

# Modifikasi Algoritma Sattolo Shuffle Untuk Mengacak Soal Pada Aplikasi Ujian Online

Surya Darma Nasution\*, Mesran Mesran

Program Studi Manajemen, Sekolah Tinggi Ilmu Manajemen Sukma Medan, Medan, Indonesia

Email: <sup>1,\*</sup>surya.darma.nasution1@gmail.com, <sup>2</sup>mesran.skom.mkom@gmail.com

Email Penulis Korespondensi: surya.darma.nasution1@gmail.com

Submitted: 13/06/2025; Accepted: 30/06/2025; Published: 30/06/2025

**Abstrak**—Penggunaan sistem *Computer-Based Test (CBT)* telah menjadi metode evaluasi yang populer karena efisiensi dan kemampuannya dalam mempercepat proses penilaian. Namun, tantangan seperti kecurangan dan kemiripan susunan soal antar peserta masih sering terjadi. Penelitian ini bertujuan untuk merancang dan mengimplementasikan algoritma modifikasi *Sattolo Shuffle* dengan tambahan *Linear Congruential Generator (LCG)* sebagai sumber bilangan acak dalam proses pengacakan soal ujian. Algoritma *Sattolo Shuffle* dipilih karena menghasilkan permutasi siklis tunggal yang memastikan setiap elemen soal berpindah tempat, mengurangi potensi pola berulang. LCG digunakan untuk menghasilkan indeks pengacakan secara deterministik namun bervariasi berdasarkan parameter dan nilai awal (seed), yang diambil dari nomor urut peserta. Implementasi dilakukan dalam sistem CBT berbasis web dengan total 50 soal dalam setiap sesi ujian. Hasil pengujian pada tiga peserta menunjukkan bahwa susunan soal yang dihasilkan sepenuhnya berbeda dan tidak ditemukan urutan yang identik. Setiap peserta menerima kombinasi soal yang unik dengan distribusi posisi soal yang merata. Hasil awal menunjukkan efektivitas algoritma dalam meningkatkan variasi soal dan mencegah duplikasi, menjadikannya solusi yang potensial untuk memperkuat keamanan dan keadilan dalam pelaksanaan CBT. Penelitian ini diharapkan dapat menjadi kontribusi nyata dalam pengembangan sistem ujian daring yang lebih acak, adil, dan tahan terhadap potensi kecurangan.

**Kata Kunci:** Aplikasi Ujian Online; Sattolo Shuffle; Linear Congruential Generator; Pengacakan Soal; Modifikasi Algoritma.

**Abstract**—The use of Computer-Based Testing (CBT) systems has become a popular evaluation method due to its efficiency and ability to accelerate the assessment process. However, challenges such as cheating and the similarity of question sequences among participants still frequently occur. This study aims to design and implement a modified Sattolo Shuffle algorithm with the addition of a Linear Congruential Generator (LCG) as a source of random numbers in the exam question randomization process. The Sattolo Shuffle algorithm was chosen because it produces a single cyclic permutation that ensures each question element is repositioned, reducing the potential for recurring patterns. The LCG is used to generate random indices deterministically but variably, based on specific parameters and an initial value (seed) derived from the participant's serial number. The implementation was carried out in a web-based CBT system consisting of 50 questions in each exam session. Testing on three participants showed that the generated question sequences were completely different, with no identical orders found. Each participant received a unique combination of questions with an even distribution of question positions. Initial results demonstrate the algorithm's effectiveness in increasing question variation and preventing duplication, making it a potential solution to enhance security and fairness in CBT administration. This research is expected to contribute significantly to the development of more randomized, fair, and cheat-resistant online exam systems.

**Keywords:** Online Exam Application; Sattolo Shuffle; Linear Congruential Generator; Question Randomization; Algorithm Modification.

