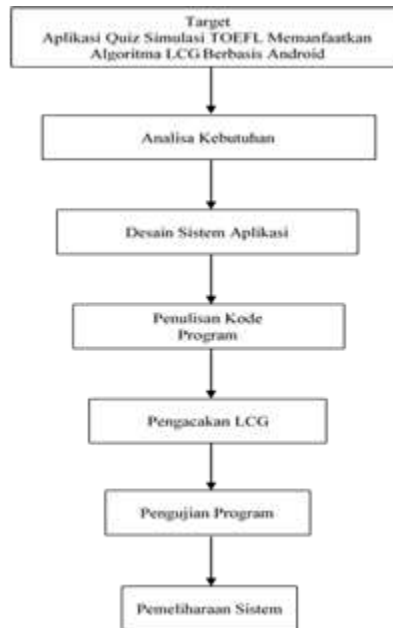
Tes TOEFL yang biasanya dilakukan oleh mahasiswa adalah tes TOEFL PBT (*Paper Based Test*) yang meliputi tes *listening comprehension*, *structure and written expression*, dan *reading comprehension*[4]. Dalam tes TOEFL yang pernah penulis lakukan diawali dengan pendaftaran melalui lembaga/instansi resmi dari TOEFL. Sebelum mengerjakan tes dilakukan pengenalan dan bimbingan TOEFL serta pembahasan mengenai soal-soal yang ada dalam tes TOEFL melalui *zoom meeting*. Setelah diberikan jadwal dan lokasi tes, penulis melakukan tes TOEFL yang bersifat tertulis (PBT) secara langsung atau tatap muka. Setiap mahasiswa seharusnya diberikan soal yang berbeda dan tidak sama dengan mahasiswa lainnya. Namun masih sering terjadi kesalahan (*human error*) yang menyebabkan mahasiswa memiliki soal yang sama atau tidak teracak. Kesulitan yang sering terjadi saat mengerjakan tes TOEFL adalah soal yang lumayan banyak dan waktu yang relatif singkat. Hal itu membuat mahasiswa tidak terbiasa dengan soal-soal yang ada dalam TOEFL dan membuat nilai/skor yang didapatkan saat tes TOEFL menjadi tidak maksimal.

Oleh karena itu dibutuhkan metode yang dapat membantu memperdalam daya ingat mahasiswa dan tidak panik saat mengerjakan soal yang ada dalam TOEFL. *Linear Congruential Generator* (LCG) merupakan sebuah metode yang efektif untuk pengacakan soal-soal yang berhubungan dengan *quiz* seperti TOEFL. LCG adalah algoritma yang sering diimplementasikan pada beberapa bahasa pemrograman untuk membangkitkan bilangan acak[5]. Penerapan algoritma LCG merupakan solusi untuk menampilkan soal nomor yang acak setiap kali pengguna membuka aplikasi TOEFL berbasis android ini, tanpa adanya pengulangan. Berdasarkan permasalahan yang terjadi, penulis tertarik untuk melakukan penelitian ini dengan judul “Rancang Bangun Aplikasi Quiz Simulasi TOEFL Memanfaatkan Algoritma *Linear Congruential Generator* (LCG) Berbasis Android”

Permasalahan yang ada pada penelitian ini yaitu sebagai berikut. Masih sering terjadi kesalahan (*human error*) yang menyebabkan mahasiswa memiliki soal yang sama atau tidak teracak, mahasiswa tidak terbiasa dengan soal-soal yang ada dalam TOEFL, nilai/skor yang didapatkan saat mengikuti tes TOEFL belum maksimal. Tujuan dari penelitian ini yaitu sebagai berikut. Memberikan pengalaman yang baru terhadap mahasiswa dengan aplikasi simulasi TOEFL *online* berbasis android menggunakan pengacakan *linear congruential generator* (LCG), membantu mahasiswa dalam mempersiapkan diri dengan menyediakan contoh-contoh soal dan melihat hasil yang diperoleh sebelum mengikuti tes TOEFL yang sebenarnya, memberikan hasil nilai/skor yang maksimal pada mahasiswa yang mengikuti tes TOEFL. Manfaat dari penelitian ini yaitu sebagai berikut. Sebagai sarana pembelajaran simulasi TOEFL yang dapat diakses dengan mudah dan tanpa biaya (gratis), sebagai sarana latihan dan tolak ukur kepandaian mahasiswa dalam mengerjakan tes TOEFL, Sebagai modul pembelajaran berupa materi materi TOEFL yang memperdalam kemampuan TOEFL mahasiswa.

