International Journal of Informatics and Computation (IJICOM)
Vol. 7, No. 2, 2025
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Fisher-Yates Shuffle and Linear Congruent Algorithm in the Mini Challenge of Computational Thinking Task

Kartarina[1*], Octa Travelian[2], Moch. Syahrir[3], Nuratun Islamiah[4], Wahyu Kamil Syarifaturrahman[5]

Abstract

Computational Thinking (CT) is one of the crucial skills in supporting the processes of problem formulation and problem solving. Further, in exploring the skills of everyone from an early age, the Bebras Bureau of Bumigora University regularly organizes the Bebras Challenge as an extracurricular educational activity for students at various educational levels. Nevertheless, both conventional and computer-based tests still face challenges, particularly related to cheating due to the sequential presentation of questions. To address this issue, the Bebras Learning Management System (LMS) was developed, featuring online testing with randomized questions to minimize cheating. The system was designed using Computer-Based Testing (CBT) and implemented with the Fisher-Yates Shuffle Algorithm and the Linear Congruent Method (LCM), which function to randomize question order and thereby reduce the possibility of cheating among students. This research employed a methodology consisting of data collection, system design, data retrieval, algorithm implementation, testing, and evaluation. The results indicate that the application of the Fisher-Yates Shuffle and LCM algorithms in question selection and randomization produced variations with different levels of correlation, namely: No Correlation (38%), Weak Correlation (29.5%), Moderate Correlation (19%), Strong Correlation (12%), and Perfect Correlation (1.5%). We conducted the testing stage at the Bebras LMS, which achieved a final performance outcome of 86%.

Keywords:
CBT, Randomize, Fisher-Yates Shuffle, Linear Congruent Generator

1. Introduction

The rapid advancement of digital technology has led to the increasing use of computer-based testing (CBT) systems and educational applications as effective tools for assessment, learning, and knowledge dissemination. Compared to traditional paper-based exams, CBT offers efficiency in administration, faster scoring, and broader accessibility for learners. However, one persistent issue in the development of CBT applications is ensuring fairness and unpredictability in the randomization of test questions. Randomization prevents cheating and memorization patterns, but poor implementation of randomization algorithms can lead to biased outcomes, repeated patterns, or security vulnerabilities in the testing process [13], [14], [15].

Random number generation (RNG) methods play a crucial role in ensuring question randomization within CBT and educational games. Among the most widely used

---

Corresponding Author: Kartarina (kartarina@universitasbumigora.ac.id)
1 Kartarina, Universitas Bumigora, kartarina@universitasbumigora.ac.id
2 Okta Travelian, Universitas Bumigora, octatravelian@gmail.com
3 Moch. Syahrir,Universitas Bumigora, muhammadsyahriralfath@gmail.com
4. Nuratun Islamiah, Universitas Bumigora, Nuratunislamiah26@gmail.com
5. Wahyu Kamil, S., Universitas Bumigora, wahyu.kamil@universitasbumigora.ac.id

approaches are the Linear Congruential Generator (LCG) and the Fisher-Yates Shuffle (FYS) algorithm. The LCG is one of the earliest RNG methods that is simple to implement and computationally efficient, but it is often criticized for generating predictable sequences if the seed value or modulus is poorly chosen [6], [17], [18]. In contrast, the Fisher-Yates Shuffle offers an unbiased method for shuffling elements, making it suitable for randomizing exam questions and game-based learning content [7], [8], [20]. The challenge, therefore, lies in selecting and implementing the right algorithm that ensures both computational efficiency and high-quality randomness in education-related applications.

Several studies have shown that the implementation of Fisher-Yates Shuffle in educational software successfully prevents repeated patterns in randomized questions, thereby increasing fairness in learning assessments. Research by Asih et al. [7] and Rohmah et al. [8] demonstrated that FYS could effectively be applied to Android-based quiz games and CBT applications, resulting in more equitable learning environments. Similarly, Aini and Wijaya [10], [20] highlighted the importance of unbiased randomization in goal-oriented educational games, where fairness in gameplay directly affects learning outcomes. These findings underline that the problem of question predictability can be substantially mitigated by careful implementation of shuffle-based algorithms.

On the other hand, the Linear Congruential Generator continues to be explored due to its simplicity and suitability for mobile platforms, despite its weaknesses. Zaid et al. [6] developed an Android-based IQ test application using the LCG algorithm, proving that it could be applied to lightweight mobile apps. However, Hangga and Prabowo [17] emphasized that modifications are often required to improve randomness and security in LCG-based systems. Supardi and Putra [18] similarly applied LCG for Android-based game design but noted limitations in unpredictability compared to shuffle-based approaches. These studies suggest that while LCG remains useful, it requires adaptation or combination with other techniques to overcome its inherent weaknesses.

Beyond technical algorithm choices, educational frameworks such as computational thinking and gamification also influence how randomization is applied. Apriani et al. [1] and Tresnawati et al. [3] argued that computational thinking skills can be fostered through problem-solving applications, particularly when supported by fair and unpredictable digital assessment tools. Similarly, Assaf et al. [2] emphasized the role of problem decomposition in computational thinking, linking it directly to randomized testing environments where learners must adapt to new problem sets rather than memorizing repeated tasks. This suggests that the technical problem of randomization is also pedagogically relevant to how students develop critical skills.

