

Implementasi Algoritma *Multiply With Carry Generator* (MWCG) dalam Pengacakan Soal Ujian Semester Berbasis Web pada SMKN 1 Kendari

Rizal Adi Saputra¹, Iklil Awalda Tariza², and Bambang Pramono³

¹Teknik Informatika, Universitas Halu Oleo, Kampus Hijau Bumi Tridharma Anduonohu, Jl. Hea Mokodompit Kendari, 93231
e-mail: ¹rizaladisaputra@uho.ac.id, ²ikliltariza280599@gmail.com, ³bambangpramono09@gmail.com

Submitted Date: November 09th, 2021
Revised Date: February 15th, 2022

Reviewed Date: January 07th, 2022
Accepted Date: March 31st, 2022

Abstract

A semester exam conducted using google form on SMK 1 Kendari could be used as a loophole for cheating. It requires a system that can randomize the question using specified algorithm. *Multiply With Carry Generator* algorithm is a simple interlinear numbers algorithm that can be used to solve this problem. In several previous studies, this algorithm has been widely used to generate random numbers in several programming applications. They are outgrown by the algorithm *Multiply With Carry Generator*, which is at the speed of generating random numbers and also the length of the randomness cycle produced by *Multiply With Carry Generator*, which uses two seeds to produce more seed variations. Based on testing runs perform to test the randomize of the algorithm *Multiply With Carry Generator* in 10 students with 20 numbers of question produce randomize question with test results received H_0 . It's prove that *Multiply With Carry Generator* algorithm can be used to randomize the question and can be implemented in SMK 1 Kendari exam test system.

Keywords: *Multiply With Carry Generator*; Randomness; Midterms; Runt Test

Abstrak

Ujian semester yang dilakukan dengan menggunakan *google form* pada SMKN 1 Kendari dapat dimanfaatkan untuk melakukan kecurangan karena adanya kesamaan soal pada setiap siswa, untuk itu dibutuhkan sebuah sistem ujian semester yang dapat mengacak soal dan menerapkan sebuah Algoritma yang dapat mengacak soal pada sistem ujian semester. Algoritma *Multiply With Carry Generator* (MWCG) adalah sebuah algoritma pembangkit bilangan acak semu yang sederhana. Pada beberapa penelitian terdahulu algoritma ini telah banyak digunakan untuk membangkitkan bilangan acak pada beberapa aplikasi pemrograman. Keunggulan dari Algoritma *Multiply With Carry Generator* yaitu pada kecepatan dalam pembangkitan bilangan acak dan juga panjang siklus pengacakan yang dihasilkan oleh *Multiply With Carry Generator* menggunakan dua buah *seed* sehingga memiliki variasi *seed* yang lebih banyak. Berdasarkan pengujian *Run Test* yang dilakukan untuk menguji keacakan dari Algoritma *Multiply With Carry Generator* (MWCG) pada 10 siswa dengan jumlah soal 20 nomor secara bersamaan menghasilkan bentuk soal acak, dengan hasil uji H_0 diterima. Hal tersebut membuktikan Algoritma *Multiply With Carry Generator* dapat digunakan dalam pengacakan soal.

Kata kunci: *Multiply With Carry Generator*; Pengacakan Soal; Ujian Semester; *Run Test*

1. Pendahuluan

Seiring perkembangan zaman di negara yang semakin maju, teknologi sudah banyak digunakan dan dikembangkan dalam kehidupan sehari-hari contohnya dibidang pendidikan. Pemerintah telah berupaya untuk membangun Sumber Daya Manusia (SDM) agar lebih memfokuskan

pendidikan sekolah yang terampil dan bermutu sehingga nantinya akan menjadi siswa-siswi yang berprestasi dan mempunyai intelektual tinggi. Kualitas pendidikan adalah salah satu hal yang harus diutamakan dalam pendidikan terutama untuk masa depan penerus bangsa, sehingga peran dari sekeliling siswa-siswi sangat berpengaruh

untuk memotivasi dan mendukung kualitas pendidikan itu sendiri (Salesiano et al., 2021).

