Indonesian Journal of Electrical Engineering and Computer Science
Vol. 32, No. 3, December 2023, pp. 1521~1532
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v32.i3.pp1521-1532

Analysis of linear congruent methods and multiplicative random number generator in computer-based test

Amrullah, Al-Khowarizmi, Firahmi Rizky
Department of Information Technology, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia

Article Info

Article history:

Received Jul 6, 2023
Revised Aug 2, 2023
Accepted Sep 2, 2023

Keywords:

Computer based test
Linear congruent method
Multiplicative random number generator
Question package
Randomization

ABSTRACT

This research focuses on the implementation of computer-based test exams in high schools which face the problem of not having differences in exam questions which results in weak security and validity of exam results. Therefore, a randomization method is needed to overcome this problem. The method used in randomization is linear congruent methods and multiplicative random number generator. There are 101 random questions, but only 40 questions are displayed for each student with a reference value of Xn and C, 2 will be added for each package of exam questions and to avoid question code=0, the calculation results will be added 1. The linear congruent methods (LCM) results achieve 100% accuracy, while the method The multiplicative random number generator (MRNG) only achieved 62.5% accuracy in randomizing the exam questions. This accuracy comparison highlights the difference in the ability of the two methods to generate random permutations of test item packages. LCM randomization accuracy ensures that each student will receive a different set of test questions in a consistent manner. However, the low accuracy of randomization using MRNG indicates a weakness in generating permutations of exam question packages. The results of this study show that the LCM method is better than the MRNG in conducting exams.

Corresponding Author:

Amrullah
Department of Information Technology, Universitas Muhammadiyah Sumatera Utara
Medan, Indonesia
Email: amrullah@umsu.ac.id

1. INTRODUCTION

In the 5.0 revolution era as it is today, the advancement of information and communication technology has brought about significant changes across various spheres of life, one of which is the world of education. The impact of this development has positive and negative sides depending on who the users will be using the technology. Mastery of technology properly and correctly requires adequate knowledge so that we can utilize it in facing the demands of a globalized world that has a lot of competition. One of the developments in information technology in education is the web-based semester exam assessment, known as computer based test (CBT) [1].

In this study, CBT is a computer based test used in vocational high schools (VHS) to evaluate students' abilities to master a subject digitally. CBT is usually carried out in a computer laboratory or classroom equipped with a computer and internet network. Students can take exams online and get their results quickly upon completion. The application of the CBT exam in VHS is becoming increasingly common as an alternative to the traditional paper-based exam. However, the implementation of CBT exams in vocational high schools still faces several problems that need to be addressed, such as the absence of differences in exam questions which results in weak security and validity of exam results.

The absence of differences in exam questions has the potential to reduce the security of exam implementation. If all students get the same questions, there is a risk of leaking the questions before the exam. In addition, if the same questions are used repeatedly, students may share answers or seek help from external sources, threatening the validity of exam results [2]–[5].

To overcome this problem, a randomization method is needed in making exam questions. In this study, the randomization method used was the linear congruential method (LCM) and the multiplicative random number generator (MRNG). Both of these randomization models have long been used in the development of randomization systems. A random number is a value derived from a process whose outcome is unpredictable and subsequent generation of the same number is impossible. The process of generating random numbers using a computer is called a pseudo random number generator (PRNG) [6]–[8]. Applying these two randomization methods can help randomize exam questions so that each student gets a different package of questions at random.

LCM is a random number generator that is widely used in computer programs. This method is used to randomize the existing questions so that the questions are arranged randomly and are ready to be used as question packages. MRNG, it is an algorithm employed to generate random numbers by utilizing mathematical formulas that are iterated as required [9]–[12].

Furthermore, in research after the two methods have been applied in the CBT exam question package, the LCM and MRNG methods will also be compared. LCM is known to have an advantage in generating random and evenly distributed permutations of questions, while MRNG has different characteristics in terms of permutations of questions. A comparison between the two will provide clearer insight into the effectiveness and suitability of each method in the context of randomizing CBT exam questions in VHSs.

Through this research, it is also hoped that we can better understand the comparative context between the LCM and MRNG methods in randomizing CBT exam questions at VHSs. This information will be useful for choosing the most suitable method in ensuring the safety, legitimacy and fairness of administering the CBT exam in VHSs.

2. RESEARCH METHOD

Here, the research steps, analytical and experimental methods used to obtain comparative results from each pseudo-random number generation algorithm are explained. For details, see Figure 1. Based on the aforementioned Figure 1, the general research steps can be elucidated as:

- Study of literature: during the library research phase, this study used a variety of library sources, including books, journals (print and online), articles, and other relevant references. This article is intended to help researchers solve problems related to LCM analysis and MRNG in the context of CBT.
- Data collection: at this stage, it is the stage of collecting all the data needed, such as the question data that will be used for CBT.
- Pseudo random number generator: at this stage, PRNG are generated by comparative analysis of the LCM and MRNG algorithms.
- Problem analysis: at this stage, the problem analysis seeks to introduce randomness into the CBT and assess the accuracy of PRNG by both the LCM and MRNG algorithms.
- Testing: at this stage, the conducted tests aimed at establishing the validity of accurate outcomes indicate that the numbers generated from both the LCM and MRNG algorithms have been effectively randomized within the context of CBT.
- Results and conclusions: at this stage, the results and conclusions regarding the validity use of LCM and MRNG algorithms for randomization and assessment of accuracy applicable to CBT are explained.

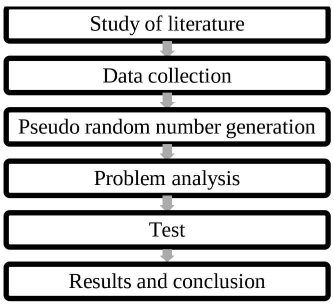


Figure 1. Research stages

2.1. Random numbers

Random numbers in general can be used for many purposes. For example, these methods find applications in a variety of fields, such as games such as dice, statistical analysis, computer simulations, and games of chance, among others. Then along with the development of computers, widely used random number, especially in cryptography, for example to generate key parameters in public key algorithms, and generate vectors initialization (IV) in the symmetric key algorithm. However, the randomness one experiences on a computer is just pseudo-random and not truly random because one day the cycle of the generated number will reappear. This is because the sequence of random numbers is generated from a mathematical formula, or in other words, the resulting sequence is derived from generating repeatable numbers [13]–[16]. The following are random and pseudorandom generator diagrams which will be explained in Figure 2.