2. METODE

Dalam penelitian ini, penulis menggunakan metode pengumpulan data untuk memperoleh data primer dan data sekunder yang diperlukan, serta metode bagan alir (*flowchart*) sebagai pengembangan sistem untuk membangun aplikasi.



Gambar 1. Prosedur Perancangan *Flowchart*

Gambar 1. Terdapat beberapa aktivitas yang ada pada prosedur perancangan *diagram flowchart* yaitu sebagai berikut. (1) Analisa Kebutuhan, adalah aktivitas menganalisa kekurangan sistem yang sudah ada dengan menambahkan beberapa dokumen dan kebutuhan sistem baru untuk memahami permasalahan dan kekurangan dari sistem yang lama. Tahap ini diperlukan untuk menentukan data-data yang diperlukan dan *software* apa yang akan digunakan untuk mengolah data tersebut. (2) Desain Sistem Aplikasi, aktivitas desain sistem aplikasi dengan menganalisis dan menyusun gambaran mengenai sistem yang akan dibuat dengan tujuan untuk memperbaiki dan menyempurnakan sistem yang sudah ada. Perancangan sistem pada aplikasi TOEFL berbasis android ini menggunakan UML (*Unified Modeling Language*) yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. (3) Penulisan Kode Program, kode program atau biasa disebut *coding* merupakan bahasa yang digunakan sebagai penyempurna desain dengan bahasa yang dikenali oleh sistem komputer. Pada tahap ini akan dilakukan aktivitas pembuatan, pengujian, perbaikan (*debug*) yang dilakukan dengan kode kode pemrograman untuk membangun suatu sistem aplikasi. (4) Pengacakan LCG, aktivitas penerapan metode pengacakan LCG (*Linear Congruential Generator*) kedalam sistem yang sedang berjalan. Penerapan metode LCG digunakan untuk mengacak soal-soal yang telah dibuat menggunakan *coding* program. (5) Pengujian Program, aktivitas yang dilakukan dengan melakukan pengujian keseluruhan aplikasi secara berulang kali. Pengujian dilakukan dengan menggunakan metode *black box testing* yang berfokus pada hasil atau *output* tanpa memerlukan pengetahuan tentang *source code* aplikasi yang digunakan. Setelah pengujian sukses aplikasi dapat diimplementasikan dan digunakan oleh pengguna. (6) Pemeliharaan Sistem, aktivitas yang dilakukan untuk menjaga kestabilan sistem yang telah selesai dibuat. Pemeliharaan sistem atau *maintenance* diperlukan saat ingin menambah fitur-fitur baru yang bertujuan menarik perhatian pengguna baru. Dengan adanya *maintenance* dapat memperbaiki masalah atau *bug* yang dialami pengguna saat memakai aplikasi. Saat proses *maintenance* selesai pengguna bisa melakukan *update* dan menginstal ulang aplikasi pada situs resminya.

Metode *Linear Congruential Generator* (LCG)

Algoritma *Linear Congruential Generator* atau biasa disingkat LCG adalah sebuah algoritma pengacakan atau *shuffle* sederhana yang digunakan untuk menghasilkan sebuah bilangan acak (*Random Number*). Algoritma LCG merupakan metode *Pseudo Random Number Generator* (PRNG) atau pembangkit bilangan acak semu yang tertua dan paling populer. Algoritma LCG diperkenalkan pertama kali oleh seorang peneliti bernama D. H. Lehmer pada tahun 1951[6]. Metode LCG adalah metode yang digunakan untuk membangkitkan bilangan acak dengan distribusi *uniform*[7]. Metode LCG menggunakan algoritma yang bersifat rekursif linear yang dikombinasikan fungsi modulus[8]. Kemampuan LCG dalam memproses *random number* menjadi suatu bilangan acak yang tidak dapat diprediksi atau tidak menghasilkan bilangan yang sama sudah terbukti efektif dalam pengacakan soal atau pertanyaan yang berhubungan dengan *quiz* seperti TOEFL.