Tingkat kualitas pendidikan juga dapat dilihat dengan cara melakukan tes atau pengujian soal-soal kepada siswa-siswi untuk dapat melihat sejauh mana kemampuan dan tingkat memahami bidang studi yang sudah dipelajarinya. Perkembangan teknologi informasi dan komunikasi semakin hari semakin baik, ujian yang dilakukan siswa-siswi sudah tidak lagi menggunakan kertas sebagai media untuk melaksanakan ujian melainkan sudah menggunakan komputer sebagai media untuk mengerjakan berbagai macam ujian. Pelaksanaan ujian yang medianya masih menggunakan kertas sangat rentan terjadi kecurangan-kecurangan sebelum pelaksanaan ujian dimulai. (Arizqia & Widodo, 2017).

SMKN 1 Kendari menerapkan sistem ujian semester *online* dengan menggunakan *google form*, kesempatan ini juga banyak dimanfaatkan untuk bisa bekerjasama dalam melaksanakan ujian karena soal yang dimiliki setiap siswa-siswi adalah sama. Pembagian *link* soal ujian juga hanya dibagikan satu *link google form* untuk satu mata pelajaran sehingga urutan soal yang muncul pada setiap siswa-siswi adalah sama. Sehingga rentan terjadinya kecurangan dan kebocoran soal antara siswa-siswi karena soal ujian yang sama. Maka dibutuhkan sebuah sistem ujian yang dapat mengacak soal sehingga soal yang didapatkan setiap siswa berbeda-beda.

Sistem pengacakan soal ujian semester ini akan mempermudah pengerjaan ujian dengan menggunakan *website* untuk sarana beberapa pengerjaan ujian seperti ulangan harian, ulangan tengah semester, dan ulangan semester bisa dilakukan. Pada sistem pengacakan soal ujian sekolah yang berbasis *web* ini perlu mengimplementasikan sebuah algoritma yang tepat dalam hal pengacakan soal ujian yang berbentuk *multiple choice* yang akan ditampilkan untuk menghindari kesamaan soal ujian. Pemilihan algoritma yang tepat untuk sistem pengacakan soal ini yaitu Algoritma *Multiply With Carry Generator*, algoritma ini akan digunakan untuk mengacak semua soal-soal pada ujian semester agar setiap soal yang keluar berbeda-beda pada setiap siswanya dan untuk menghindari siswa-siswi melakukan kecurangan, sehingga nantinya soal yang muncul pada saat mengerjakan ujian tidak memiliki kesamaan. Sistem ujian semester ini juga memberikan hasil evaluasi secara *real time* atau dapat diketahui langsung ketika siswa-siswi sudah

selesai mengerjakan soal ujian tanpa harus menunggu berhari-hari.

Algoritma *Multiply With Carry Generator* adalah sebuah algoritma pembangkit bilangan acak semu yang sederhana. Algoritma *Multiply With Carry Generator* adalah algoritma yang telah banyak digunakan untuk membangkitkan bilangan acak pada beberapa aplikasi pemrograman. Algoritma ini dikembangkan oleh George Marsaglia, seorang pakar dalam pembangkitan bilangan acak Keunggulan dari Algoritma *Multiply With Carry Generator* yaitu pada kecepatan dalam pembangkitan bilangan acak dan juga panjang siklus pengacakan yang dihasilkan oleh *Multiply With Carry Generator* menggunakan dua buah *seed* sehingga memiliki variasi *seed* yang lebih banyak (Rivaldo & Herriyance, 2021).

2. Metode Penelitian

2.1 Penjelasan Ujian

Ujian merupakan salah satu cara untuk mengetahui tingkat pemahaman seseorang terhadap pelajaran yang sudah ditempuhnya, dengan melakukan ujian seseorang dapat mengetahui capaian atau hasil dari pemahaman pelajaran sebelumnya, sehingga dapat mengevaluasi kualitas belajar. (Pratomo & Mantala, 2016).

Ujian adalah salah satu tes yang tujuannya dapat melihat kemampuan dan pemahaman seseorang. Pada umumnya ujian bisa dilaksanakan dengan tertulis, wawancara, ataupun praktek sesuai dengan kemampuan masing-masing. Pengerjaan soal ujian dikerjakan untuk peserta ujian dan menggunakan target kelulusan sebagai acuan kelulusan untuk seorang yang melakukan ujian. (T. J. C. Sitanggang, 2016).