The role of system modeling and software design methodologies also becomes central in developing robust CBT and randomization systems. Tools such as Unified Modeling Language (UML) are commonly used for structuring educational software projects, ensuring that the randomization features are well-integrated into the system architecture. Studies by Dharwiyanti and Wahono [12], Sulistyorini [21], and Putra and Andriani [23] demonstrate how UML-based design methodologies can enhance software clarity, reducing the likelihood of design flaws in question-randomization modules. This highlights that algorithmic choices alone are insufficient; instead, robust software engineering principles are also required.

Another pressing issue in CBT implementation is related to fairness in access and the effectiveness of assessments. Lestari [13] found that while CBT enhances efficiency in history education, concerns persist regarding whether randomized questions are appropriately distributed across different levels of difficulty. Similarly, Utama and Asriningtias [19] compared FYS and LCG algorithms in web-based try-out systems, concluding that while both methods can randomize questions, FYS outperforms LCG in terms of unbiased access times and fairness. Such studies illustrate that algorithm choice directly impacts students' experiences and perceptions of fairness in assessments.

Finally, the integration of randomization algorithms into mobile and web-based learning platforms highlights broader implications for educational technology adoption. Research on mobile English learning apps [9], [16] and culinary or algorithm learning systems [11], [24] shows that randomization enhances user engagement, prevents predictability, and supports adaptive learning environments. As more educational institutions embrace digital assessment and game-based learning, ensuring that the underlying algorithms for question distribution are both technically sound and pedagogically effective becomes increasingly important [5], [7], [8], [18]. This forms the foundation for exploring hybrid approaches that combine algorithmic robustness, software engineering principles, and computational thinking frameworks in the development of next-generation CBT systems.

2. Related Works

Randomization in computer-based testing (CBT) has been studied extensively to improve fairness and reduce cheating opportunities. Lestari [13] explored the use of CBT as an evaluation tool in high school history subjects, highlighting its ability to streamline assessment processes and reduce the logistical burden compared to traditional exams. However, the study also noted limitations in question randomization, where improper distribution of question difficulty could affect fairness among test takers. This finding emphasizes the importance of robust algorithms in ensuring the equitable distribution of randomized questions within CBT systems.

The Fisher-Yates Shuffle (FYS) algorithm has been widely applied to educational applications to ensure unbiased shuffling of items. Asih et al. [7] developed an Android-based examination system using FYS, which successfully randomized questions without creating patterns. Similarly, Rohmah et al. [8] applied FYS to a quiz game environment, showing that the algorithm provided uniform randomness across iterations. While these works highlight the strengths of FYS in producing unbiased results, they did not address computational performance at scale, which can become an issue in large CBT deployments with thousands of users.

Several works have also compared the efficiency of FYS with alternative methods such as the Linear Congruential Generator (LCG). Utama and Asriningtias [19] conducted a comparative study of FYS and LCG in web-based try-out systems, concluding that while both methods can effectively randomize questions, FYS achieved better fairness and lower bias in distribution. The limitation, however, lies in the increased computational time of FYS when processing large datasets, which can impact system response times. This tradeoff between fairness and performance remains a key issue for CBT developers.

Despite its limitations, LCG continues to be employed due to its simplicity and ease of implementation. Zaid et al. [6] implemented LCG in an Android-based IQ test application, showing that the algorithm was lightweight and suitable for mobile environments. Likewise, Supardi and Putra [18] applied LCG in an Android-based game application, demonstrating its ability to generate pseudo-random outcomes with low resource requirements. Nonetheless, both studies acknowledged that LCG sequences can become predictable if the seed values are exposed, raising security concerns in high-stakes testing environments.

Efforts to improve LCG's predictability have been reported by Hangga and Prabowo [17], who proposed modifications to the standard LCG formula to increase randomness quality. Their work showed improvements in unpredictability, making it more competitive with shuffle-based algorithms. However, the modifications added complexity and required additional computational resources, diminishing one of LCG's primary strengths—its simplicity. This suggests that while LCG can be adapted, it may still be less ideal than shuffle-based algorithms for certain educational applications where fairness is a priority.

Beyond algorithmic approaches, research has also linked question randomization to broader educational frameworks such as computational thinking. Apriani et al. [1]

emphasized the importance of computational thinking in primary education, demonstrating that digital tools that randomize problem sets can foster problem-solving and critical thinking skills. Similarly, Assaf et al. [2] introduced a mobile app for assessing decomposition skills in computational thinking, underscoring the role of randomized and adaptive tasks. While these studies did not directly focus on randomization algorithms, they show that effective randomization contributes to the development of essential 21st-century skills.

From a software engineering perspective, design methodologies such as Unified Modeling Language (UML) have been employed to structure CBT systems and ensure robustness in the integration of randomization features. Studies by Dharwiyanti and Wahono [12], Sulistyorini [21], and Putra and Andriani [23] highlighted UML's role in clarifying system workflows and preventing design flaws. While these works do not provide new randomization algorithms, they emphasize that poor system design can undermine even the strongest algorithmic approaches, pointing to the need for careful integration of randomization within the broader CBT architecture.

Finally, educational applications beyond CBT also demonstrate the relevance of randomization algorithms. Desliyanti et al. [9] applied LCG in a mobile English-learning app, while Hindarto [11] and Maulana [24] implemented digital learning platforms that relied on randomized tasks to increase learner engagement. These applications highlight randomization as a cross-cutting issue in educational technology. However, most of these studies focus on specific applications rather than the comparative performance of algorithms, leaving a gap in the systematic evaluation of randomization methods across multiple contexts.