Bilangan acak adalah sebuah bilangan yang dihasilkan dari sebuah proses, yang keluarannya tidak dapat diprediksi dan secara berurutan tidak bisa dihasilkan bilangan yang sama. Penggunaan konstanta LCG (a , c , dan m) menentukan baik tidaknya bilangan yang akan diperoleh dalam perhitungan bilangan acak yang seakan-akan tidak terjadi pengulangan. Periode yang ada dalam LCG biasanya tidak melebihi m , dan sebagian besar persoalan yang melibatkan LCG, periodenya kurang dari itu. Intinya adalah deret bilangan acak yang dihasilkan tidak memiliki lebih banyak elemen daripada modulusnya. Kunci pembangkit adalah X_0 yang disebut *seed*. LCG mempunyai periode tidak lebih besar dari m , dan pada kasus periodenya kurang dari itu.

LCG mempunyai periode penuh $(m-1)$ jika memenuhi syarat yaitu sebagai berikut:

- b relatif prima terhadap m .
- $a-1$ dapat dibagi dengan semua faktor prima dari m .
- $m > \max(a, b, x_0)$ 4. $a > 0, b > 0$. [9]

Secara teori LCG mampu menghasilkan bilangan acak, namun sensitif terhadap pemilihan nilai a , b , dan m . Pemilihan nilai-nilai yang tidak sesuai dapat mempengaruhi implementasi pada LCG[10]. Keunggulan yang dimiliki oleh algoritma LCG terletak pada kecepatan operasinya, karena hanya memerlukan sedikit operasi bit. Kelemahan yang dimiliki oleh algoritma LCG adalah kumpulan nilai bilangan acaknya sering mudah diprediksi yang membuat metode ini tidak terlalu aman untuk kriptografi, tetapi LCG sangat berguna untuk aplikasi seperti simulasi dan enkripsi karena terbukti efektif dan efisien secara statistik.

Adapun perhitungan pengacakan LCG dapat didefinisikan melalui rumus-rumus berikut:

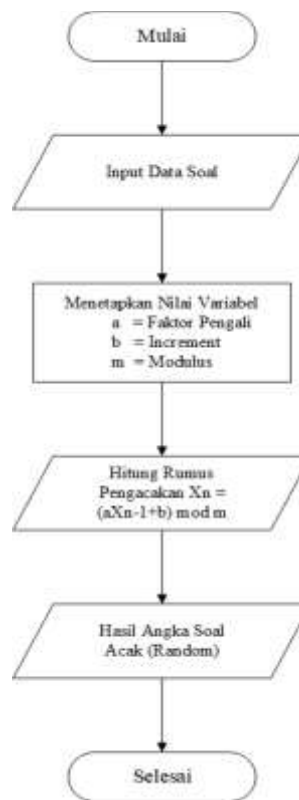
$$X_n = (aX_{n-1} + b) \bmod m \quad (1)$$

X_n = Bilangan acak ke- n dari deretnya
 X_{n-1} = Bilangan acak sebelumnya
 a = Faktor pengali
 b = Increment
 m = Modulus [11]

Secara umum, terdapat beberapa langkah penerapan algoritma *linear congruential generator* dalam pengacakan soal, yaitu sebagai berikut:

- 1) Menentukan nilai pembangkit yaitu a, b dan m , m adalah nilai acak atau jumlah yang diacak.
- 2) Mendefinisikan nilai awal atau *state* awal yaitu X_0 .
- 3) Proses pembangkitan bilangan acak dengan $X_n = (aX_{n-1} + b) \bmod m$ (2)
- 4) Menampilkan hasil bilangan acak. [12]

Adapun proses penggunaan metode LCG ini digambarkan dalam bentuk *diagram flowchart* sebagai berikut:



Gambar 2. Flowchart Diagram Penerapan LCG

Gambar 2. Setelah data soal yang akan digunakan di *input* dan nilai variabel a, b, m ditetapkan, langkah selanjutnya adalah proses pengacakan LCG menggunakan rumus: $X_n = (aX_{n-1} + b) \mod m$ (3) Setelah proses pengacakan selesai akan dihasilkan *output* berupa angka soal acak (*random*).

3. HASIL DAN PEMBAHASAN

A. Validasi Data

Studi kasus dalam melakukan perhitungan pada proses pengacakan soal menggunakan metode LCG dilakukan dengan beberapa tahap, mulai dari menentukan nilai X_0 yang diambil dari 3 angka terakhir kode soal yang ada pada tes TOEFL dan nilai m diambil dari jumlah data soal. Berikut ini merupakan tabel hasil proses dari pengacakan algoritma LCG dengan jumlah soal $m = 50$, $a = 10$, $c = 7$ dan X_n = tiga angka terakhir kode soal.