Dari penjelasan tersebut dapat disimpulkan bahwa ujian adalah cara mengukur pengetahuan seseorang yang sudah dimiliki, agar nantinya dapat melihat bagaimana perkembangan pembelajaran yang sudah di pahami dan juga bisa memotivasi seseorang untuk melakukan pembelajaran yang lebih giat dan baik lagi dari sebelumnya.

2.2 Pengertian Website

Pada umumnya website merupakan singkatan dari *World Wide Web* (WWW). Pemberitahuan *World Wide Web* itu sendiri disampaikan di web server agar bisa diakses pada beberapa browser contohnya *Explorer Mozilla Firefox*, Chrome, dan lain sebagainya.

Website itu sendiri dapat didefinisikan sebagai kumpulan tampilan website dalam suatu

domain yang dapat memberikan berbagai macam informasi dan berbagai macam visualisasi baik itu yang bersifat statis ataupun dinamis, website juga berfungsi sebagai sarana informasi untuk mengetahui informasi-informasi yang dibagikan secara online. (Hartono, 2013).

2.3 Pembangkit Bilangan Acak

Bilangan acak adalah sebuah bilangan yang hasilnya tidak dapat diprediksi atau diketahui, seperti contohnya dalam kehidupan sehari-hari yaitu peluang munculnya mata dadu pada saat pelemparan sebuah dadu. Jika mengambil contoh 5 kali pelemparan sebuah dadu, maka hasil pelemparan dalam satu kali kemungkinan akan terlihat salah satu angka dari 1,2,3,4,5,6, begitu pun untuk percobaan berikutnya (Samsuriati et al., 2019).

Pembangkit bilangan acak (*Random Number Generator*) adalah sebuah perhitungan algoritma yang digunakan untuk dapat menghasilkan bilangan urut dari hasil perhitungan komputer sehingga nantinya akan muncul bilangan acak secara terus menerus (Mulya et al., 2020).

2.4 Multiply With Carry Generator

Algoritma *Multiply With Carry Generator* adalah sebuah algoritma pembangkit bilangan acak semu yang sederhana. Algoritma *Multiply With Carry Generator* adalah algoritma yang telah banyak digunakan untuk membangkitkan bilangan acak pada beberapa aplikasi pemrograman. Algoritma ini dikembangkan oleh George Marsaglia, seorang pakar dalam pembangkitan bilangan acak. Agar dapat menghasilkan bilangan acak, maka memperoleh persamaan sebagai berikut (Marsaglia, 2003).

$$X_n = (a X_{n-1} + C_{n-1}) \bmod b \quad (1)$$

$$C_n = \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \quad (2)$$

Keterangan :

X_n = Bilangan acak ke – (n) dari barisan bilangan acak

X_{n-1} = Bilangan acak ke – (n-1)

a = Konstanta pengali

C_n = Konstanta kenaikan ke – (n)

C_{n-1} = Konstanta kenaikan ke – (n-1)

b = Konstanta modulus dan pembagi

Agar algoritma dapat berjalan dengan baik, dalam arti dapat menghasilkan siklus terpanjang

yang mungkin terdapat sejumlah syarat yang harus dipenuhi, yakni :

1. $C_0 < b$, dan
2. $ab-1$ adalah bilangan prima

Sebagai contoh implementasi untuk membangkitkan 5 bilangan acak dengan nilai $a=28$, $b=300$, $X_0=33$, $C_0=18$ menggunakan rumus pada algoritma *Multiply With Carry Generator* (MWCG).