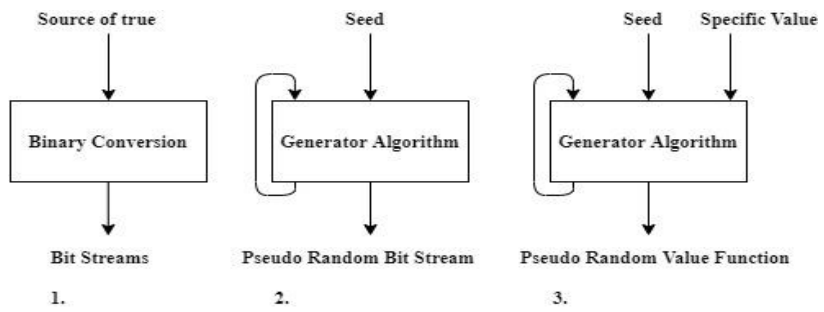


Figure 2. Random and pseudo random generator scheme

2.2. Algorithm of linear congruence method

Algorithm is a sequence of instructions to solve a problem. Below is the algorithm that will be built using the LCM method, which can be seen as [17]–[21]:
- Initialize the problem id with the values Xn, a, Xn-1, c and m.
- Enter question id data and values in Xn, a, Xn-1, c and m.
- The process of calculating randomization of questions using LCM method.
- Xn+1=(aXn-1+c) mod m.
- The process of CBT exam questions is random.
- View CBT exam question pack.

2.3. Algorithm multiplicative random number generator

A computer-generated (pseudo-random) random number algorithm, these random numbers are generated using a recurring mathematical formula as required. This is the algorithm that will be built using the MRNG method, the algorithm is as [22]–[26]:
- Initialize the problem id with the values Xn, a, Xn-1, c and m.
- Enter question id data and values in Xn, a, Xn-1, c and m.
- The process of calculating randomization of questions by MRNG method.
- Xn+1=(aXn-1+c) mod m.
- R1=Xn+1/m.
- Random CBT exam question process.
- Appears in the ranking with the lowest R value.

2.4. System analysis

The system analysis in this study was performed by implementing LCM and MRNG with the aim of randomizing CBT. The fundamental concept in random number generation using LCM and MRNG is the utilization of Xi-1 as the seed value. Both LCM and MRNG exhibit a period not exceeding the modulus value, denoted as 'm,' and in this particular case, the period is smaller than m. A complete cycle in LCM and MRNG occurs when the subsequent conditions are satisfied.
- Make sure the constant C is greater than the square root of m.
- Avoid using the constant C as a multiple of m.
- Choose the prime number corresponding to the value of m.
- Set the initial value of Xn to be greater than 0 and less than m.
- Use an odd value for the constant a.

The analysis of the system, using various algorithms at different stages, is shown in Figure 3. The objective of the randomization test is to utilize an algorithm for generating new question packages that will be presented in the CBT. In this study, an in-depth analysis was carried out to obtain new question packages because basically the randomization method only applies to one round in terms of randomization. However, in this study the randomization of the question packages was not only for one round but more than one round according to the number of students taking the CBT exam. So below, we will describe in detail the additional novelty algorithm that will be applied by the researcher so that the question packages can be randomized in more than one round which will be displayed using LCM and MRNG:

- The value of C is set to 23, taken from the year the CBT was carried out. For the next set of questions, the value of C is increased by 2 compared to the value of the previous C. The sum of the 2 values assigned to the next set of questions increases creating a new random number in this set.
- The constant C must not be a multiple of the value M.
- The value of m is 101, chosen based on the total number of questions that the CBT will randomly select. It is chosen to be a prime number and not a multiple of the value C.
- The value of Xn=40, determined from the number of questions to be displayed, is 40, and the value of Xn in the next question pack is the same as the value of C that will be added 2 from the previous packet. The sum of the 2 values provided for the next question pack in Xn will form a new random number in the next question pack.
- The constant 'a' must be an odd number, therefore, a=1, Selected due to its initial odd value within the decimal range.

From the above procedure, it is known that the reference values for randomization are Xn=40, C=23, a=1 and m=101. To avoid question code=0, the calculation result will be added to the number 1 and for the next question pack as described above to find more than one random, the addition algorithm will be applied in the LCM and MRNG, i.e. the reference values Xn and C will be added 2 from known results will serve as a reference for the randomization and novelty understanding applied to LCM and MRNG algorithms.

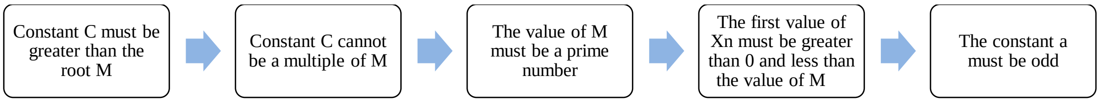


Figure 3. Flowchart of LCM and MRNG

3. RESULTS AND DISCUSSION

3.1. LCM randomization results

Knowing the value Xn=40, value C=23, value a=1 and value m=101. And to avoid question code=0, the calculation result will be added 1, Figure 4 LCM random results using Google colab. Figure 4 illustrates the results of calculating the first question package. In this discussion, a random analysis was performed on 10 question packets, each consisting of 101 question, with 40 items presented in each individual packet. See Figure 5 for the outcomes out of 10 question packs. Figure 5 shows the results of question packages 1 to 10, using the LCM method and using the Google Colab application utilizing the Python programming language.