Tabel 1. Menentukan Umpan (X_0)

No.	Tes Toefl	Kode Soal	X_0 (Seed)
1.	<i>Listening Comprehension</i>	2022.001	1
2.	<i>Structure and Written Expression</i>	2022.002	2
3.	<i>Reading Comprehension</i>	2022.003	3

Untuk pengacakan soal yang ada pada *listening comprehension* yang diacak bukan keseluruhan soal tapi hanya jawaban pilihan bergandanya saja. Proses pengacakan soal yang ada pada *listening comprehension* dapat dilihat pada tabel 2. sebagai berikut:

Tabel 2. Pengacakan Soal *Listening Comprehension*

Peserta Toefl	Pengacakan Soal <i>Listening Comprehension</i>
1.	$X1 = (10*1+7) \bmod 50 = 17$
2.	$X2 = (17*1+7) \bmod 50 = 24$
3.	$X3 = (24*1+7) \bmod 50 = 31$
4.	$X4 = (31*1+7) \bmod 50 = 38$
5.	$X5 = (38*1+7) \bmod 50 = 45$
6.	$X6 = (45*1+7) \bmod 50 = 02$
7.	$X7 = (02*1+7) \bmod 50 = 09$
8.	$X8 = (09*1+7) \bmod 50 = 16$
9.	$X9 = (16*1+7) \bmod 50 = 23$
10.	$X10 = (23*1+7) \bmod 50 = 30$

Proses pengacakan soal yang ada pada *structure and written expression* dapat dilihat pada tabel 3. sebagai berikut:

Tabel 3. Pengacakan Soal *Structure and Written Expression*

Peserta Toefl	Pengacakan Soal <i>Structure and Written Expression</i>
1.	$X1 = (10*2+7) \bmod 50 = 27$
2.	$X2 = (27*2+7) \bmod 50 = 11$
3.	$X3 = (11*2+7) \bmod 50 = 29$
4.	$X4 = (29*2+7) \bmod 50 = 15$
5.	$X5 = (15*2+7) \bmod 50 = 37$
6.	$X6 = (37*2+7) \bmod 50 = 31$
7.	$X7 = (31*2+7) \bmod 50 = 19$
8.	$X8 = (19*2+7) \bmod 50 = 45$
9.	$X9 = (45*2+7) \bmod 50 = 47$
10.	$X10 = (47*2+7) \bmod 50 = 01$

Proses pengacakan soal yang ada pada *reading comprehension* dapat dilihat pada tabel 4. sebagai berikut:

Tabel 4. Pengacakan Soal *Reading Comprehension*

Peserta Toefl	Pengacakan Soal <i>Reading Comprehension</i>
1.	$X1 = (10*3+7) \bmod 50 = 37$
2.	$X2 = (37*3+7) \bmod 50 = 18$
3.	$X3 = (18*3+7) \bmod 50 = 11$
4.	$X4 = (11*3+7) \bmod 50 = 40$
5.	$X5 = (40*3+7) \bmod 50 = 27$
6.	$X6 = (27*3+7) \bmod 50 = 38$
7.	$X7 = (38*3+7) \bmod 50 = 21$
8.	$X8 = (21*3+7) \bmod 50 = 20$
9.	$X9 = (20*3+7) \bmod 50 = 17$
10.	$X10 = (17*3+7) \bmod 50 = 08$

Adapun hasil dari proses pengacakan algoritma LCG yang ada pada kepada 10 peserta tes simulasi TOEFL dapat dilihat pada tabel 5. sebagai berikut:

Tabel 5. Hasil Pengacakan Soal TOEFL

No.	Tes Toefl	Soal Acak (<i>Random Number</i>)
------------	------------------	---