## 1. PENDAHULUAN

Perkembangan teknologi informasi mendorong transformasi signifikan dalam sistem evaluasi pembelajaran, khususnya melalui aplikasi ujian online, yang biasanya lebih dikenal dengan istilah *Computer-Based Test (CBT)*. CBT menjadi pilihan populer karena mampu mengurangi biaya logistik, mempercepat penilaian, dan meningkatkan aksesibilitas pelaksanaan ujian daring [1], [2]. Dalam penerapannya, keadilan dan integritas soal ujian menjadi perhatian utama, yang salah satunya dapat dicapai melalui pengacakan soal [3]. Pengacakan soal bertujuan agar setiap peserta mendapat susunan berbeda, meminimalisir potensi kecurangan seperti saling mencontek [4]. Salah satu algoritma yang banyak digunakan adalah *Fisher-Yates Shuffle*. Juniawan dan Hengki [5] menunjukkan bahwa algoritma ini efektif menekan kecurangan dalam seleksi masuk karena setiap peserta mendapat urutan soal berbeda. Alamsyah et al [6], juga mencatat bahwa Fisher-Yates dapat menjaga kestabilan distribusi meski diuji sebanyak 40 kali pada 30 soal tanpa menghasilkan susunan yang sama.

Penelitian Febriani et al [7], mengungkapkan bahwa Fisher-Yates memungkinkan pengacakan soal matematika tanpa pengulangan, dengan waktu akses cepat dan efisiensi tinggi, menghasilkan peluang acak setara antar peserta. Prakarsa et al [8], menggabungkan *Linear Congruential Generator (LCG)* dengan Fisher-Yates untuk ujian online tingkat SMA, dan berhasil mencegah kecurangan serta menghasilkan susunan soal yang berbeda antar siswa. Keunggulan *Fisher-Yates* terletak pada kemampuannya menghasilkan permutasi acak yang adil dengan kompleksitas waktu  $O(n)$ , khususnya pada versi Durstenfeld [9]. Abdullah [10] menunjukkan bahwa peluang dua siswa memperoleh urutan soal sama dari 15 soal hanya 0,00832%, atau 1 dari 360.360 kombinasi, sehingga sangat efektif dalam menekan kecurangan. Pohan et al [11], juga membuktikan efisiensi Knuth Shuffle varian Fisher-Yates dalam aplikasi ujian online di SMK yang menghasilkan variasi soal tanpa repetisi dalam milidetik.

Namun, *Fisher-Yates* memiliki keterbatasan. Nasution dan Suginam [12] mencatat bahwa algoritma ini memerlukan generator bilangan acak eksternal dan tidak dapat berdiri sendiri. Dalam kasus LCM, Gunawan dan

Prabowo [13] menekankan bahwa pemilihan konstanta sangat memengaruhi kualitas bilangan acak, dan kesalahan dalam pemilihan ini dapat menghasilkan pola yang dapat ditebak. Prakarsa et al [8], juga menyarankan kombinasi metode untuk menghindari bias distribusi akibat operator modulo. Beberapa peneliti mencoba mengevaluasi dan menggabungkan algoritma untuk mengatasi keterbatasan tersebut. Izzah dan Mufarrihah [14] mengembangkan sistem CBT berbasis web dengan *Fisher-Yates Shuffle* untuk menciptakan distribusi soal yang lebih bervariasi dan efisien. Risnasari et al [4], mengintegrasikan Fisher-Yates dengan algoritma Smith-Waterman untuk penilaian esai otomatis. Suprpto et al [15], menerapkan algoritma ini dalam aplikasi CBT Android, menunjukkan hasil konsisten dalam mencegah duplikasi soal dan meningkatkan efisiensi.

Syalsabella dan Prapanca [16] menekankan pentingnya pengacakan dalam ujian seperti UTBK, dan membuktikan bahwa *Fisher-Yates Shuffle* efektif dalam menjaga keamanan soal serta mencegah kemiripan antar peserta. Ariandi dan Ariyadi [17] menyatakan bahwa meskipun algoritma ini efektif dalam aplikasi edukasi anak usia dini, pola tetap bisa muncul bila implementasinya tidak optimal. Hal serupa dicatat oleh Haditama et al [18], yang menyebutkan bahwa *Knuth Shuffle* mampu menghasilkan distribusi acak tanpa pengulangan, tetapi dalam konteks terbatas seperti game kuis berbasis Android, pola masih bisa terlihat.