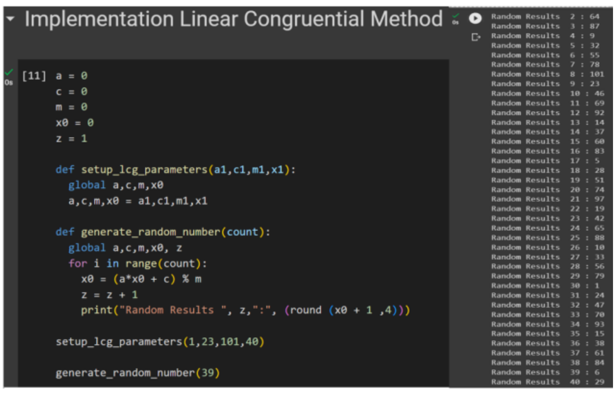
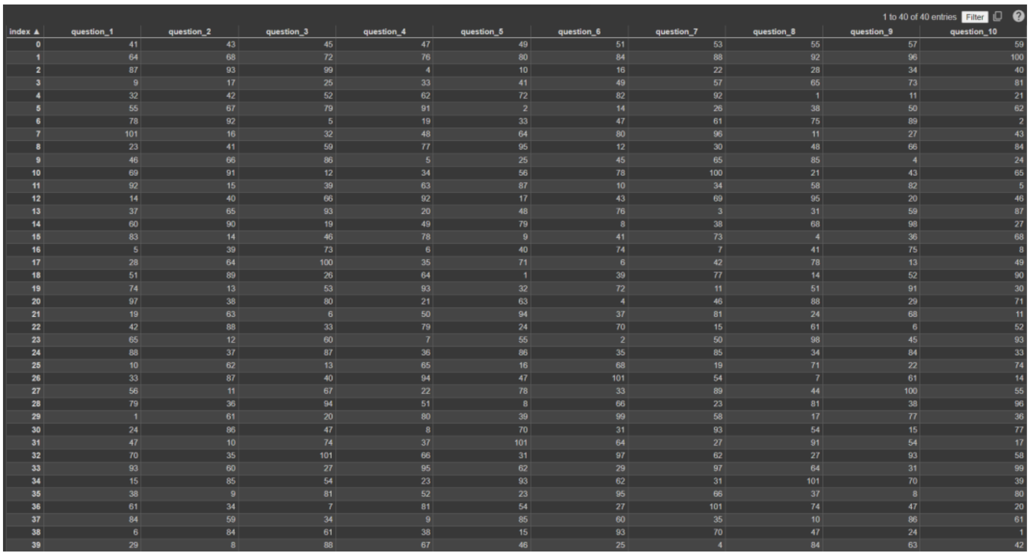


Figure 4. Results of LCM method randomization

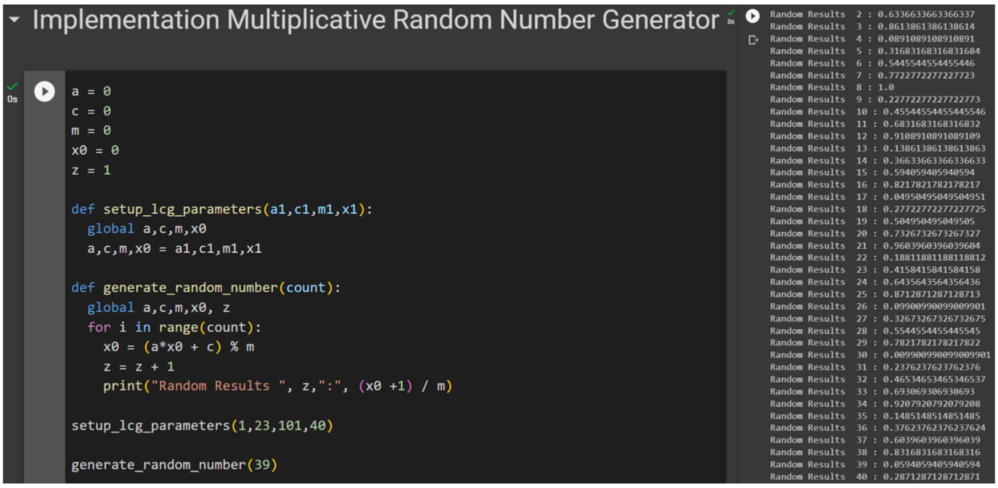


index	question_1	question_2	question_3	question_4	question_5	question_6	question_7	question_8	question_9	question_10
0	41	43	45	47	49	51	53	55	57	59
1	64	68	72	76	80	84	88	92	96	100
2	87	93	99	4	10	16	22	28	34	40
3	9	17	25	33	41	49	57	65	73	81
4	32	42	52	62	72	82	92	1	11	21
5	55	67	79	91	2	14	26	38	50	62
6	78	92	5	19	33	47	61	75	89	2
7	101	16	32	48	64	80	96	11	27	43
8	23	41	59	77	95	12	30	48	66	84
9	46	66	86	5	25	45	65	85	4	24
10	69	91	12	34	56	78	100	21	43	65
11	92	15	39	63	87	10	34	58	82	5
12	14	40	66	92	17	43	69	95	20	46
13	37	65	93	20	48	76	3	31	59	87
14	60	90	19	49	79	8	38	68	98	27
15	83	14	46	78	9	41	73	4	36	68
16	5	39	73	6	40	74	7	41	75	8
17	28	64	100	35	71	6	42	78	13	49
18	51	89	26	64	1	39	77	14	52	90
19	74	13	53	93	32	72	11	51	91	30
20	97	38	80	21	63	4	46	88	29	71
21	19	63	6	50	94	37	81	24	68	11
22	42	88	33	79	24	70	15	61	6	52
23	65	12	60	7	55	2	50	98	45	93
24	88	37	87	36	86	35	85	34	84	33
25	10	62	13	65	16	68	19	71	22	74
26	33	87	40	94	47	101	54	7	61	14
27	56	11	67	22	78	33	89	44	100	55
28	79	36	94	51	8	66	23	81	38	96
29	1	61	20	80	39	99	58	17	77	36
30	24	86	47	8	70	31	93	54	15	77
31	47	10	74	37	101	64	27	91	54	17
32	70	35	101	66	31	97	62	27	93	58
33	93	60	27	95	62	29	97	64	31	99
34	15	85	54	23	93	62	31	101	70	39
35	38	9	81	52	23	95	66	37	8	80
36	61	34	7	81	54	27	101	74	47	20
37	84	59	34	9	85	60	35	10	86	61
38	6	84	61	38	15	93	70	47	24	1
39	29	8	88	67	46	25	4	84	63	42