3. Proposed Method

3.1. Research Design

Research Flow which consists of several phases. In the initial phase, the researchers collected data in the form of Bebras challenge questions. This phase also involved outlining the stages of data processing. Subsequently, the researchers designed a question randomization system and implemented the Fisher-Yates Shuffle (FYS) and Linear Congruential Generator (LCG) algorithms. This web-based randomization system was developed using the Laravel programming language. After the system design and implementation stage, the next phase involved the initialization process, which consisted of collecting computational thinking (CT) tasks and determining the parameters of the LCG. This was followed by the execution of the Fisher-Yates Shuffle process, where the randomization of questions was carried out using the random numbers generated by the LCG. The outcomes produced by the algorithms were then obtained as the output of the system. Finally, the research concluded with the testing and evaluation stage to assess the effectiveness and performance of the proposed randomization system. Fig. 1 depicts the research flow or stages of the study.

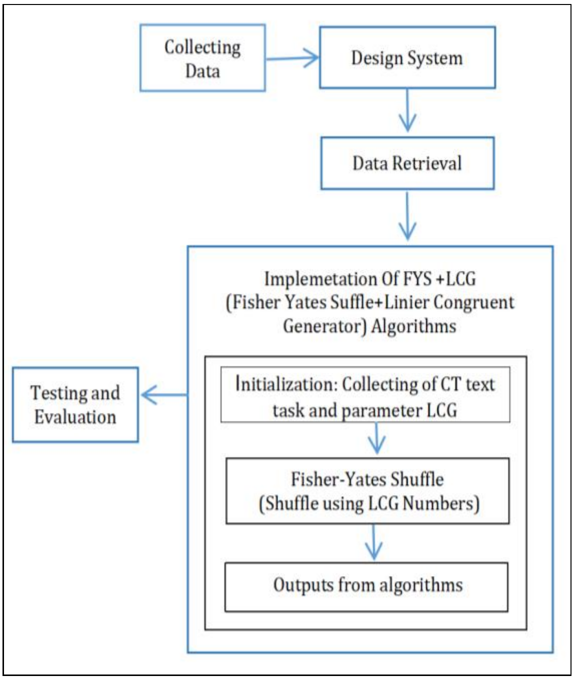


Fig. 1 Research Flow

3.2. Implementation of Fisher-Yates shuffle and linear congruent method

In this research, a quiz or test in the CBT model in each answer response is stored, scored, or both electronically. As the name suggests, this computer-based assessment uses computers or electronic devices to evaluate students' learning outcomes. This approach allows teachers or instructors to manage, schedule, and administer exams, as well as manage and report data by utilizing computer technology. The method applied in the Computer-Based Test is the Managed Mode method. Randomization algorithms play an important role in the application. These algorithms are used to shuffle the question order in each test session, ensuring that each participant receives a different question order. Two commonly used randomization algorithms in web-based CBT are Fisher-Yates Shuffle and Linear Congruent Method. In the application, the Linear Congruent Method is used to generate a sequence of random numbers that serve as the seed in the Fisher-Yates Shuffle. By combining the Linear Congruent Method with Fisher-Yates Shuffle, the question order in each test session will be randomized to enhance security and fairness in the execution of the test.

The Fisher–Yates Shuffle algorithm is a well-known method for generating an unbiased random permutation of a finite sequence. The process begins by initializing an array containing all the elements to be shuffled. Starting from the last element of the array, a random index is selected from the range of indices up to the current position. The element at the current position is then swapped with the element at the randomly selected index. This process of selecting a random index and performing a swap is repeated for each preceding element in the array until every element has been processed. By the end of this iterative procedure, all elements have been randomly rearranged, ensuring that each possible permutation of the array is equally likely to occur.

The final result is the shuffled array. The use of Fisher-Yates Shuffle in web-based CBT provides variation in the question order for each test session, thus preventing cheating and ensuring fairness in the execution of the test. The *preprocessing* of random number generation by applying the LCM method can be seen below:

$$x_i = (a.x_0 + c)\, mod\, m \tag{1}$$

Where:
x_i = resulting sequence of pseudo-random numbers.
a = multiplier.
x_0 = initial value or seed.
c = an addition
m = modulus

In the Linear Congruential Method (LCM), also known as the Linear Congruential Generator (LCG), the sequence of pseudo-random numbers is generated using the recurrence relation $X_{n+1} = (aX_n + c) \mathrm{mod}\ m$, where a is the multiplier, c is the increment, m is the modulus, and X_0 is the initial seed. To ensure the generator produces a long and well-distributed sequence, several conditions must be met: the multiplier a must be smaller than the modulus m to prevent arithmetic overflow and maintain correct modular arithmetic; the increment c should be an odd number, while m is preferably a power of two to guarantee a full period cycle, ensuring all possible values appear before repetition; moreover, c must not be a multiple of m to avoid short cycles. The modulus m is ideally chosen as a prime number to maximize the period and improve the statistical uniformity of the sequence. Lastly, the initial seed X_0 must be an integer within the range $0 \leq X_0 < m$, and ideally co-prime with m, since an inappropriate seed value may reduce randomness or cause early repetition of the generated sequence.

Correlation analysis is a statistical tool used to assess the degree of linear relationship between two variables. The main difference and essence between correlation analysis and regression analysis lie in their purposes. Correlation analysis is used to determine the direction and strength of the relationship between two variables, while regression analysis is used to predict how much change in the dependent variable occurs due to changes in the independent variable.