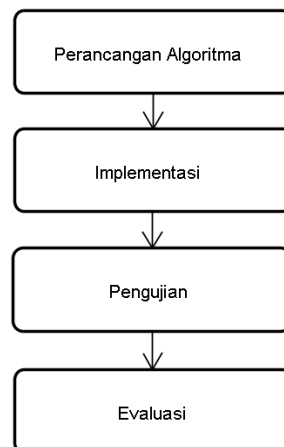
Dalam konteks ini, *Sattolo Shuffle* muncul sebagai alternatif menarik. Algoritma ini menghasilkan permutasi siklis tunggal, yakni setiap elemen dalam himpunan data pasti berpindah posisi dalam satu siklus tertutup. Konsep ini diperkenalkan oleh Sandra Sattolo pada tahun 1986 [19], dan telah digunakan dalam berbagai konteks seperti pengacakan memori dalam game edukatif [20]. Properti dasar ini membuat *Sattolo Shuffle* berpotensi unggul dalam menjaga keragaman urutan soal. Namun, sejauh ini kajian empiris mengenai penerapannya dalam sistem CBT masih sangat terbatas. Studi pustaka menunjukkan bahwa sebagian besar penelitian fokus pada *Fisher-Yates* dan variannya, baik versi standar maupun kombinasi dengan algoritma lain. Masih sedikit bahkan belum ada penelitian yang secara khusus mengeksplorasi penerapan *Sattolo Shuffle* dalam pengacakan soal CBT. Hal ini menunjukkan adanya gap penelitian yang signifikan dan layak dieksplorasi.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan versi modifikasi dari algoritma *Sattolo Shuffle* dalam sistem CBT. Modifikasi dilakukan dengan menambahkan LCG sebagai generator bilangan acak untuk proses shuffle, mengikuti pendekatan Prakarsa et al [8]. Dengan pendekatan ini, penelitian diharapkan memberikan kontribusi praktis dalam pengembangan sistem CBT yang lebih adil, variatif, dan tahan terhadap potensi kecurangan.

## 2. METODOLOGI PENELITIAN

### 2.1 Tahapan Penelitian

Tahapan penelitian yang dilaksanakan pada penelitian ini dapat dilihat pada Gambar 1 berikut ini.



**Gambar 1.** Tahapan Penelitian

Dapat dilihat pada Gambar 1, bahwa penelitian dilaksanakan melalui beberapa tahapan yaitu:

#### a. Perancangan Algoritma

1. Mendesain algoritma *Sattolo Shuffle* untuk proses mengacak urutan soal.
2. Mengintegrasikan *Linear Congruential Generator (LCG)* sebagai pembangkit bilangan acak untuk proses shuffle.
3. Menentukan parameter LCG yang sesuai untuk menjamin kualitas kerandoman. Parameter yang digunakan dalam penelitian ini adalah:
  - a)  $a$  (konstanta pengali) = 7
  - b)  $c$  (konstanta penambah) = 49
  - c)  $m$  (modulus) = nilai variabel  $i$  yang menurun pada tiap iterasi
  - d)  $X_i$  (seed) = nomor urut peserta

Pemilihan ini bersifat eksperimental. Namun, karena m berubah-ubah (mengikuti i), kualitas keacakannya masih perlu diuji secara lebih lanjut.

- b. Implementasi
  1. Algoritma diimplementasikan ke dalam sistem simulasi CBT berbasis web.
  2. Soal ujian diinput di *database* dan akan diambil untuk ditampilkan sebagai array, kemudian diacak menggunakan algoritma hasil rancangan.
- c. Pengujian
  1. Dilakukan simulasi pengacakan soal ujian.
  2. Setiap hasil pengacakan dicatat untuk:
    - a) Dilakukan simulasi pengacakan sebanyak  $N = 3$  peserta ujian untuk nomor urut peserta yang berdekatan. Dari  $N$  hasil tersebut, dihitung berapa kombinasi urutan soal yang unik dengan membandingkan masing-masing urutan secara langsung dan mengukur persentase duplikasi.
    - b) Untuk mengukur distribusi posisi, dilakukan analisis frekuensi kemunculan masing-masing soal pada setiap posisi (1–50) dari hasil  $N$  simulasi. Distribusi yang ideal akan menunjukkan penyebaran yang mendekati seragam di seluruh posisi.
    - c) Deteksi dilakukan dengan membandingkan seluruh urutan soal dalam bentuk string antar peserta atau antar simulasi. Jika ditemukan urutan identik, dianggap sebagai pola berulang.
- d. Evaluasi
  1. Menilai kerandoman dan keragaman urutan soal.
  2. Mengidentifikasi kemungkinan duplikasi urutan antar simulasi.