Figure 5. Results of questionnaires using LCM 1-10

3.2. MRNG randomization results

During the process of randomizing using the MRNG method, there is no distinction in the reference for establishing randomness when compared to the LCM method. As mentioned, the values Xn=40, C=23, a=1, and m=101 are employed. To prevent a question code of 0, the computed result will be incremented by 1. Refer to Figure 6 for the outcomes of randomization using the MRNG technique in Google Colab:




```
a = 0
c = 0
m = 0
x0 = 0
z = 1

def setup_lcg_parameters(a1,c1,m1,x1):
  global a,c,m,x0
  a,c,m,x0 = a1,c1,m1,x1

def generate_random_number(count):
  global a,c,m,x0, z
  for i in range(count):
    x0 = (a*x0 + c) % m
    z = z + 1
    print("Random Results ", z,":", (x0 +1) / m)

setup_lcg_parameters(1,23,101,40)

generate_random_number(39)
```

```
Random Results  2 : 0.6336633663366337
Random Results  3 : 0.8613861386138614
Random Results  4 : 0.0891089108910891
Random Results  5 : 0.31683168316831684
Random Results  6 : 0.5445544554455446
Random Results  7 : 0.7722772277227723
Random Results  8 : 1.0
Random Results  9 : 0.22772277227722773
Random Results  10 : 0.45544554455445546
Random Results  11 : 0.6831683168316832
Random Results  12 : 0.9108910891089109
Random Results  13 : 0.13861386138613863
Random Results  14 : 0.36633663366336633
Random Results  15 : 0.594059405940594
Random Results  16 : 0.8217821782178217
Random Results  17 : 0.04950495049504951
Random Results  18 : 0.27722772277227725
Random Results  19 : 0.504950495049505
Random Results  20 : 0.7326732673267327
Random Results  21 : 0.9603960396039604
Random Results  22 : 0.18811881188118812
Random Results  23 : 0.4158415841584158
Random Results  24 : 0.6435643564356436
Random Results  25 : 0.8712871287128713
Random Results  26 : 0.09900990099009901
Random Results  27 : 0.32673267326732675
Random Results  28 : 0.5544554455445545
Random Results  29 : 0.7821782178217822
Random Results  30 : 0.009900990099009901
Random Results  31 : 0.2376237623762376
Random Results  32 : 0.46534653465346537
Random Results  33 : 0.693069306930693
Random Results  34 : 0.9207920792079208
Random Results  35 : 0.1485148514851485
Random Results  36 : 0.37623762376237624
Random Results  37 : 0.6039603960396039
Random Results  38 : 0.8316831683168316
Random Results  39 : 0.0594059405940594
Random Results  40 : 0.2871287128712871
```

Figure 6. Results of randomization of the MRNG method

Figure 6 illustrates the results of calculating the first question package. In this discussion, a random analysis of 10 question packages is carried out comprising 101 questions with 40 questions presented within each package, the configuration is detailed. Refer to Figure 7 for the depiction of the outcomes out of 10 question packs. Figure 7 displays the outcomes of question packages 1 to 10, implemented using the MRNG method, utilizing the Google Colab implementation using the Python programming language.

3.3. Analyzing LCM and MRNG randomization

During this phase, a complete and unambiguous analysis will be given, in-depth examination of identical questions on each question pack and between question packs. Below, the differences are described and the randomization results obtained by applying the LCM and MRNG methods. Table 1 presents the randomization results obtained by the LCM method.

Then Table 2 the results of randomization using LCM as many as 1-10 question packages will be displayed and it can be seen whether the randomization has similarities in the question packages and between question packages. Furthermore, below can be seen the results of randomization in Table 3 in *appendix* using the MRNG method. Then Table 4 in *appendix* the results of randomization using MRNG as many as 1-10 question packages will be displayed and it can be seen whether the randomization has similarities in the question packages and between question packages.

After applying the results of the analysis, it appears that the use of the LCM and MRNG methods in randomizing the CBT exam question packages was completely successful. The results of the analysis of this study show that the reference values for the randomization process are Xn=40, C=23, a=1, and m=101. In addition, to avoid the question code having a value of 0, the calculation results will be added 1. For the next package of questions, the reference values of Xn and C will be added by 2. The LCM method proved to be very effective, because there were no similarities between the CBT exam question packages either in application or between the question packages. Meanwhile, the use of the MRNG method in randomizing the CBT exam question packages, although it has similarities with LCM in that there are no similarities in the CBT exam question packages, in the context of application between CBT exam question packages, there are similarities. Further information can be found in Table 5 which illustrates the similarities in the CBT exam question packages. From Table 5 in *appendix* it can be seen that the same question packages from packages 1-10 where as many as 31 questions from 15 positions in the question package got the same question position results which were weaknesses and shortcomings in the implementation of randomization using MRNG.

1 to 40 of 40 entries Filter

index	question_1	question_2	question_3	question_4	question_5	question_6	question_7	question_8	question_9	question_10
0	25	21	25	24	21	20	22	19	19	18
1	15	11	14	12	8	6	8	4	3	1
2	6	1	3	41	37	34	35	30	28	26
3	38	32	33	30	25	21	20	15	12	9
4	29	22	22	18	12	7	6	41	37	33
5	19	12	11	6	40	35	33	26	22	16
6	10	2	41	35	28	22	18	11	6	40
7	1	33	30	23	15	8	4	37	31	24
8	33	23	19	11	2	36	31	22	15	7
9	23	13	8	40	31	23	16	7	40	32
10	13	3	38	29	18	9	2	33	25	15
11	4	34	27	17	5	37	29	18	9	39
12	36	24	16	5	34	24	14	3	34	23
13	27	14	5	34	22	10	41	29	18	6
14	17	4	35	22	9	38	27	14	2	31
15	8	35	24	10	38	25	12	40	27	14
16	40	25	13	39	26	11	39	25	11	38
17	31	15	2	28	13	39	25	10	36	22
18	21	5	32	16	41	26	11	36	21	5
19	11	36	21	4	29	12	38	21	5	30
20	2	26	10	33	16	40	24	6	30	13
21	34	16	40	21	3	27	10	32	14	37
22	24	6	29	9	32	13	37	17	39	21
23	14	37	18	38	19	41	23	2	24	4
24	5	27	7	27	6	28	9	28	8	29
25	37	17	37	15	35	14	36	13	33	12
26	28	7	26	3	23	1	21	39	17	36
27	18	38	15	32	10	29	7	24	1	20
28	9	28	4	20	39	15	34	9	26	3
29	41	18	34	8	27	2	19	35	10	28
30	32	8	23	37	14	30	5	20	35	11
31	22	39	12	26	1	16	32	5	20	35
32	12	29	1	14	30	3	17	31	4	19
33	3	19	31	2	17	31	3	16	29	2
34	35	9	20	31	4	17	30	1	13	27
35	26	40	9	19	33	4	15	27	38	10
36	16	30	39	7	20	32	1	12	23	34
37	7	20	28	36	7	18	28	38	7	17
38	39	10	17	25	36	5	13	23	32	41
39	30	41	6	13	24	33	40	8	16	25