In this study, the Pearson correlation method is used to evaluate the linear relationship between two variables. The Pearson correlation coefficient, first introduced by Karl Pearson in 1900, has a value range between -1, 0, and 1. A value of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, while a value of 0 indicates no correlation. This method is specific to measuring linear relationships, and if the relationship between the variables is not linear, this method is not optimal. The Pearson correlation requires interval or ratio-scaled data and assumes a normal distribution. There are two symbols for the Pearson correlation: "p" for population measurement and "r" for sample measurement. The variables involved in this method include the dependent variable (X) and independent variables $(Y1, Y2, Y3, \ldots)$.

The Pearson correlation itself has two forms: Pearson product-moment correlation and Spearman rank correlation, which have different classifications and uses. The variables used, X and Y, may or may not be related, according to the Pearson correlation analysis [15].

During the question randomization process, the questions will first be selected using the Linear Congruent Method algorithm. The authors use the Linear Congruent Method (LCM) algorithm with the following parameters, as shown in the example below:

Seed x_0 : 12345
Multiplier a : 1664525
Increment c : 1013904223
Modulus m : 2^32

Table 1 describes a system that generates a sequence of randomly selected questions using this algorithm:

Iteration	Equations	Score of Modulus $2^{32} = 1283620776$
X_1	$(1664525 \times 12345 + 1013904223) \bmod 2^{32}$	1283620776
X_2	$(1664525 \times 1283620776 + 1013904223) \bmod 2^{32}$	1283620776
X_3	$(1664525 \times 1923468610 + 1013904223) \bmod 2^{32}$	1283620776
X_4	$(1664525 \times 1923468610 + 1013904223) \bmod 2^{32}$	1283620776
X_5	$(1664525 \times 1923468610 + 1013904223) \bmod 2^{32}$	1283620776
X_6	$(1664525 \times 1923468610 + 1013904223) \bmod 2^{32}$	1283620776
X_7	$(1664525 \times 1923468610 + 1013904223) \bmod 2^{32}$	1283620776
X_8	$(1664525 \times 1923468610 + 1013904223) \bmod 2^{32}$	1283620776
X_9	$(1664525 \times 1923468610 + 1013904223) \bmod 2^{32}$	1283620776
X_{10}	$(1664525 \times 1923468610 + 1013904223) \bmod 2^{32}$	1283620776

Next, the system selects 10 questions with the following numbers: 16, 9, 4, 12, 5, 11, 13, 7, 15, 18. After selecting the questions using the Linear Congruent Method algorithm, the system will then shuffle the order of the previously selected questions (16, 9, 4, 12, 5, 11, 13, 7, 15, 18). The steps for shuffling are as follows:

- Exchange the question number 16 with question number 7.
  (7, 9, 4, 12, 5, 11, 13, 16, 15, 18)
- Exchange question number 9 with question number 12.
  (7, 12, 4, 9, 5, 11, 13, 16, 15, 18)
- Exchange question number 5 with question number 16
  (7, 12, 4, 9, 16, 11, 13, 5, 15, 18)
- Exchange the question number 11 with question number 13
  (7, 12, 4, 9, 16, 13, 11, 5, 15, 18)
- Exchange the question number 18 with question number 16
  (7, 12, 4, 9, 18, 13, 11, 5, 15, 16)
- Exchange the question number 13 with question number 18
  (7, 12, 4, 9, 13, 18, 11, 5, 15, 16)

After the shuffling is completed, the system obtains the following sequence of questions: 7, 12, 4, 9, 13, 18, 11, 5, 15, 16. This sequence will then be displayed for the students to work on.

4. Experimental Setup

The question randomization system in this mini-challenge application was developed using the Unified Modeling Language. The Unified Modeling Language (UML) was used in this study to analyze the system. It served to illustrate the various procedures and processes involved in the functioning system. UML contains graphical notations for designing, modeling, and documenting software systems in this system.

4.1. Use Case Diagram

A Use Case is a visual representation of information system modeling. This diagram explains the relationship between one or more roles. A Use Case diagram can also be used to identify the functions within a system and determine who has the right to access and use these functions. In this study, there are four actors who can access or use the new system: admin, volunteer, teacher, and student. Fig.2 illustrates a Use Case diagram of the study.