## 2.2 Algoritma Sattolo Shuffle

Algoritma *Sattolo Shuffle* merupakan pengembangan dari algoritma Fisher-Yates Shuffle yang diperkenalkan oleh Sandra Sattolo pada tahun 1986. Tujuan utama dari algoritma ini adalah untuk menghasilkan permutasi acak berbentuk siklus tunggal dari suatu himpunan data berhingga, sehingga setiap elemen dalam himpunan pasti berpindah posisi. Berbeda dari Fisher-Yates yang memungkinkan elemen tetap pada posisi semula, *Sattolo Shuffle* memastikan tidak ada elemen yang tetap dan seluruh hasil pengacakan memiliki peluang yang merata, sehingga tidak menimbulkan bias dalam distribusi [21], [22], [23].

Berikut ini adalah tahapan dalam algoritma *Sattolo Shuffle*:

- a. Pemasukan Data:  
Kumpulan data yang akan diacak dimasukkan ke dalam array.
  - b. Penentuan Panjang:  
Panjang array disimpan ke dalam variabel  $i$ .
  - c. Pemilihan Indeks Acak:  
Pilih angka acak  $r$  dalam rentang antara 0 hingga  $i-2$ .
  - d. Pertukaran Elemen:  
Elemen pada indeks  $r$  ditukar dengan elemen pada indeks  $i-1$ .
  - e. Pengurangan Indeks:  
Kurangi nilai  $i$  sebanyak satu. Jika nilai  $i$  masih lebih besar dari 1, ulangi proses dari langkah ketiga.
- Proses ini berjalan secara iteratif hingga seluruh elemen berpindah tempat. Keunggulan utama algoritma ini terletak pada kesederhanaan proses dan efisiensi penggunaan sumber daya, menjadikannya pilihan yang efektif dalam aplikasi yang memerlukan hasil acak yang tidak berulang [22], [23], [24].

## 2.3 Linear Congruential Generator

Linear Congruential Generator (LCG), yang juga kerap disebut sebagai Linear Congruential Method (LCM), merupakan salah satu teknik klasik dalam menghasilkan bilangan acak semu dan masih banyak digunakan dalam berbagai aplikasi pemrograman. Metode ini memanfaatkan rumus rekursif berbasis aritmatika modular untuk menghasilkan urutan bilangan yang tampak acak, meskipun bersifat deterministik. Salah satu karakteristik utama dari metode ini adalah adanya siklus atau periode tertentu di mana deret bilangan akan mulai mengulang, tergantung pada parameter yang digunakan [12]. Rumus dasar LCG dapat dituliskan sebagai berikut:

$$X(i + 1) = (a * X_i + c) \text{ Mod } m \quad (1)$$

Dengan penjelasan sebagai berikut:

- a.  $X_{i+1}$  = Bilangan acak yang baru dihasilkan
- b.  $X_i$  = Bilangan acak sebelumnya (atau nilai awal yang disebut *seed*)
- c.  $a$  = Konstanta pengali (*multiplier*)
- d.  $c$  = Konstanta penambah (*increment*)
- e.  $m$  = Modulus, yaitu batas maksimum nilai bilangan acak

Kualitas dari deret bilangan acak yang dihasilkan sangat ditentukan oleh pemilihan parameter  $a$ ,  $c$ , dan  $m$ . Jika parameter tidak dipilih dengan tepat, LCG dapat menghasilkan pola yang mudah dikenali atau siklus yang



terlalu pendek, sehingga mengurangi kualitas keacakannya. Sebaliknya, dengan konfigurasi parameter yang sesuai, metode ini dapat menghasilkan urutan acak dengan periode panjang dan distribusi yang cukup merata.

Dalam implementasi sistem seperti pengacakan soal ujian berbasis komputer, LCG sering digunakan sebagai pembangkit bilangan acak yang menjadi dasar untuk proses pengacakan algoritmik, seperti *Sattolo Shuffle*. Karena efisiensinya yang tinggi dan kemudahan implementasi, LCG sangat cocok untuk diterapkan dalam sistem real-time atau berbasis web, di mana respons cepat dan kestabilan hasil sangat dibutuhkan.