Figure 7. Results of question packages using MRNG 1-10

3.5. Testing result

From the results that have been implemented regarding the randomization of CBT exam questions using LCM and MRNG, conclusions can be drawn about which randomization method is better applied to the CBT exam application. Based on the findings from the earlier comparative analysis, it is evident that LCM stands out as a notably superior method for randomizing CBT exam questions. This superiority is evident in its ability to effectively randomize both individual CBT exam question packages and the questions spanning across different CBT exam packages. The implementation of LCM ensures a complete absence of similarity in the randomization process.

Whereas in the implementation of the MRNG there are weaknesses and deficiencies that occur, namely the same package of questions from packages 1-10 which are randomly randomized as many as 31 questions from 15 positions in the question package get the same position results, even though in terms of the question packages the same as LCM classified as having no similarities in this research. The following test results in percentage accuracy result format will then be obtained from the LCM and MRNG methods applied in this study. The results of the accuracy of the level of complexity of randomized questions using LCM.

$$The\ difficulty\ of\ the\ question = \frac{number\ of\ questions - same\ number\ of\ questions}{number\ of\ models} \times 100\%$$

$$The\ difficulty\ of\ the\ question = \frac{40-0}{40} \times 100\%$$

$$The\ difficulty\ of\ the\ question = 100\ \%$$

The results of the accuracy of the level of complexity of randomized questions using MRNG.

$$The\ difficulty\ of\ the\ question = \frac{number\ of\ questions - same\ number\ of\ questions}{number\ of\ models} \times 100$$

$$The\ difficulty\ of\ the\ question = \frac{40-15}{40} \times 100\%$$

$$The\ difficulty\ of\ the\ question = 62.5\ \%$$

From the results of calculating the percentage level of complexity, LCM is obtained with a percentage of 100%, whereas because the MRNG between question packages has the same position between one question package and another question package as many as 15 positions in the question package, the MRNG percentage obtained is 62%.

Table 1. Randomization of question packages using LCM

No	Xn+1=((a(Xn+1)+c)) mod m	(Xn +1)	Rangking
1	40	40+1=41	41
2	((1×41)+23) mod 101=63	63+1=64	64
3	((1×64)+23) mod 101=86	86+1=87	87
4	((1×87)+23) mod 101=8	8+1=9	9
5	((1×9)+23) mod 101=31	31+1=32	32
6	((1×32)+23) mod 101=54	54+1=55	55
7	((1×55)+23) mod 101=77	77+1=78	78
8	((1×78)+23) mod 101=100	100+1=101	101
9	((1×101)+23) mod 101=22	22+1=23	23
10	((1×23)+23) mod 101=45	45+1=46	46
11	((1×46)+23) mod 101=68	68+1=69	69
12	((1×69)+23) mod 101=91	91+1=92	92
13	((1×92)+23) mod 101=13	13+1=14	14
14	((1×14)+23) mod 101=36	36+1=37	37
15	((1×37)+23) mod 101=59	59+1=60	60
16	((1×60)+23) mod 101=82	82+1=83	83
17	((1×83)+23) mod 101=4	4+1=5	5
18	((1×5)+23) mod 101=27	27+1=28	28
19	((1×28)+23) mod 101=50	50+1=51	51
20	((1×28)+23) mod 101=73	73+1=74	74
21	((1×74)+23) mod 101=96	95+1=97	97
22	((1×97)+23) mod 101=18	18+1=19	19
23	((1×19)+23) mod 101=41	41+1=42	42
24	((1×42)+23) mod 101=64	64+1=65	65
25	((1×65)+23) mod 101=87	87+1=88	88
26	((1×88)+23) mod 101=9	9+1=10	10
27	((1×10)+23) mod 101=32	32+1=33	33
28	((1×33)+23) mod 101=56	56+1=57	56
29	((1×57)+23) mod 101=78	238+1=79	79
30	((1×79)+23) mod 101=0	0+1=1	1
31	((1×1)+23) mod 101=23	23+1=24	24
32	((1×24)+23) mod 101=46	46+1=47	47
33	((1×302)+23) mod 101=69	69+1=70	70
34	((1×323)+23) mod 101=92	92+1=93	93
35	((1×93)+23) mod 101=14	14+1=15	15
36	((1×15)+23) mod 101=37	37+1=38	38
37	((1×38)+23) mod 101=60	60+1=61	61
38	((1×61)+23) mod 101=83	83+1=84	84
39	((1×84)+23) mod 101=5	5+1=6	6
40	((1×6)+23) mod 101=28	28+1=29	29

Table 2. Randomization results of problem packages 1-10 using LCM

	Question Package 1-10									
No	1	2	3	4	5	6	7	8	9	10
1	41	43	45	47	49	51	53	55	57	59
2	64	68	72	76	80	84	88	92	96	100
3	87	93	99	4	10	16	22	28	34	40
4	9	17	25	33	41	49	57	65	73	81
5	32	42	52	62	72	82	92	1	11	21
6	55	67	79	91	2	14	26	38	50	62
7	78	92	5	19	33	47	61	75	89	2
8	101	16	32	48	64	80	96	11	27	43
9	23	41	59	77	95	12	30	48	66	84
10	46	66	86	5	25	45	65	85	4	24
11	69	91	12	34	56	78	100	21	43	65
12	92	15	39	63	87	10	34	58	82	5
13	14	40	66	92	17	43	69	95	20	46
14	37	65	93	20	48	76	3	31	59	87
15	60	90	19	49	79	8	38	68	98	27
16	83	14	46	78	9	41	73	4	36	68
17	5	39	73	6	40	74	7	41	75	8
18	28	64	100	35	71	6	42	78	13	49
19	51	89	26	64	1	39	77	14	52	90
20	74	13	53	93	32	72	11	51	91	30
21	97	38	80	21	63	4	46	88	29	71
22	19	63	6	50	94	37	81	24	68	11
23	42	88	33	79	24	70	15	61	6	52
24	65	12	60	7	55	2	50	98	45	93
25	88	37	87	36	86	35	85	34	84	33
26	10	62	13	65	16	68	19	71	22	74
27	33	87	40	94	47	101	54	7	61	14
28	56	11	67	22	78	33	89	44	100	55
29	79	36	94	51	8	66	23	81	38	96
30	1	61	20	80	39	99	58	17	77	36
31	24	86	47	8	70	31	93	54	15	77
32	47	10	74	37	101	64	27	91	54	17
33	70	35	101	66	31	97	62	27	93	58
34	93	60	27	95	62	29	97	64	31	99
35	15	85	54	23	93	62	31	101	70	39
36	38	9	81	52	23	95	66	37	8	80
37	61	34	7	81	54	27	101	74	47	20
38	84	59	34	9	85	60	35	10	86	61
39	6	84	61	38	15	93	70	47	24	1
40	29	8	88	67	46	25	4	84	63	42