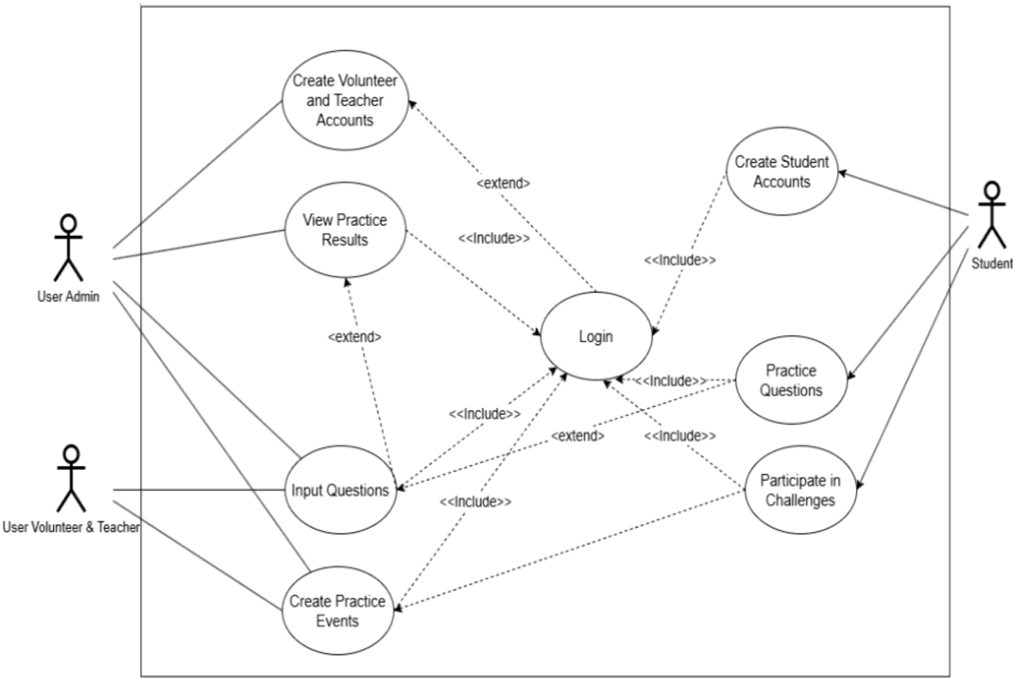


Fig. 2 Use Case Diagram

4.2. Sequence Diagram

A *Sequence Diagram* illustrates object interactions organized in a time sequence. It depicts the order of processes or stages that must be executed to achieve a specific outcome, as outlined in the *Use Case* diagram. Fig. 3 presents the *Sequence Diagram* of the system, where challenge participants initiate the login process by entering their email and password. The system then validates the provided credentials, and upon successful verification, participants are granted access to join the event created by the administrator.

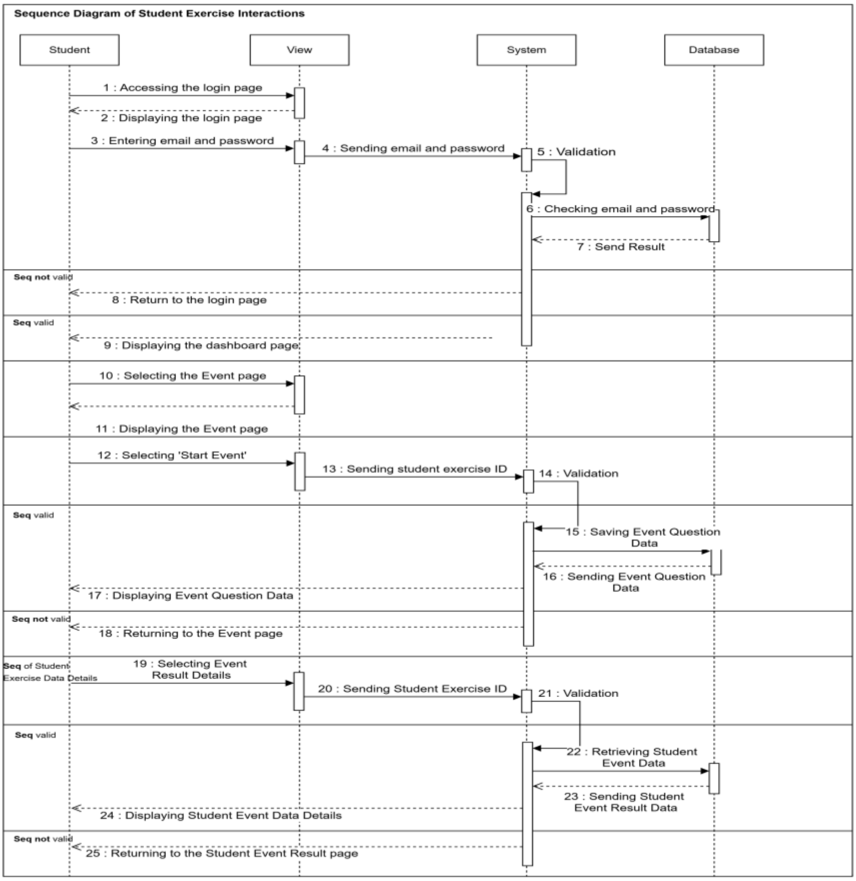


Fig. 3 Sequence Diagram

4.3. Activity Diagram

An Activity Diagram is a visual representation of the sequence of activities and interactions between several use cases in a system. Similar to a flowchart, this diagram illustrates the workflow from one activity to another. Creating an activity diagram in the early stages of process modeling can provide a better understanding of the overall process. The activity diagram in this study is as follows:

In the student practice process, after the student selects to start the practice, the system will retrieve questions from the database and randomly select 10 questions from the available question pool. After the selection is complete, the system will display the 10 selected questions to the student, who can then begin answering the questions. If the student submits answers, but they are incomplete, the system will redirect them back to the question page. Fig. 4 depicts the activity diagram of the study.