### 3. HASIL DAN PEMBAHASAN

Penerapan algoritma *Sattolo Shuffle* yang telah dimodifikasi dilakukan pada sebuah aplikasi Ujian Online (Computer-Based Test / CBT). Tujuan utama implementasi ini adalah untuk menghasilkan susunan soal yang berbeda pada setiap peserta ujian, meskipun mereka mengikuti ujian pada waktu yang sama dan berada dalam posisi duduk yang berdekatan, seperti berdampingan atau bersebelahan secara vertikal maupun horizontal.

Pengacakan soal dengan *Sattolo Shuffle* menjamin bahwa setiap elemen (soal) dalam daftar selalu berpindah posisi dalam satu siklus tertutup, yang membuat algoritma ini memiliki keunggulan dibanding Fisher-Yates dalam hal memastikan tidak ada soal yang tetap pada posisinya semula. Untuk memperkuat keacakannya, algoritma ini dimodifikasi dengan penambahan *Linear Congruential Generator (LCG)* sebagai pembangkit bilangan acak, yang digunakan dalam proses penentuan indeks pengacakan.

Secara teknis, algoritma bekerja dengan memuat daftar soal ke dalam sebuah array. Kemudian, secara iteratif, elemen dalam array diacak menggunakan bilangan acak hasil dari LCG. Pemilihan nilai awal (seed) dan parameter LCG yang tepat memungkinkan sistem menghasilkan pola acak yang berbeda untuk setiap pengguna, bahkan jika mereka memulai ujian secara bersamaan. Hasil pengacakan urutan soal ini langsung diproses di sisi server dan dikirim secara personal ke klien pengguna saat sesi ujian dimulai.

#### 3.1 Modifikasi Algoritma *Sattolo Shuffle*

Modifikasi algoritma *Sattolo Shuffle* dilakukan dengan menambahkan rumus LCG pada langkah ketiga yaitu pada pemilihan indeks acak. Berikut ini adalah tahapan dalam algoritma *Sattolo Shuffle* yang telah dimodifikasi:

- a. Pemasukan Data:  
Kumpulan data yang akan diacak dimasukkan ke dalam array, pada penelitian ini jumlah soal yang akan diacak urutannya sebanyak 50 soal, yaitu array 0 sampai 49.
- b. Penentuan Panjang:  
Panjang array disimpan ke dalam variabel  $i$ , sehingga  $i=49$ .
- c. Pemilihan Indeks Acak:
  1. Tentukan diawal nilai variabel  $a, c, m$  dan  $X_i$  untuk proses LCG.
  2. Ketentuan nilai variabel tersebut pada penelitian ini diambil bahwa  $a = 7$ ,  $c = 49$ ,  $m =$  nilai variabel  $i$  yang berubah setiap iterasinya, serta nilai  $X_i =$  diambil dari no urut peserta ujian.
  3. Gunakan rumus LCG pada persamaan (1).
- d. Pertukaran Elemen:  
Elemen pada indeks  $r$  ditukar dengan elemen pada indeks  $i-1$ .
- e. Pengurangan Indeks:  
Kurangi nilai  $i$  sebanyak satu. Jika nilai  $i$  masih lebih besar dari 1, ulangi proses dari langkah ketiga.

Uji coba perhitungan dilakukan sebagai contoh kepada 3 peserta ujian dengan nomor urut 1,2, dan 3, dengan jumlah soal sebanyak 50 soal. Berikut adalah hasil bilangan acaknya:

Peserta 1:

Diketahui  $\rightarrow$  nilai  $i = 49$  (pada iterasi 1),

$$X_i=1, a=7, c=49, m=i$$

Hasil bilangan acak:

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 25 | 32 | 9  | 42 | 43 | 7  | 18 | 26 | 14 | 34 |
| 20 | 27 | 45 | 11 | 36 | 33 | 47 | 17 | 6  | 5  |
| 37 | 28 | 30 | 49 | 46 | 15 | 21 | 50 | 39 | 13 |
| 4  | 48 | 38 | 23 | 16 | 1  | 19 | 12 | 44 | 35 |
| 10 | 41 | 29 | 22 | 40 | 31 | 24 | 3  | 2  | 8  |

Peserta 2:

Diketahui  $\rightarrow$  nilai  $i = 49$  (pada iterasi 1),

$$X_i=2, a=7, c=49, m=i$$

Hasil bilangan acak:

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 25 | 32 | 9  | 34 | 43 | 46 | 18 | 26 | 42 | 14 |
| 20 | 47 | 40 | 11 | 36 | 33 | 48 | 17 | 6  | 5  |
| 37 | 44 | 30 | 2  | 31 | 50 | 21 | 8  | 39 | 13 |
| 45 | 3  | 38 | 23 | 16 | 1  | 19 | 12 | 22 | 35 |
| 10 | 41 | 29 | 28 | 49 | 7  | 26 | 24 | 4  | 15 |

Peserta 3:

Diketahui -> nilai  $i = 49$  (pada iterasi 1),

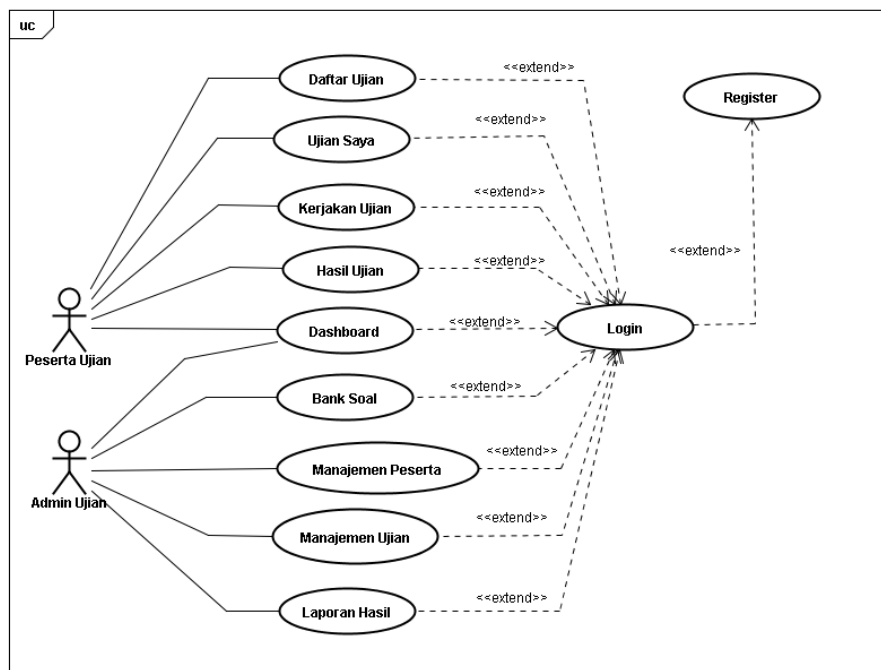
$X_i=3, a=7, c=49, m=i$

Hasil bilangan acak:

|    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|
| 25 | 32 | 9  | 42 | 43 | 7  | 18 | 26 | 14 | 34 |
| 20 | 46 | 40 | 11 | 36 | 33 | 24 | 17 | 6  | 49 |
| 37 | 28 | 47 | 2  | 48 | 15 | 21 | 8  | 39 | 13 |
| 44 | 3  | 38 | 23 | 16 | 1  | 45 | 12 | 50 | 35 |
| 10 | 41 | 29 | 4  | 19 | 27 | 30 | 31 | 5  | 22 |

### 3.2 Implementasi Hasil Modifikasi Algoritma *Sattolo Shuffle* Pada Aplikasi Ujian Online

Implementasi dari algoritma *Sattolo Shuffle* yang telah dimodifikasi pada aplikasi Ujian Online berhasil diteapkan dan hasil pembangkitan bilangan acaknya sesuai dengan perhitungan yang telah dilakukan. Use case diagram dari aplikasi ujian online ini dapat dilihat pada gambar 2 berikut ini.



Gambar 2. Use Case Aplikasi Ujian Online

Berikut beberapa tampilan dari aplikasi ujian online:

a. Halaman Admin Ujian

Halaman Admin Ujian ditujukan untuk pengelola ujian, dimana terdapat menu dashboard, manajemen peserta, bank soal, manajemen ujian dan melihat laporan hasil. Dashboard untuk melihat rekap semua informasi, Manajemen peserta untuk melihat peserta yang telah mendaftar, bank soal untuk melihat dan menambahkan bank soal dari setiap pelajaran. Manajemen ujian untuk membuat jadwal ujian. Laporan Hasil untuk melihat hasil ujian setiap sesinya. Halaman admin ujian dapat dilihat pada Gambar 3.