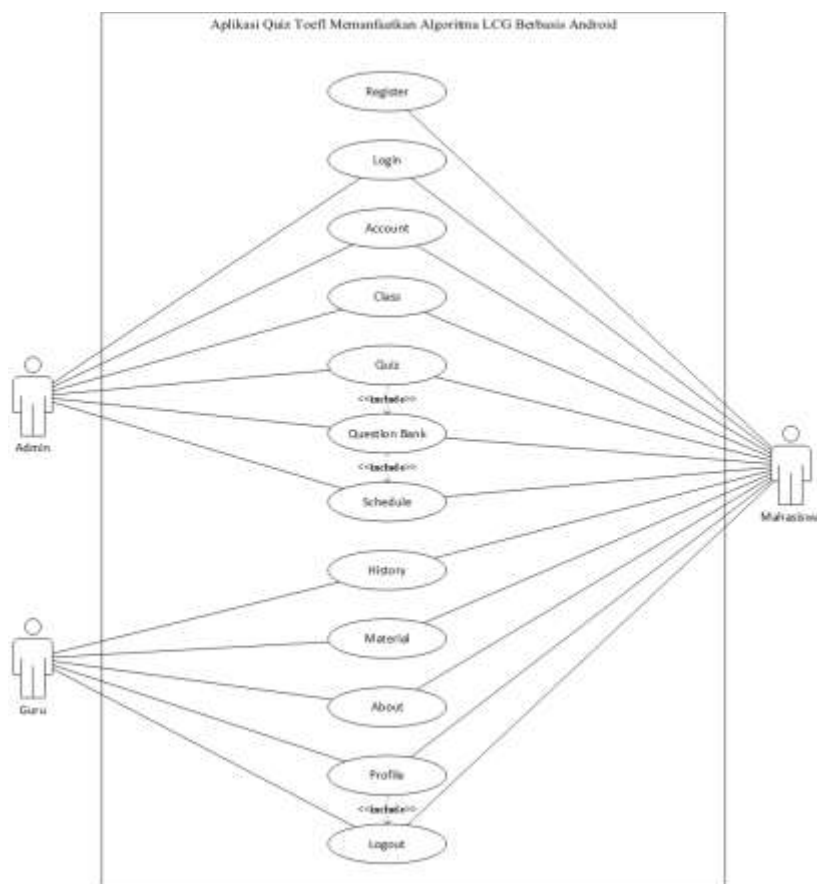
1.	<i>Listening Comprehension</i>	17, 24, 31, 38, 45, 02, 09, 16, 23, 30
2.	<i>Structure and Written Expression</i>	27, 11, 29, 15, 37, 31, 19, 45, 47, 01
3.	<i>Reading Comprehension</i>	37, 18, 11, 40, 27, 38, 21, 20, 17, 08

B. Desain Sistem

Desain sistem merupakan tahap yang menentukan proses dan data yang diperlukan oleh sistem baru[13]. Desain sistem dalam aplikasi *quiz* simulasi TOEFL memanfaatkan algoritma *linear congruential generator* (LCG) berbasis android dengan menggunakan *pemodelan Unified Modeling Language* (UML) yang mencakup beberapa *diagram* rancangan, seperti *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram*.

1) Use Case Diagram

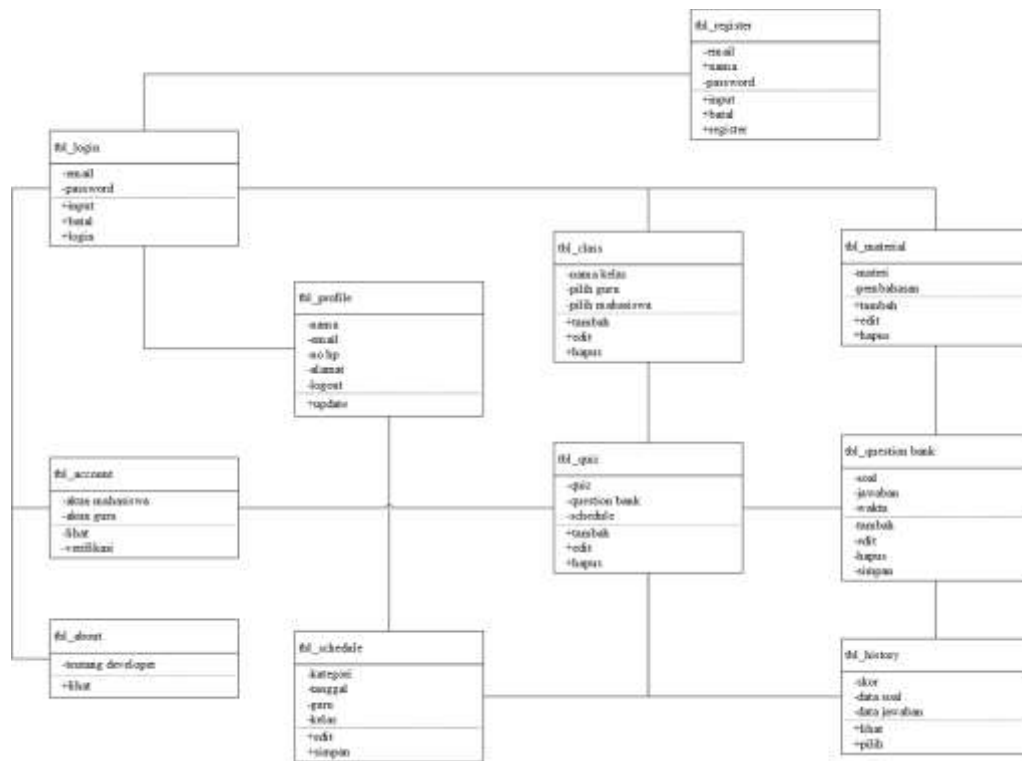
Use case diagram adalah *diagram use case* yang digunakan untuk menggambarkan secara ringkas siapa saja yang menggunakan sistem dan sistem apa saja yang digunakan. *Use Case Diagram* menjabarkan aktor yang terlibat dan hal-hal yang dapat dilakukan pada sistem. [14] Berikut adalah desain rancangan *use case diagram* dalam aplikasi *quiz* simulasi TOEFL, yang dapat dilihat pada gambar 1. sebagai berikut:



Gambar 1. Use Case Diagram

2) Class Diagram

Class diagram atau *diagram kelas* adalah salah satu jenis *diagram* struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek[15]. Berikut adalah desain rancangan *class diagram* dalam aplikasi *quiz* simulasi TOEFL, yang dapat dilihat pada gambar 2. sebagai berikut:



Gambar 2. Class Diagram

C. Implementasi

Pada implementasi akan memaparkan tampilan-tampilan menu yang ada pada aplikasi *quiz* simulasi TOEFL memanfaatkan algoritma *linear congruential generator* (LCG) berbasis android. Sehingga dapat dilakukan uji coba hasil dan mengetahui kelebihan serta kekurangan yang ada pada sistem aplikasi tersebut. Aplikasi ini bertujuan untuk menampilkan soal-soal *quiz* simulasi TOEFL yang telah diacak menggunakan metode *linear congruential generator* (LCG). Aplikasi ini dibuat menggunakan *software* android studio dan bahasa pemrograman *dart* dengan *database* aplikasinya yaitu *firebase*.

1) Form Login

Form login pada aplikasi memiliki fungsi sebagai keamanan sistem dengan memverifikasi setiap *user* yang mencoba untuk masuk ke dalam aplikasi. Pada aplikasi ini memiliki 3 hak akses *user* yang dapat melakukan *login* kedalam aplikasi yaitu admin, mahasiswa dan guru. *Form login* dapat dilihat pada gambar 3. sebagai berikut:



Gambar 3. *Form Login*

2) *Form Register*

Form register digunakan oleh *user* untuk melakukan pendaftaran akun baru pada aplikasi. Setelah berhasil mendaftar akun, *user* dapat menunggu *approve* dari admin sebelum dapat melakukan *login* kedalam aplikasi menggunakan akun tersebut. *Form register* dapat dilihat pada gambar 4. sebagai berikut:

Gambar 4. *Form Register*

3) *Menu Class*

Menu class adalah menu yang digunakan oleh admin untuk membuat dan menambahkan kelas pada tampilan admin. *Menu class* berisi data guru dan mahasiswa yang sedang mengikuti kelas TOEFL aktif. *Menu class* dapat dilihat pada gambar 5. sebagai berikut:

Gambar 5. Menu *Class*

4) *Form Quiz*

Form quiz merupakan menu yang digunakan oleh mahasiswa untuk mengerjakan *quiz* simulasi TOEFL yang ada pada aplikasi. *Quiz* yang ditambahkan oleh admin akan muncul pada tampilan mahasiswa dan guru akan memantau setiap mahasiswa yang akan mengerjakan *quiz* simulasi TOEFL pada aplikasi. *Form quiz* yang ada pada mahasiswa dapat dilihat pada gambar 6. sebagai berikut:

Gambar 6. *Form Quiz*

5) *Form Question Bank*

Form question bank merupakan menu yang digunakan oleh admin untuk membuat dan menambahkan soal *quiz* simulasi TOEFL pada tampilan *user*. Tampilan *question bank* berisi data soal dan waktu durasi dari pengerjaan *quiz* tersebut. *Form question bank* dapat dilihat pada gambar 7. sebagai berikut:

Gambar 7. Form Question bank

6) Form Schedule

Form schedule merupakan menu yang digunakan oleh admin untuk membuat dan menambahkan jadwal *quiz* TOEFL pada tampilan *user*. *Formschedule* berisi data tanggal dan waktu durasi dari pengerjaan *quiz* tersebut. *Form schedule* dapat dilihat pada gambar 8. sebagai berikut:

Gambar 8. Form Schedule

7) Menu History

Menu *history* merupakan menu yang digunakan oleh guru untuk mengecek hasil pengerjaan *quiz* mahasiswa pada tampilan *user*. Menu *history* berisi data nilai/skor dan riwayat-riwayat soal benar atau salah dari pengerjaan *quiz* tersebut. Menu *history* dapat dilihat pada gambar 9. sebagai berikut:



Gambar 9. Menu History

8) Form Material

Form material merupakan menu yang digunakan oleh guru untuk membuat dan menambahkan materi-materi TOEFL pada tampilan *user*. Tampilan *material* berisi data materi dan pembahasan yang berhubungan dengan TOEFL. *Form material* pada guru dapat dilihat pada gambar 10. sebagai berikut:



Gambar 10. Form Material

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan maka penulis menarik beberapa kesimpulan yaitu sebagai berikut. (1) Aplikasi *quiz* simulasi TOEFL berbasis android memanfaatkan algoritma *Linear Congruential Generator* (LCG) dengan *online system* yang dapat menghasilkan soal-soal acak sehingga mahasiswa tidak dapat bekerja sama pada saat mengerjakan *quiz* TOEFL. (2) Pemaparan materi dan uraian contoh soal yang ada dalam aplikasi dapat menjadi simulasi dan media pembelajaran bagi mahasiswa yang hendak mengikuti tes TOEFL. (3) Sistem aplikasi yang dapat mengelompokkan data skor berdasarkan jenis soal berupa *listening comprehension*, *structure and written expression* dan

reading comprehension sebagai uji kemampuan mahasiswa untuk mendapatkan skor yang tinggi saat mengikuti tes TOEFL yang sebenarnya.

Adapun kelebihan sistem yang ada pada aplikasi *quiz* TOEFL ini yaitu sebagai berikut. (1) Aplikasi ini memberi pengguna akses mudah dan fleksibel untuk berlatih soal TOEFL kapan saja, di mana saja melalui perangkat android mereka. (2) Penggunaan algoritma *linear congruential generator* (LCG) memastikan pembangkitan urutan soal yang acak dan variasi yang konsisten, serta meningkatkan pengalaman belajar. (3) Hasil ujian simulasi TOEFL dapat memberikan umpan balik segera kepada pengguna, membantu mereka untuk mengidentifikasi kelemahan dan memperbaiki performa mereka. Sedangkan kekurangan sistem yang ada pada aplikasi *quiz* TOEFL ini yaitu sebagai berikut. (1) Tersedia di *platform* android saja, sehingga mungkin tidak mencakup semua pengguna potensial yang menggunakan *platform* lain. (2) Diperlukan koneksi internet untuk mengunduh soal soal atau memperbarui aplikasi, yang bisa menjadi kendala bagi pengguna dengan akses terbatas

Berdasarkan kesimpulan dan hasil penelitian yang telah diuraikan diatas, maka penulis memberikan saran-saran yang kiranya dapat membantu penelitian dan pengembangan aplikasi *quiz* TOEFL berbasis android ini pada masa yang akan datang. Adapun saran-saran penulis dalam penelitian ini yaitu sebagai berikut. (1) Untuk kedepannya, aplikasi *quiz* simulasi TOEFL berbasis android ini dapat berkembang menjadi sebuah sarana ujian pada lembaga TOEFL berstandar nasional. (2) Untuk kedepannya, aplikasi ini dapat dikembangkan untuk perangkat selain android. (3) Untuk kedepannya, dapat digunakan algoritma pengacakan lainnya untuk membuat aplikasi TOEFL. (4) Untuk kedepannya, dapat digunakan bahasa pemrograman yang lain selain *dart*.

UCAPAN TERIMA KASIH

Pada kesempatan ini perkenalkanlah penulis untuk menyampaikan rasa terima kasih kepada Universitas Potensi Utama yang memberikan kesempatan bagi penulis untuk mengembangkan dan menyelesaikan penelitian ini dan dosen pembimbing saya dalam penelitian ini ibu Lili Tanti, M. Kom terima kasih atas bimbingannya selama ini. Terima Kasih juga kepada orang tua saya yang telah memberikan doa, masukan, dan dukungan dalam penyelesaian penelitian ini.