4. CONCLUSION

The study findings reveal that relying solely on the LCM and MRNG methods for randomizing the CBT exam questions at VHS does not lead to the desired outcome of randomizing all 10 question packages effectively. Therefore, it becomes crucial to incorporate an additional algorithm, meticulously considered by the researchers, to enhance the success of randomization within the LCM and MRNG methods.This entails augmenting the reference values Xn and C by 2 for each CBT exam question package. Furthermore, to prevent encountering a question code of 0, the calculation result will be incremented by 1. Moreover, within the implemented CBT exam package using LCM and MRNG, it is apparent that LCM outperforms MRNG in terms of randomization. LCM exhibits distinct superiority as it introduces no similarities within or between CBT exam question packages. On the other hand, MRNG, while achieving similarity-free randomization within individual CBT exam question packages akin to LCM, exhibits shortcomings in terms of the randomization between packages. Notably, between packages 1-10, 31 questions from 15 positions in the question packages yield identical question positions. In terms of accuracy, represented as a percentage, the test results indicate that LCM achieves a value of 100%, while MRNG attains a value of 62.5%.

ACKNOWLEDGEMENTS

The author would like to thank the Research Institute and Publication Institute of the Muhammadiyah University of North Sumatra and the University of North Sumatra, Medan Indonesia for supporting the dissemination of novice lecturer research (PDP) with number: 12/II.3-AU/UMSU-LP2M/C/2023.

APPENDIX

Table 3. Randomization of question packages using MRNG

No	Xn+1=((a(Xn+1)+c)) mod m	(Xn +1)	Ri=Xn+1 / m	Rangking
1	40	40+1=41	41/101=0.4059	25
2	((1×41)+23) mod 101=63	63+1=64	64/101=0.6337	15
3	((1×64)+23) mod 101=86	86+1=87	87/101=0.8614	6
4	((1×87)+23) mod 101=8	8+1=9	9/101=0.0891	38
5	((1×9)+23) mod 101=31	31+1=32	32/101=0.3168	29
6	((1×32)+23) mod 101=54	54+1=55	55/101=0.5446	19
7	((1×55)+23) mod 101=77	77+1=78	78/101=0.7723	10
8	((1×78)+23) mod 101=100	100+1=101	101/101=1.0000	1
9	((1×101)+23) mod 101=22	22+1=23	23/101=0.2277	33
10	((1×23)+23) mod 101=45	45+1=46	46/101=0.4554	23
11	((1×46)+23) mod 101=68	68+1=69	69/101=0.6832	13
12	((1×69)+23) mod 101=91	91+1=92	92/101=0.9109	4
13	((1×92)+23) mod 101=13	13+1=14	14/101=0.1386	36
14	((1×14)+23) mod 101=36	36+1=37	37/101=0.3663	27
15	((1×37)+23) mod 101=59	59+1=60	60/101=0.5941	17
16	((1×60)+23) mod 101=82	82+1=83	83/101=0.8218	8
17	((1×83)+23) mod 101=4	4+1=5	5/101=0.0495	40
18	((1×5)+23) mod 101=27	27+1=28	28/101=0.2772	31
19	((1×28)+23) mod 101=50	50+1=51	51/101=0.5050	21
20	((1×28)+23) mod 101=73	73+1=74	74/101=0.7327	11
21	((1×74)+23) mod 101=96	95+1=97	97/101=0.9604	2
22	((1×97)+23) mod 101=18	18+1=19	19/101=0.1881	34
23	((1×19)+23) mod 101=41	41+1=42	42/101=0.4158	24
24	((1×42)+23) mod 101=64	64+1=65	65/101=0.6436	14
25	((1×65)+23) mod 101=87	87+1=88	88/101=0.8713	5
26	((1×88)+23) mod 101=9	9+1=10	10/101=0.0990	37
27	((1×10)+23) mod 101=32	32+1=33	33/101=0.3267	28
28	((1×33)+23) mod 101=56	56+1=57	57/101=0.5545	18
29	((1×57)+23) mod 101=78	238+1=79	79/101=0.7822	9
30	((1×79)+23) mod 101=0	0+1=1	1/101=0.0099	41
31	((1×1)+23) mod 101=23	23+1=24	24/101=0.2376	32
32	((1×24)+23) mod 101=46	46+1=47	47/101=0.4653	22
33	((1×302)+23) mod 101=69	69+1=70	70/101=0.6931	12
34	((1×323)+23) mod 101=92	92+1=93	93/101=0.9208	3
35	((1×93)+23) mod 101=14	14+1=15	15/101=0.1485	35
36	((1×15)+23) mod 101=37	37+1=38	38/101=0.3762	26
37	((1×38)+23) mod 101=60	60+1=61	61/101=0.6040	16
38	((1×61)+23) mod 101=83	83+1=84	84/101=0.8317	7
39	((1×84)+23) mod 101=5	5+1=6	6/101=0.0594	39
40	((1×6)+23) mod 101=28	28+1=29	29/101=0.2871	30