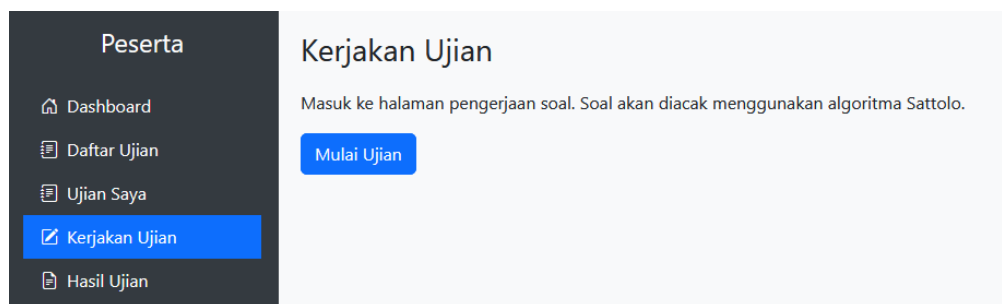


Gambar 3. Halaman Admin Ujian

b. Halaman Peserta

Halaman Peserta ditujukan untuk peserta ujian yang telah mendaftar dan telah login, dimana terdapat menu dashboard, daftar ujian, ujian saya, kerjakan ujian, dan hasil ujian. Dashboard untuk melihat rekap semua informasi, Daftar ujian untuk mendaftar ke ujian yang tersedia, Ujian Saya untuk melihat ujian yang telah diikuti

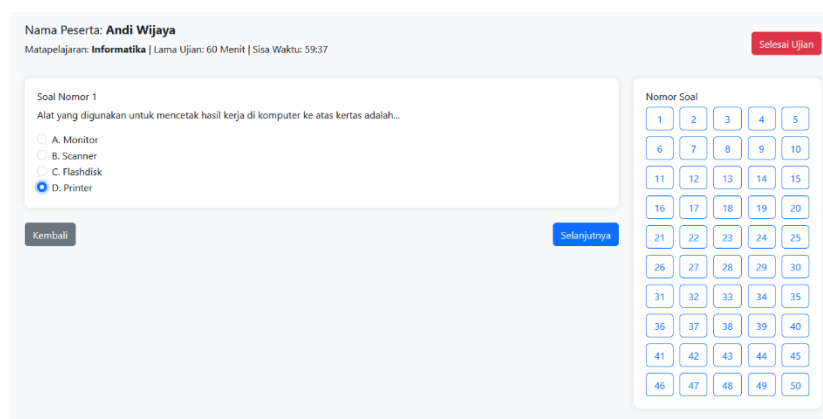
dan yang akan diikuti oleh peserta. Kerjakan Ujian untuk peserta melaksanakan ujian, dan hasil ujian untuk melihat hasil dari ujian. Halaman peserta dapat dilihat pada Gambar 4.



**Gambar 4.** HalamanPeserta

c. Halaman Ujian

Halaman ujian adalah halaman untuk peserta ujian melaksanakan ujian dengan masuk ke menu kerjakan ujian. Halaman ujian dapat dilihat pada Gambar 5.



**Gambar 5.** Halaman Ujian

**3.3 Pembahasan**

Hasil menunjukkan bahwa angka acak yang dihasilkan dari modifikasi algoritma *Sattolo Shuffle* tidak memenuhi harapan awal. Meskipun secara teoritis *Sattolo Shuffle* menjamin setiap elemen berpindah posisi dan LCG menghasilkan indeks yang bervariasi, pada praktiknya ditemukan pola urutan soal yang identik pada lebih dari separuh soal. Hal ini menunjukkan bahwa sistem belum mampu menghasilkan variasi yang cukup signifikan untuk mencegah potensi kecurangan. Terdapat angka yang sama, bahkan sampai 31 soal memiliki urutan tampil yang sama. Perbandingan nilai acak yang sama dapat dilihat pada Tabel 1.