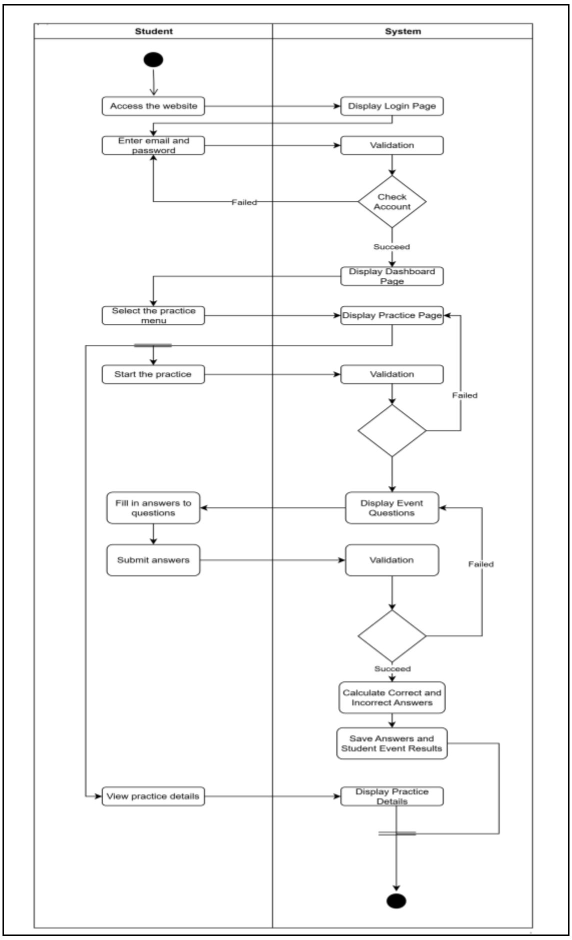


Fig. 4 Activity Diagram

5. Result and Analysis

5.1. LMS Bebras Feature System

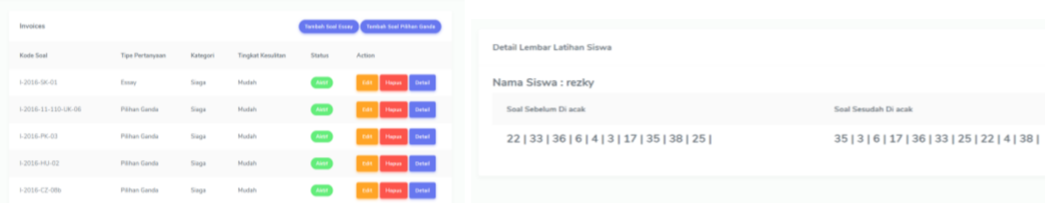
This section provides a clearer overview of the developed system, where the main features are implemented in the Bebras LMS. Each feature is designed to support the research objectives, namely the provision of a question management mechanism, event implementation, and interaction between users and the system. The details of each feature are shown in Table 2.

Table 2. LMS Bebras Feature System

No	Feature	Description
1	Login as Admin and a student	The login feature is used to verify the identity of users (both admin and students) before accessing the system. If the entered data is incorrect, the system displays a login failed message. If correct, the user is directed to the dashboard page.

2	Navigation	The navigation menu facilitates users in accessing various pages, such as the dashboard, questions, events, practice results, and student data. Access rights are restricted according to roles (admin, teacher, student).
3	Question	The admin is able to create, modify, and delete questions, either in multiple-choice or essay format. The added questions will be stored in the question list and can be used in both exercise sessions and events.
4	Event	The admin can create events containing a set of certain questions with restricting the time for conducting the test. Students may select the events to work on, while the admin may be able to monitor in detail the events, participants, and results.
5	Exercise	Students can do the exercise consisting of 10 questions with a duration of 30 minutes. The type of questions may become multiple-choice or essay, and upon completion, students can submit their answers and review the practice results.
6	Result/Scoring	The admin is able to view the results of practices and events completed by students, including scores and details of question completion. Students can also review the answers they have worked on.
7	Question Randomization	The system implements the Linear Congruent Method (LCM) algorithm to select questions and the Fisher-Yates Shuffle (FYS) to randomize their order, ensuring that each student receives a different sequence.
8	Logout	Users can log out of their active account through the logout menu. After that, the system will redirect them back to the login page.

Table 2 presents a snippet of the application feature displayed on the admin and user interfaces. Fig. 5 depicts the question input page and randomization question.



Kode Soal	Tipe Pertanyaan	Kategori	Tingkat Kesulitan	Status	Action
I-2016-SK-01	Essay	Siaga	Mudah	Aktif	Edit Hapus Detail
I-2016-11-110-UK-06	Pilihan Ganda	Siaga	Mudah	Aktif	Edit Hapus Detail
I-2016-PK-03	Pilihan Ganda	Siaga	Mudah	Aktif	Edit Hapus Detail
I-2016-HU-02	Pilihan Ganda	Siaga	Mudah	Aktif	Edit Hapus Detail
I-2016-CZ-08b	Pilihan Ganda	Siaga	Mudah	Aktif	Edit Hapus Detail

Detail Lembar Latihan Siswa

Nama Siswa : rezky

Soal Sebelum Di acak	Soal Sesudah Di acak
22 | 33 | 36 | 6 | 4 | 3 | 17 | 35 | 38 | 25 |	35 | 3 | 6 | 17 | 36 | 33 | 25 | 22 | 4 | 38 |

Fig. 5 Question Input Page and Randomization Question

5.2. Evaluation Application

To ensure the function of the Bebras Learning Management System (LMS) works effectively for users, a series of comprehensive evaluations has been carried out. These evaluations include functional testing using the Black Box Testing method, correlation analysis between the Linear Congruent Method algorithm and the Fisher-Yates Shuffle, as well as user satisfaction assessment through questionnaires. The purpose of these tests is to evaluate system stability, the accuracy of core features, ease of use, and the level of user satisfaction with the provided interface and navigation flow. The results of these three methods offer a comprehensive overview of the system's technical performance as well as the user experience during interaction.

In the Black Box Testing, various usage scenarios were tested to ensure the system operates as expected. This included testing the login feature, question management (creation, editing, deletion), event management, practice exercises, as well as practice and event result summaries. Each test showed that the tested features functioned properly according to requirements without significant errors. A correlation analysis was conducted to evaluate the relationship between the Linear Congruent Method and the Fisher-Yates Shuffle algorithms. Based on the data analysis carried out, interpretations were obtained regarding the results of question randomization for each student, which were then documented in Table 3.