REFERENSI

- [1] Rusdiyanto, Hakim, L., & Martadinata, A. T. (2022). Aplikasi Ujian Online dan Penerapan Algoritma LCG Untuk Proses Pengacakan Soal Ujian Di SMK Negeri Tugumulyo. JUTIM (Jurnal Teknik Informatika Musirawas), Vol. 7(2), 99–108.
- [2] Azharuddin, M. (2021). Penerapan Scoring System Dalam Penilaian Latihan Pemahaman Materi Toefl. Terapan Informatika Nusantara, 1(8), 395–401.
- [3] Ashari, M. I., & Ardiansyah, H. (2022). Perancangan Aplikasi Simulasi Pelatihan TOEFL Berbasis Android Menggunakan Metode Prototype Pada Lembaga Bahasa Universitas Pamulang. Journal of Artificial Intelligence and Innovative Applications (JOAIIA), 3(1), 9–15.
- [4] Sukmawan, R., Susilawati, L., & Hestiana, S. (2022). Peningkatan Kemampuan Pemahaman Toefl Mahasiswa Teknik Sipil Universitas Muhammadiyah Sukabumi. Jurnal Pengabdian Kepada Masyarakat Nusantara, 2(2), 41–52.
- [5] Samsuriati, N., Saputra, R. A., & Pramono, B. (2019). Aplikasi Ujian Toefl Menggunakan Metode Linear Congruential Generators (Lcg) Berbasis Web Pada Universitas Halu Oleo. Prosiding, 233–242.

- [6] Triwibowo, D. N., -, P., Ashari, I. A., Sandi, A. S., & Rahman, Y. F. (2022). Enkripsi Pesan Menggunakan Algoritma Linear Congruential Generator (LCG) dan Konversi Kode Morse. *Buletin Ilmiah Sarjana Teknik Elektro*, 3(3), 194–201.
- [7] Fanani, A. (2021). Pengacakan Soal Pada Sistem Computer Based Test (Cbt) Dengan Metode Linear Congruential Generator (Lcg) Di Sma Negeri Jogoroto. *Jurnal Ilmiah Teknologi Informasi Dan Sains*, 1(2001), 50–56.
- [8] Alamsyah, A., Shofi, I. M., & Suseno, H. B. (2021). Prototipe Sistem Computer Based Test dengan Pengacakan Soal Menggunakan Metode Fisher-Yates Shuffle. *Prototype of Computer Based Test System with Question Randomization Using Fisher-Yates Shuffle Method*. 8, 81–89.
- [9] Royani, R., & Darusalam, U. (2022). Media Pembelajaran Smart Quiz Berbasis Android menggunakan Metode Congruential Generator. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 7(2), 419–428.
- [10] Syah, R., Zarkasi, A., Hormansyah, D. S., & Wibowo, D. W. (2020). IMPLEMENTASI METODE FUZZY MAMDANI DAN LINEAR CONGRUENTIAL GENERATOR (LCG) PADA GAME HIDDEN. *Jurnal Informatika Polinema (JIP)*, 6, 23–30.
- [11] Prakarsa, A., Sunarto, A. A., & Prajoko, P. (2020). Model Pengacakan Soal Ujian Online SMA Menggunakan Metode Linear Congruential Generator dan Fisher Yates. *Progresif: Jurnal Ilmiah Komputer*, 16(2), 133.
- [12] Sumiah, A., Hakim, R. R. (2021). Implementasi Metode Linear Congruential Generator pada Game Puzzle Kesenian Tari. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 8(1), 129–140.
- [13] Rosmalasari, T. D., Lestari, M. A., Dewantoro, F., & Russel, E. (2020). Pengembangan E-Marketing Sebagai Sistem Informasi Layanan Pelanggan Pada Mega Florist Bandar Lampung. *Journal of Social Sciences and Technology for Community Service (JSSTCS)*, 1(1), 27.
- [14] Utama, I. P. Y., & Supriana, I. W. (2019). Sistem Informasi Management Absensi Pegawai CV. Avatar Solution. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 7(3), 169.
- [15] Prasetya, A. F., Sintia, & Putri, U. L. D. (2022). Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language). *Jurnal Ilmiah Komputer Terapan Dan Informasi*, 1(1), 14–18.