Table 4. Randomization results of problem packages 1-10 using MRNG

	Question package 1-10									
No	1	2	3	4	5	6	7	8	9	10
1	25	**21**	25	24	**21**	20	22	**19**	**19**	18
2	15	11	14	12	**8**	6	**8**	4	3	1
3	6	1	3	41	37	34	35	30	28	26
4	38	32	33	30	25	21	20	15	12	9
5	29	**22**	**22**	18	12	7	6	41	37	33
6	19	12	11	6	40	35	33	26	22	16
7	10	2	41	35	28	22	18	11	6	40
8	1	33	30	23	15	8	4	37	31	24
9	33	23	19	11	2	36	31	22	15	7
10	23	13	8	40	31	23	16	7	40	32
11	13	3	38	29	18	9	2	33	25	15
12	4	34	27	17	5	37	29	18	9	39
13	36	**24**	16	5	**34**	**24**	14	3	**34**	23
14	27	14	5	34	22	10	41	29	18	6
15	17	4	35	22	9	38	27	14	2	31
16	8	35	24	10	38	25	12	40	27	14
17	40	25	13	39	26	11	39	25	11	38
18	31	15	2	28	13	39	25	10	36	22
19	21	**5**	32	16	41	26	11	36	21	**5**
20	11	36	**21**	4	29	12	38	**21**	5	30
21	2	26	10	33	16	40	24	6	30	13
22	34	16	40	21	3	27	10	32	14	37

Table 4. Randomization results of problem packages 1-10 using MRNG (*continue*)

	Question Package 1-10									
No	1	2	3	4	5	6	7	8	9	10
23	24	6	29	9	32	13	37	17	39	21
24	14	37	18	38	19	41	23	2	24	4
25	5	**27**	7	**27**	6	**28**	9	**28**	8	29
26	**37**	17	**37**	15	35	14	36	13	33	12
27	28	7	26	3	23	1	21	39	17	36
28	18	38	15	32	10	29	7	24	1	20
29	**9**	28	4	20	39	15	34	**9**	26	3
30	41	18	34	8	27	2	19	35	10	28
31	32	8	23	37	14	30	5	20	35	11
32	22	39	12	26	1	16	32	5	20	35
33	12	29	1	14	30	3	17	31	4	19
34	**3**	19	31	2	17	31	**3**	16	29	2
35	35	9	20	31	4	17	30	1	13	27
36	26	40	9	19	33	4	15	27	38	10
37	16	30	39	7	20	32	1	12	23	34
38	**7**	20	**28**	36	**7**	18	**28**	38	**7**	17
39	39	10	17	25	36	5	13	23	32	41
40	30	41	6	13	24	33	40	8	16	25

Table 5. Same question package MRNG

No	Question package	Initial question number	New question number
1	2	1	21
2	5	1	21
3	**8**	**1**	**19**
4	**9**	**1**	**19**
5	5	2	8
6	7	2	8
7	**2**	**7**	**22**
8	**3**	**7**	**22**
9	2	13	24
10	6	13	24
11	**5**	**13**	**34**
12	**9**	**13**	**34**
13	2	19	5
14	10	19	5
15	**3**	**20**	**21**
16	**8**	**20**	**21**
17	2	25	27
18	4	25	27
19	**6**	**25**	**28**
20	**8**	**25**	**28**
21	1	26	37
22	3	26	37
23	**1**	**29**	**9**
24	**8**	**29**	**9**
25	1	34	3
26	7	34	3
27	**1**	**38**	**7**
28	**5**	**38**	**7**
29	**9**	**38**	**7**
30	3	38	28
31	7	38	28

REFERENCES

[1] K. Mahmood, T. Tariq, A. K. Sangaiah, Z. Ghaffar, M. A. Saleem, and S. Shamshad, “A neural computing-based access control protocol for AI-driven intelligent flying vehicles in industry 5.0-assisted consumer electronics,” *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2023, doi: 10.1109/TCE.2023.3276066.

[2] A. Idaman, Roslina, and R. Rosnelly, “Implementation of linear congruent methods and multiplication random numbers for academic potential tests,” *International Journal of Research in Vocational Studies (IJRVOCAS)*, vol. 2, no. 4, pp. 32–41, Jan. 2023, doi: 10.53893/ijrvocas.v2i4.160.

[3] N. H., F. Febriati, and E. Ervianti, “The impact of computer-based test and students’ ability in computer self - efficacy on mathematics learning outcomes,” *Journal of Education Technology*, vol. 5, no. 4, p. 603, Nov. 2021, doi: 10.23887/jet.v5i4.34942.

[4] F. Artuğer and F. Özkaynak, “An effective method to improve nonlinearity value of substitution boxes based on random selection,” *Information Sciences*, vol. 576, pp. 577–588, Oct. 2021, doi: 10.1016/j.ins.2021.07.036.

[5] S. D. Simic, R. Sajina, N. Tankovic, and D. Etinger, “A review on generating random numbers in decentralised environments,” in *2020 43rd International Convention on Information, Communication and Electronic Technology, MIPRO 2020 - Proceedings*, Sep. 2020, pp. 1668–1673, doi: 10.23919/MIPRO48935.2020.9245302.

[6] S. A. Jassim and A. K. Farhan, "Combined Chebyshev and logistic maps to generate pseudorandom number generator for internet of things," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 3, pp. 3287–3297, Jun. 2022, doi: 10.11591/ijece.v12i3.pp3287-3297.

[7] M. Saber and M. M. Eid, "Low power pseudo-random number generator based on lemniscate chaotic map," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 1, pp. 863–871, Feb. 2021, doi: 10.11591/ijece.v11i1.pp863-871.

[8] O. Z. Akif, S. M. Ali, R. S. Ali, and A. K. Farhan, "A new pseudorandom bits generator based on a 2d-chaotic system and diffusion property," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 1580–1588, Jun. 2021, doi: 10.11591/eei.v10i3.2610.

[9] P. G. Saraswathy and R. Siddaiah, "Randomness properties of sequence generated using logistic map with novel permutation and substitution techniques," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 4, pp. 4369–4378, Aug. 2023, doi: 10.11591/ijece.v13i4.pp4369-4378.

[10] F. James and L. Moneta, "Review of high-quality random number generators," *Computing and Software for Big Science*, vol. 4, no. 1, p. 2, Dec. 2020, doi: 10.1007/s41781-019-0034-3.

[11] P. Kietzmann, T. C. Schmidt, and M. Wählisch, "A guideline on pseudorandom number generation (PRNG) in the IoT," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, Jul. 2021, doi: 10.1145/3453159.