Table 3. Interpretation of the Question Shuffling for Each Student

No	Method	Test Order										Person Correlation	Sig	Interpretation
1	LCM	16	9	4	12	5	11	13	7	15	18	0.44	0.203	MC
	FYS	7	12	4	9	13	18	11	5	15	16			
2	LCM	12	5	11	7	16	6	13	10	4	14	0.73	0.017	SC
	FYS	13	10	6	11	16	5	14	7	4	12			
3	LCM	12	26	15	28	32	23	16	3	8	11	0.151	0.677	NC
	FYS	16	28	11	23	3	32	8	26	15	12			
4	LCM	30	20	29	5	36	39	17	14	16	3	0.558	0.094	MC
	FYS	17	39	14	16	30	36	29	3	20	5			
5	LCM	22	33	36	6	4	3	17	35	38	25	0.623	0.054	SC
	FYS	35	3	6	17	36	33	25	22	4	38			
6	LCM	28	32	20	27	4	18	38	12	7	10	0.329	0.354	WC
	FYS	20	32	7	38	12	10	18	28	27	4			
7	LCM	18	20	36	24	4	17	35	29	30	37	0.044	0.905	NC
	FYS	36	4	30	18	35	17	37	29	20	24			
8	LCM	29	17	24	36	15	22	3	11	25	6	0.375	0.286	WC
	FYS	24	6	11	36	25	3	15	17	29	22			
9	LCM	29	17	24	36	15	22	3	11	25	6	0.118	0.745	NC
	FYS	6	29	25	22	15	3	11	24	36	17			
10	LCM	40	8	13	10	12	28	9	33	37	16	0.41	0.24	MC
	FYS	13	16	12	8	10	40	33	37	28	9			
	FYS	10	37	9	8	12	33	28	40	13	16			

Based on the analysis in Table 2, it can be concluded that the application of the Fisher-Yates Shuffle algorithm and the Linear Congruent Method to the web-based Bebras challenge question randomization demonstrated varying levels of performance continuity. Pearson correlation results show varying correlations between question sequences, ranging from strong and significant, moderate, weak, to uncorrelated. This indicates that the applied randomization mechanism is capable of producing a random distribution of questions, but the level of performance continuity is not always uniform across trials.

In general, the question randomization generated by this web-based system met its primary objective, namely, providing a variety of question sequences for each participant. However, differences in correlation levels between trials indicate that the algorithm's performance continuity is still dynamic, so the resulting randomization pattern is not completely stable across conditions. Thus, this randomization system can be considered effective in reducing question repetition, although there is still variation in its performance continuity. This is also supported by the questionnaire results, as shown in the table. Additionally, a user satisfaction assessment was conducted using a questionnaire containing five statements regarding ease of use, feature usefulness, and the app's support for student preparation for the Bebras challenge, as shown in Table 4.

Table 4. Questionnaire Result

No	**Question**	**Choose**				
		STS	**TS**	**N**	**S**	**SS**
1	Is this web-based application easy to use for challenge participants?	0	0	0	12	8
2	Exercise features in this application help students improve their workmanship skills?	0	0	0	14	6
3	This application can help students understand the challenges of Bebras.	0	0	0	14	6
4	This application can help students prepare themselves to face the challenges.	0	0	1	12	7
5	With this application, can the quality of learning of the Father or Mother of the teacher be improved?	0	0	1	12	7

Based on the percentage value of the questionnaire, it can be concluded that this test obtained a final value of 86%, which falls under the "Strongly Agree" category. This result indicates that the LMS developed can assist the Bebras UBG bureau in conducting the Bebras challenge, and the Randomization System in this application helps students create randomized practice questions.

6. Conclusion

This study demonstrates that applying the Linear Congruential Method (LCM) and Fisher–Yates Shuffle algorithms for question selection and randomization effectively. The proposed approach can generate diverse correlation levels, including No Correlation (38%), Weak Correlation (29.5%), Moderate Correlation (19%), Strong Correlation (12%), and Perfect Correlation (1.5%). The variation in correlation outcomes depends on the dataset characteristics and distribution. The findings confirm that both algorithms can produce sufficiently random question orders under appropriate parameter configurations. Furthermore, the analysis shows that higher correlation levels correspond to lower randomization quality, whereas lower correlation levels indicate better randomness and unpredictability. The evaluation conducted at the Bebras UBG bureau achieved an accuracy rate of 86%, validating the practical applicability of the proposed approach. For future work, we plan to enhance the randomness evaluation using larger and more heterogeneous datasets, integrate hybrid stochastic algorithms to improve distribution uniformity, and analyze the computational efficiency of both methods in real-time assessment systems.

References

[1] Apriani, I. Ismarmiaty, D. Susilowati, K. Kartarina, and W. Suktiningsih, “Application of Computational Thinking in Mathematics Lessons at Madratsah Ibtidaiyah Nurul Islam Sekarbela Mataram,” *ADMA J. Community Service and Empowerment*, vol. 1, no. 2, pp. 47–56, 2021, doi: 10.30812/adma.v1i2.1017.

[2] D. Assaf et al., “The CTSkills App -- Measuring Problem Decomposition Skills of Students in Computational Thinking,” vol. 1, no. 1, 2024, [Online]. Available: http://arxiv.org/abs/2411.14945.