$$\begin{aligned} X_1 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.33 + 18) \bmod 300 \\ &= 942 \bmod 300 \\ &= 42 \end{aligned}$$

$$\begin{aligned} C_1 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.33+18}{300} \right\rfloor \\ &= \frac{942}{300} \\ &= 3 \end{aligned}$$

$$\begin{aligned} X_2 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.42 + 3) \bmod 300 \\ &= 1179 \bmod 300 \\ &= 279 \end{aligned}$$

$$\begin{aligned} C_2 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.42+3}{300} \right\rfloor \\ &= \left\lfloor \frac{1179}{300} \right\rfloor \\ &= 4 \end{aligned}$$

$$\begin{aligned} X_3 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.279 + 4) \bmod 300 \\ &= 7816 \bmod 300 \\ &= 16 \end{aligned}$$

$$\begin{aligned} C_3 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.279+4}{300} \right\rfloor \\ &= \left\lfloor \frac{7816}{300} \right\rfloor \\ &= 36 \end{aligned}$$

$$\begin{aligned} X_4 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.16 + 36) \bmod 300 \\ &= 484 \bmod 300 \\ &= 184 \end{aligned}$$

$$\begin{aligned} C_4 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.16+36}{300} \right\rfloor \\ &= \left\lfloor \frac{484}{300} \right\rfloor \\ &= 2 \end{aligned}$$

$$\begin{aligned} X_5 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.184 + 2) \bmod 300 \\ &= 2354 \bmod 300 \\ &= 254 \\ C_5 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.184 + 2}{300} \right\rfloor \\ &= \left\lfloor \frac{2354}{300} \right\rfloor \\ &= 8 \end{aligned}$$

$$\begin{aligned} X_6 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.254 + 8) \bmod 300 \\ &= 7120 \bmod 300 \\ &= 220 \\ C_6 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.254 + 8}{300} \right\rfloor \\ &= \left\lfloor \frac{7120}{300} \right\rfloor \\ &= 24 \end{aligned}$$

$$\begin{aligned} X_7 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.220 + 24) \bmod 300 \\ &= 6185 \bmod 300 \\ &= 185 \\ C_7 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.220 + 24}{300} \right\rfloor \\ &= \left\lfloor \frac{6185}{300} \right\rfloor \\ &= 21 \end{aligned}$$

$$\begin{aligned} X_8 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.185 + 21) \bmod 300 \\ &= 5201 \bmod 300 \\ &= 101 \\ C_8 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.185 + 21}{300} \right\rfloor \\ &= \left\lfloor \frac{5201}{300} \right\rfloor \\ &= 17 \end{aligned}$$

$$\begin{aligned} X_9 &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.101 + 17) \bmod 300 \\ &= 2845 \bmod 300 \\ &= 145 \\ C_9 &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.101 + 17}{300} \right\rfloor \\ &= \left\lfloor \frac{2845}{300} \right\rfloor \\ &= 9 \end{aligned}$$

$$\begin{aligned} X_{10} &= (a X_{n-1} + C_{n-1}) \bmod b \\ &= (28.154 + 9) \bmod 300 \\ &= 4069 \bmod 300 \\ &= 169 \\ C_{10} &= \left\lfloor \frac{a X_{n-1} + C_{n-1}}{b} \right\rfloor \\ &= \left\lfloor \frac{28.154 + 9}{300} \right\rfloor \\ &= \left\lfloor \frac{4069}{300} \right\rfloor \\ &= 13 \end{aligned}$$

Maka hasil yang didapatkan dari pembangkitan bilangan acak diatas adalah 42, 279, 16, 184, 254, 220, 185, 101, 145, 169.

2.5 PHP (*Hypertext Preprocessing*)

PHP singkatan dari *Hypertext Preprocessing* adalah suatu bahasa dalam sebuah pemrograman yang menggunakan *open source*, bahasa ini sangat tepat digunakan untuk pengembangan sebuah *website* dan dapat ditanamkan pada penggunaan berbagai macam sistem contohnya pengerjaan tugas akhir kuliah. Bahasa pada PHP bisa menjelaskan beberapa bahasa pemrograman contohnya C, Java, dan Perl dan bisa untuk dimengerti. *Hypertext Preprocessing* atau PHP juga dapat diartikan sebagai salah satu bahasa pemrograman yang menggunakan *code*, dari *code* tersebut dapat dikelola dan kemudian *code* tersebut akan menjadi HTML.