[12] S. Cang, Z. Kang, and Z. Wang, "Pseudo-random number generator based on a generalized conservative Sprott-A system," *Nonlinear Dynamics*, vol. 104, no. 1, pp. 827–844, Mar. 2021, doi: 10.1007/s11071-021-06310-9.

[13] X. Tian, R. Ding, X. Wu, and G. Bai, "Hardware implementation of a cryptographically secure pseudo-random number generators based on Koblitz elliptic curves," in *2020 IEEE 3rd International Conference on Electronics Technology, ICET 2020*, May 2020, pp. 91–94, doi: 10.1109/ICET49382.2020.9119643.

[14] A. A. Rezk, A. H. Madian, A. G. Radwan, and A. M. Soliman, "Multiplierless chaotic pseudo random number generators," *AEU - International Journal of Electronics and Communications*, vol. 113, p. 152947, Jan. 2020, doi: 10.1016/j.aeue.2019.152947.

[15] S. Krishnamoorthi, P. Jayapaul, R. K. Dhanaraj, V. Rajasekar, B. Balusamy, and S. H. Islam, "Design of pseudo-random number generator from turbulence padded chaotic map," *Nonlinear Dynamics*, vol. 104, no. 2, pp. 1627–1643, Apr. 2021, doi: 10.1007/s11071-021-06346-x.

[16] M. Kostrzewski, "Sensitivity analysis of selected parameters in the order picking process simulation model, with randomly generated orders," *Entropy*, vol. 22, no. 4, p. 423, Apr. 2020, doi: 10.3390/E22040423.

[17] Y. A. Izotov, A. A. Velichko, and P. P. Boriskov, "Method for fast classification of MNIST digits on Arduino UNO board using LogNNet and linear congruential generator," *Journal of Physics: Conference Series*, vol. 2094, no. 3, p. 032055, Nov. 2021, doi: 10.1088/1742-6596/2094/3/032055.

[18] N. Akhila, C. U. Kumari, K. Swathi, T. Padma, and N. M. Rao, "Performance analysis of pseudo random bit generator using modified dual-coupled linear congruential generator," in *2021 International Conference on Intelligent Technologies, CONIT 2021*, Jun. 2021, pp. 1–5, doi: 10.1109/CONIT51480.2021.9498354.

[19] S. Wibawani *et al.*, "Information system web based for group randomization of state defense education class using linear congruential method," May 2022, doi: 10.11594/nstp.2022.2420.

[20] R. Cybulski, "Pseudo-random number generator based on linear congruence and delayed Fibonacci method," *Technical Sciences*, no. 2021, Dec. 2021, doi: 10.31648/ts.7238.

[21] N. A. Ali, A. M. S. Rahma, and S. H. Shaker, "Multi-level encryption for 3D mesh model based on 3D Lorenz chaotic map and random number generator," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 6486–6495, Dec. 2022, doi: 10.11591/ijece.v12i6.pp6486-6495.

[22] K. Liao *et al*., "A multiple linear regression model with multiplicative log-normal error term for atmospheric concentration data," *Science of the Total Environment*, vol. 767, p. 144282, May 2021, doi: 10.1016/j.scitotenv.2020.144282.

[23] O. L. Suryo, M. Syahrizal, and A. Fau, "Modification of the international data encryption algorithm key algorithm uses a multiplicative random number generator to secure digital images (in Indonesia: *Modifikasi kunci algoritma international data encryption algorithm menggunakan multiplicative random number generator untuk mengamankan citra digital*)," *KOMIK (Konferensi Nasional Teknologi Informasi dan Komputer)*, vol. 4, no. 1, pp. 326–333, 2020, doi: 10.30865/komik.v4i1.2715.

[24] S. K.B. and G. Aithal, "Generation of pseudo random number sequence from discrete oscillating samples of equally spread objects and application for stream cipher system," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 1, Jan. 2020, doi: 10.1002/cpe.5181.

[25] A. S. Chauhan, V. Sahula, A. S. Mandal, and A. Dutta, "Intensifying challenge obfuscation by cascading FPGA RO-PUFs for random number generation," in *2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems (VLSID)*, Jan. 2020, pp. 195–200, doi: 10.1109/VLSID49098.2020.00051.

[26] L. Pasqualini and M. Parton, "Pseudo random number generation: A reinforcement learning approach," *Procedia Computer Science*, vol. 170, pp. 1122–1127, 2020, doi: 10.1016/j.procs.2020.03.057.

BIOGRAPHIES OF AUTHORS

Amrullah born in Medan in 1986. Diploma 3 graduation at STMIK Triguna Dharma Medan in the field of Informatics Management in 2014, Undergraduate Graduation at Triguna Dharma Medan in the Field of Information Systems Science in 2016 and completed the Computer Masters program, Faculty of Computer Science, Informatics Engineering study program concentrating on Information Systems at UPI YPTK Padang in 2019. As a lecturer, he actively writes in various national and international journals. Apart from being a lecturer at FIKTI, he is also active as an author on international digital asset design websites and typography fonts which can be viewed on myfont.com. His works are also spread across various other national marketplaces such as envato, creative market. He can be contacted at email: amrullah@umsu.ac.id.

Dr. Al-Khowarizmi is a son born in 1992 in Medan City who has completed his last education at the University of North Sumatra (USU) in the Informatics Engineering Masters Study Program (S2) in 2017. However, in his journey he has been active as an IT practitioner in Sumatra Province and has become an expert in several regencies/cities. In addition, he is also active in conducting research in the fields of artificial intelligence, data mining, neural networks and machine learning in reputable national and international journals and conferences. In addition, he is also active in association activities such as holding the position of Deputy Chair of the IPKIN (Computer and Informatics Professional Association) North Sumatra Branch, PRSI Medan City, and so on. In order to deepen his knowledge and career in 2020, he ventured to take further studies in the Computer Science Doctoral (S3) program at the University of North Sumatra. He can be contacted at email: alkhowarizmi@umsu.ac.id.

Dr. Firahmi Rizky has completed his last education at Padang State University (UNP) in the Doctoral Study Program (S3). In addition, he is also active in conducting research in the fields of artificial intelligence, data mining, neural networks and machine learning in national and international reputable journals and conferences. She can be contacted at email: firahmirizky@umsu.ac.id.