[3] D. Tresnawati et al., “Developing Students’ Computational Thinking in Garut through Bebras Challenge,” *J. PkM MIFTEK*, vol. 1, no. 1, pp. 55–60, 2020, doi: 10.33364/miftek/v.1-1.55.

[4] R. R. C. Putra and T. Sugihartono, “Application of Fisher-Yates Shuffle Algorithm in Computer Based Test at SMKN 1 Payung,” *MATRIK J. Management, Information Technology and Computer Engineering*, vol. 18, no. 2, pp. 276–283, 2019, doi: 10.30812/matrik.v18i2.399.

[5] R. Zaid, S. Sutardi, and B. Pramono, “Implementation of Linear Congruential Generator (LCG) Method in Android-Based IQ Test,” *semanTIK*, vol. 7, no. 1, p. 27, 2021, doi: 10.55679/semantik.v7i1.12925.

[6] V. Asih, A. Saputra, and R. T. Subagio, “Application of Fisher Yates Shuffle Algorithm for Android-Based Examination Application,” *J. Digit*, vol. 10, no. 1, p. 59, 2020, doi: 10.51920/jd.v10i1.156.

[7] W. A. Rohmah, A. Asriyanik, and W. Apriyandari, “Implementation of Fisher Yates Shuffle Algorithm in Quiz Game Environment,” *J. Informatics Telecommun. Eng.*, vol. 4, no. 1, pp. 161–172, 2020, doi: 10.31289/jite.v4i1.3863.

[8] D. Desliyanti, A. Fahry, and U. M. Bengkulu, “English Learning Application Using Linear Congruential Generator Algorithm Based on Android,” vol. 5, pp. 9–15, 2023.

[9] N. Aini and E. Y. Wijaya, “Implementation of Fisher-Yates Algorithm for Randomization of Goalpro Education Game Questions,” *J. Ilm. Edutic Education and Informatics*, vol. 8, no. 2, pp. 147–156, 2022, doi: 10.21107/edutic.v8i2.14418.

[10] D. Hindarto, “Indonesian Culinary Application System Design with UML Method,” *J. Comput. Networks, Architecture and High Performance Computing*, vol. 5, no. 2, pp. 612–622, 2023.

[11] S. Dharwiyanti and R. S. Wahono, “Introduction to Unified Modeling Language (UML),” *IlmuKomputer.com*, pp. 1–13, 2003, [Online]. Available: http://www.unej.ac.id/pdf/yanti-uml.pdf.

[12] D. Lestari, “The Use of Computer Based Test as an Evaluation Tool and Its Impact on Assessment Effectiveness in History Subject at SMA Negeri 1 Boyolali Academic Year 2015/2016,” *Jurnal CANDI*, vol. 19, no. 1, pp. 29–39, 2019.

[13] D. Purba and M. Purba, “Application of Correlation and Regression Analysis using Pearson Product Moment and Simple Linear Regression,” *Citra Sains Teknol.*, vol. 1, no. 2, pp. 97–103, 2022.

[14] M. K. Alim and D. B. Arianto, “Correlation Analysis Between Economic Factors and Population Distribution in East Java in 2022 Using Pearson Correlation Method,” *Kohesi J. Multidisciplinary Saintek*, vol. 1, no. 4, pp. 20–30, 2023.

[15] D. Aipina and H. Witriyono, “Utilization of Laravel Framework and Bootstrap Framework in Developing Web-Based Hijab Sales Application,” *J. Media Infotama*, vol. 18, no. 1, 2022.

[16] Hangga and E. Prabowo, “Modification of Linear Congruential Generator for Randomization System in Computer Based Test (CBT),” *J. Electrical Eng. Unnes*, vol. 8, no. 2, pp. 47–49, 2016.

[17] R. Supardi and T. D. Putra, “Design of Android-Based Suitcake Game Using Linear Congruent Algorithm,” *J. Inf. Technol.*, vol. 4, no. 1, pp. 28–34, 2020, doi: 10.36294/jurti.v4i1.1152.

[18] D. S. Utama and Y. Asriningtias, “Comparison of Access Time of Fisher-Yates Shuffle Algorithm and Linear Congruent Method in Web-Based Try-Out Questions,” *JISKA (J. Informatics Sunan Kalijaga)*, vol. 2, no. 2, pp. 93–102, 2017, doi: 10.14421/jiska.2017.22-04.

[19] N. Aini and E. Y. Wijaya, “Implementation of Fisher-Yates Algorithm for Randomization of Goalpro Education Game Questions,” *J. Ilm. Edutic*, vol. 8, no. 2, pp. 147–156, 2022.

[20] P. Sulistyorini, “Visual Modeling Using UML and Rational Rose,” *J. Inf. Technol. Din.*, vol. XIV, no. 1, pp. 23–29, 2009.

[21] D. Purnomo, “Prototyping Model in Information System Development,” *JIMP - J. Inform. Merdeka Pasuruan*, vol. 2, no. 2, pp. 54–61, 2017, doi: 10.37438/jimp.v2i2.67.

[22] D. W. T. Putra and R. Andriani, “Unified Modelling Language (UML) in the Design of Restitution Payment Application System,” *J. Teknolf*, vol. 7, no. 1, p. 32, 2019, doi: 10.21063/jtif.2019.v7.1.32-39.

[23] G. G. Maulana, “Basic Learning of Algorithm and Programming Using Web-Based El-Goritma,” *J. Mech. Eng.*, vol. 6, no. 2, p. 8, 2017, doi: 10.22441/jtm.v6i2.1183.