PHP terus mengalami perkembangan, saat menulis pembaruan terbaru pada PHP 7.4 (*Hypertext Preprocessing Version 7.4*) dengan berbagai fitur baru. PHP 7.4 memiliki fitur terbaru yang sangat dinanti yaitu *preloading*. Fitur ini akan digunakan untuk mempermudah pengguna dan mempercepat proses hasil *script*, dan juga dapat mengenakan kelebihan kode yang lebih cepat. Kasus ini bisa terjadi karena adanya sebuah penyederhanaan pada barik-baris *code* yang umum.

2.6 HTML (*Hyper Text Markup Language*)

HTML yang merupakan kepanjangan dari *Hyper Text Markup Language* adalah kumpulan sebuah kode-kode pemrograman dasar hasil dari representasi sebuah *website*. Pada sebuah *website* menggunakan beberapa tanda-tanda atau *tag* tertentu, di mana tanda-tanda atau *tag* digunakan untuk memperjelas kode-kode pada sebuah *website* (Constantianus & Suteja, 2005).

HTML 5 (*Hyper Text Markup Language Version 5*) adalah sebuah bahasa pemrograman yang dibangun dari *Word Wide Web*, sebuah

teknologi yang berperan banyak dalam internet. HTML5 melengkapi bagian-bagian lama yang sebelumnya masih banyak kekurangan, dengan menambahkan berbagai fitur yang menyuport agar pembuatan sebuah *website* lebih baik lagi.

2.7 Xampp

Xampp adalah perangkat lunak yang sifatnya *open source* dan mendukung begitu banyak aplikasi, fungsi dari xampp itu sendiri untuk mengembangkan beberapa software agar lebih mudah dan terstruktur dengan baik. Nama *Xampp* merupakan singkatan dari X (empat sistem operasi apapun) *Apache*, *MySQL*, *PHP* dan *Perl*.

2.8 Uji Keacakan (*Run Test*)

Proses dikatakan terkendali jika semua ukuran mutu (kualitas) produk berada diantara batas kendali dan menyebar secara *random*, maka untuk menguji keacakan digunakan metode uji keacakan (*run test*).

Uji keacakan adalah salah satu pengujian yang digunakan untuk melihat hasil dari sampel yang sudah diperoleh sebelumnya dari hasil penelitian apakah sampel tersebut benar-benar teracak atau tidak. Aturan untuk melihat keacakan biasanya dinilai dari data, sifat atau runtun yang didefinisikan sebagai serangkaian kejadian, hal atau tanda-tanda yang beda. Pengacakan data dikatakan valid apabila rangkaian penentu akhirnya sedikit dan terlalu banyak (Pramantya, 2017).

Run test dapat digunakan dengan langkah-langkah sebagai berikut:

1. Menentukan Hipotesis
Terdapat tiga bentuk rumusan hipotesis, yaitu :
 - a. Uji Hipotesis dua pihak
 H_0 : Runtutan bersifat acak
 H_1 : Runtutan tidak bersifat acak
 - b. Uji Hipotesis satu pihak kiri
 H_0 : Runtutan bersifat acak
 H_1 : Runtutan terlalu sedikit
 - c. Uji Hipotesis satu pihak kanan
 H_0 : Runtutan bersifat acak
 H_1 : Runtutan terlalu banyak
2. Menentukan tingkat signifikan
Tingkat signifikan juga dinotasikan α dan biasanya digunakan 5% atau 0,05.
3. Statistik Uji
 - a. Tentukan nilai median data
 - b. Untuk data yang $>$ median, beri tanda +

- a. Untuk data yang $<$ median, beri tanda -
- b. Untuk data yang $=$ median, beri tanda 0
- c. Setelah data dinyatakan dalam tanda + dan -, selanjutnya tentukan banyaknya *run* dalam urutan data tersebut (urutan data tidak boleh diubah).
- d. *Run* yaitu banyaknya runtun (suatu urutan lambang-lambang yang sama yang diikuti ataupun mengikuti lambang-lambang berikutnya yang berbeda).

Keterangan:

- n_1 = Banyaknya data bertanda (+) atau yang berada diatas rata-rata
 n_2 = Banyaknya data bertanda (-) atau yang berada dibawah rata-rata
 r = Banyaknya *run* dalam urutan

4. Menentukan Daerah kritis
Membandingkan nilai tabel (r_a dan r_b) dengan nilai statistik ujinya. Nilai kritis suatu uji hipotesis sesuai bentuk rumusan hipotesis, yaitu H_0 ditolak jika :
 - a. $r \leq r_a$ atau $r \geq r_b$
 - b. $r \leq r_a$
 - c. $r \geq r_b$

Di mana:

- r : Nilai statistik uji
 r_a : Nilai kritis bawah
 r_b : Nilai kritis atas
 r_a dan r_b diperoleh dari tabel uji keacakan Swed dan Eisenhart, 1943.

5. Membuat Kesimpulan
Keputusan ditolak atau diterima H_0 dilakukan setelah melihat perbandingan nilai hasil perhitungan statistik uji r dengan nilai kritis.

3. Hasil Dan Pembahasan

3.1 Data

Data-data yang diambil dan dijadikan sebagai sampel ini adalah soal mata pelajaran Bahasa Indonesia, Bahasa Inggris, Kewirausahaan, dan Matematika yang disimpan dalam basis data *website* ujian semester. Soal yang diperoleh dari hasil studi literatur ke tempat penelitian, masing-masing guru mata pelajaran yang akan dijadikan *sample* memberikan soal untuk dijadikan data pada *website* ujian semester. Jumlah keseluruhan soal adalah 1.200 soal. Soal terdiri dari 3 kategori soal yaitu mudah, menengah, dan sulit.

3.2 Implementasi Interface

Implementasi *interface* dibuat dan dirancang berdasarkan rancangan *interface* pada *mockup* yang sudah di desain. Implementasi *interface* admin, guru dan siswa ditampilkan dalam bentuk *website*.

1. Tampilan Login

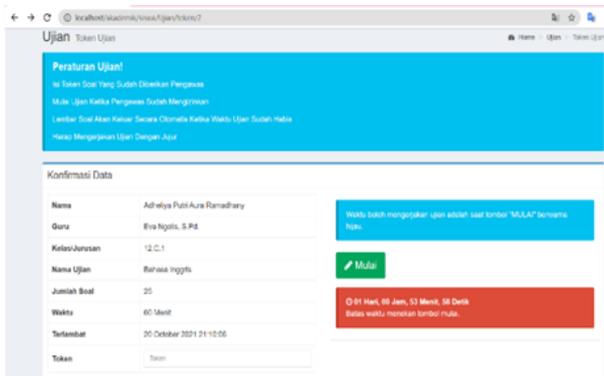
Gambar 1 adalah halaman login yang bisa dilakukan oleh admin dan user sebelum melakukan login admin, guru, dan siswa harus mengisi *email* dan *password* terlebih dahulu agar bisa mengakses *website* ujian semester.



Gambar 1. Halaman Login

2. Halaman Ikut Ujian Siswa

Gambar 2 adalah tampilan pengisian token ujian siswa yang sudah diberikan oleh pengawas, setelah siswa sudah mengisi token ujian dan token tersebut benar maka sistem akan mengacak soal menggunakan Algoritma *Multiply With Carry Generator* (MWCG) dan siswa dapat mulai melakukan ujian sesuai dengan instruksi dari pengawas.

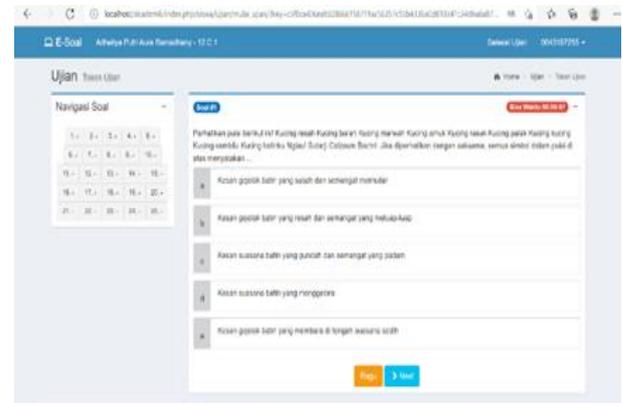


Gambar 2. Ikut Ujian Siswa

3. Halaman Ujian Siswa

Gambar 3 adalah tampilan ujian siswa yang berisikan navigasi soal sebelah kiri sesuai dengan nomor soal yang akan dikerjakan, lembar

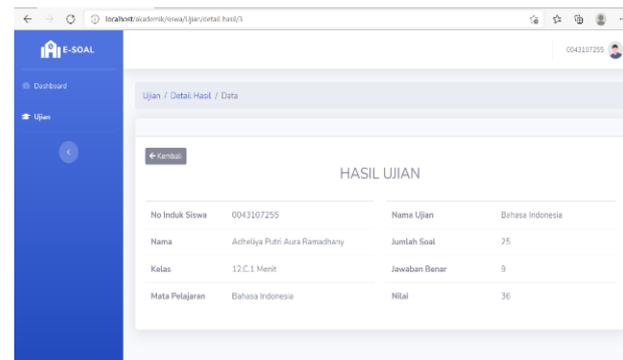
soal sebelah kanan dengan pilihan option a,b,c,d,e dan terdapat juga fitur ragu, *next*, dan selesai ujian.



Gambar 3. Halaman Ujian Siswa

4. Halaman Hasil Ujian

Gambar 4 adalah tampilan hasil ujian siswa yang berisikan data-data hasil ujian siswa seperti jumlah soal, jumlah benar, dan nilai siswa.



Gambar 4. Halaman Hasil Ujian

3.3 Pengujian Pengacakan (*Run Test*)

Dalam pengujian pengacakan (*Run Test*) ini bertujuan untuk melihat hasil dari pengacakan soal pada setiap siswanya, pengacakan soal dengan menggunakan 300 data soal pada *database* setiap mata pelajaran, berikut pengujian pengacakan (*run test*) yang di lakukan kepada 10 siswa dengan jumlah soal acak 20 nomor yang dilakukan secara bersamaan.

Tabel 1 Analisis Pengujian *Run Test*

Percobaan	Nilai Keputusan	Hasil Keputusan
Siswa 1	$6 \leq 8 \leq 16$	H_0 diterima dan teracak
Siswa 2	$6 \leq 10 \leq 16$	H_0 diterima dan teracak

Siswa 3	$6 \leq 12 \leq 16$	H_0 diterima dan teracak
Siswa 4	$6 \leq 11 \leq 16$	H_0 diterima dan teracak
Siswa 5	$6 \leq 8 \leq 16$	H_0 diterima dan teracak
Siswa 6	$6 \leq 11 \leq 16$	H_0 diterima dan teracak
Siswa 7	$6 \leq 14 \leq 16$	H_0 diterima dan teracak
Siswa 8	$6 \leq 10 \leq 16$	H_0 diterima dan teracak
Siswa 9	$6 \leq 14 \leq 16$	H_0 diterima dan teracak
Siswa 10	$6 \leq 7 \leq 16$	H_0 diterima dan teracak

Berdasarkan pengujian diatas menunjukkan 10 siswa hasil dari H_0 adalah diterima. Dibuktikan dengan nilai 10 siswa dapat memenuhi syarat *run test* yaitu $r_a \leq r \leq r_b$ dalam arti 10 siswa mendapatkan 20 nomor soal acak. Pengacakan pada saat menentuka *id* soal yang akan di dapatkan oleh siswa menggunakan metode *Multiply With Carry Generator* (MWCG) dengan membangkitkan soal-soal yang ada.

Dari hasil pengujian tersebut menunjukkan bahwa metode *Multiply With Carry Generator* (MWCG) layak untuk digunakan dalam metode pengacakan dengan memperhatikan beberapa ketentuan dari metode *Multiply With Carry Generator* (MWCG) itu sendiri dan tergantung pada penerapannya, apakah memerlukan pembangkit bilangan acak yang seragam atau tidak.

4. Kesimpulan

Setelah melakukan penelitian mengenai Implementasi Algoritma *Multiply With Carry Generator* (MWCG) Dalam Pengacakan Soal Ujian Semester Berbasis *Web* Pada SMKN 1 Kendari, maka diperoleh kesimpulan sebagai berikut:

1. Algoritma *Multiply With Carry Generator* (MWCG) dapat diterapkan dalam pengacakan soal ujian semester berbasis *website*.
2. Dari pengujian *Run Test* yang dilakukan untuk menguji keacakan dari Algoritma *Multiply With Carry Generator* (MWCG) pada 10 siswa dengan jumlah soal 20 nomor secara bersamaan menghasilkan bentuk soal acak, dengan hasil uji H_0 diterima. Hal tersebut membuktikan Algoritma *Multiply*

With Carry Generator dapat digunakan dalam pengacakan soal.

5. Saran

Adapun beberapa masukan yang perlu dilihat untuk pengembangan penelitian selanjutnya yaitu sebagai berikut:

1. Dalam penelitian selanjutnya dapat dikembangkan lagi dengan menambahkan lebih banyak lagi mata pelajaran dan dapat diterapkan bukan hanya untuk ujian semester.
2. Pengembangan Peneliti selanjutnya dapat menggunakan algoritma pembangkit bilangan acak lainnya, selain Algoritma *Multiply With Carry Generator* (MWCG).

References

- Arizqia, M. G., & Widodo, A. A. (2017). Rancang Bangun Aplikasi Dengan Linear Congruent Method (LCM) Sebagai Pengacakan Soal. *JOINTECS (Journal of Information Technology and Computer Science)*, 2(1). <https://doi.org/10.31328/jointecs.v2i1.412>
- Constantianus, F., & Suteja, B. (2005). Analisa dan Desain Sistem Bimbingan Tugas Akhir Berbasis Web dengan Studi Kasus Fakultas Teknologi Informasi. *Jurnal Informatika*, 1(2), 93–106.
- Hartono, H. (2013). Pengertian Website dan Unsur-Usurnya. *Ilmu Teknologi Informasi (Ilmuti)*, 1–7.
- Marsaglia, G. (2003). Random number generators. *Journal of Modern Applied Statistical Methods*, 2(1), 2–13. <https://doi.org/10.22237/jmasm/1051747320>
- Mulya, Y. P., Latuconsina, R., Siswo, A., Ansori, R., Teknik, F., Telkom, U., Teknik, F., Telkom, U., Teknik, F., & Telkom, U. (2020). Implementasi Algoritma *Multiply With Carry* Pada Game Panahan Algorithm Implementation Of *Multiply With Carry On*. 7(1), 1464–1475.
- Pramantya, Q. B. (2017). Analisis Kapabilitas Proses Pembuatan Batu Gamping Di CV. Anugerah Abadi Tuban. 62. <http://repository.its.ac.id/41974/>
- Pratomo, A., & Mantala, R. (2016). Pengembangan Aplikasi Ujian Berbasis Komputer beserta Analisis Uji Guna Sistem Perangkat Lunaknya Menggunakan Metode SUMI (Software Usability Measurement Inventory). *Jurnal POSITIF*, 2(1), 1–11.
- Rivaldo, R., & Herriyance, H. (2021). Pengamanan

- Pesan Menggunakan Metode MLSB PRNG dan Kompresi File dengan Algoritma RLE pada File Audio. *Jurnal Sistem Informasi Bisnis*, 01, 1–8.
<https://doi.org/10.21456/vol11iss1pp1-8>
- Salesiano *et al.*, “Implementasi Aplikasi Ujian Online Pada SMK PGRI 4 Kendal Menggunakan Framework Codeigniter” *J. Chem. Inf. Model.*, vol. 53, no. 9, p. 6, 2021, [Online]. Available: <http://dspace.ucuenca.edu.ec/bitstream/123456789/35612/1/TrabajodeTitulacion.pdf%0Ahttps://educacion.gob.ec/wpcontent/uploads/downloads/2019/01/GUI-A-METODOLOGICA-EF.pdf>.
- Samsuriati, N., Saputra, R. A., Pramono, B., Informatika, J. T., Teknik, F., Oleo, U. H., & Belakang, L. (2019). *Aplikasi Ujian Toefl Menggunakan Metode Linear Congruential Generators (Lcg) Berbasis Web Pada*. 233–242.
- T. J. C. Sitanggang, T. L. (2016). Aplikasi Pengacak Soal Ujian Berkategori Menggunakan Metode LCM (Linear Congruent Method). *MEANS (Media Informasi Analisa Dan Sistem)*, December 2016.