**Tabel 1.** Perbandingan Nilai Acak Pada Setiap Peserta

| Perbandingan          | Jumlah Soal Yang Sama |
|-----------------------|-----------------------|
| Peserta 1 - Peserta 2 | 28                    |
| Peserta 2 – Peserta 3 | 29                    |
| Peserta 1 – Peserta 3 | 31                    |

Dari Tabel 1 dapat disimpulkan bahwa setiap peserta yang duduk berdekatan dapat melakukan kecurangan atau mencontek pada saat ujian, hal ini dikarenakan terdapat urutan soal yang sama. Fenomena ini kemungkinan disebabkan oleh beberapa faktor. Pertama, penggunaan nilai *m* yang berubah-ubah (mengikuti *i*) dalam LCG sangat tidak lazim dan kemungkinan besar menyebabkan distribusi bilangan acak yang tidak stabil. Umumnya, *m* disarankan berupa bilangan prima besar yang tetap. Kedua, jumlah peserta (tiga orang) masih terlalu kecil untuk menarik kesimpulan menyeluruh, meskipun tingkat kemiripan urutan hingga 31 soal merupakan indikasi kuat adanya pola. Ketiga, besar kemungkinan terdapat kesalahan pada integrasi antara LCG dan proses pemilihan indeks pada algoritma *Sattolo Shuffle*. Terakhir, penting untuk meninjau ulang pemahaman dan implementasi prinsip "permutasi siklis tunggal", karena jika benar diimplementasikan, seharusnya tidak ada dua peserta dengan urutan identik.

**4. KESIMPULAN**

Penelitian ini mengimplementasikan algoritma *Sattolo Shuffle* yang dimodifikasi dengan *Linear Congruential Generator (LCG)* untuk mengacak urutan soal dalam sistem Computer-Based Test (CBT) berbasis web. Tujuan utama

modifikasi ini adalah meningkatkan keadilan dan keamanan ujian dengan menghasilkan urutan soal yang berbeda bagi setiap peserta, sehingga potensi kecurangan dapat diminimalkan. Dari hasil implementasi dan pengujian terhadap tiga peserta dengan nomor urut berbeda, sistem berhasil menghasilkan variasi urutan soal. Namun, analisis lebih lanjut menunjukkan bahwa sebagian besar soal tetap muncul dalam urutan yang sama di antara peserta, dengan tingkat kemiripan mencapai 31 soal dari 50. Hal ini menunjukkan bahwa keacakan yang dihasilkan masih belum optimal, dan peserta yang duduk berdekatan tetap memiliki peluang untuk melakukan kecurangan. Dengan demikian, dapat disimpulkan bahwa meskipun pendekatan ini memberikan dasar pengacakan soal berbasis identitas peserta, modifikasi yang dilakukan belum cukup efektif dalam memastikan keunikan susunan soal secara menyeluruh. Ke depan, penggunaan pembangkit bilangan acak yang lebih kuat, seperti Mersenne Twister atau penggunaan kombinasi LCG dengan nilai parameter yang telah teruji secara matematis, sangat disarankan. Selain itu, verifikasi ulang terhadap integrasi algoritma *Sattolo Shuffle* juga diperlukan untuk memastikan bahwa permutasi yang dihasilkan benar-benar membentuk siklus tunggal dan bebas dari pola yang berulang. Penelitian ini menjadi langkah awal dalam pemanfaatan algoritma *Sattolo Shuffle* dalam konteks CBT dan membuka peluang pengembangan lebih lanjut dalam bidang ini.

## REFERENCES

- [1] M. A. Hasan, Supriadi, and Zamzami, "Implementasi Algoritma Fisher-Yates Untuk Mengacak Soal Ujian Online Penerimaan Mahasiswa Baru (Studi Kasus : Universitas Lancang Kuning Riau)," *Jurnal Teknologi dan Sistem Informasi*, vol. 3, no. 2, 2017, doi: 10.25077/TEKNOSI.v3i2.2017.291-298.
- [2] K. Andesa, N. Nurjayadi, H. Herwin, and T. Nasution, "Sistem Ujian Online Menggunakan Algoritma Fisher Yates Shuffle," *SATIN - Sains dan Teknologi Informasi*, vol. 6, no. 2, pp. 67–74, Dec. 2020, doi: 10.33372/stn.v6i2.669.
- [3] T. Sugihartono and R. R. C. Putra, "Penerapan Algoritma Fisher Yates untuk Pengacakan Soal Pada Sistem Ujian Kompetensi Wartawan," *Infotek : Jurnal Informatika dan Teknologi*, vol. 4, no. 2, pp. 238–248, 2021.
- [4] M. Risnasari, M. A. Effindi, P. Dellia, L. Cahyani, and N. Aini, "Computer Based Test Using the Fisher-Yates Shuffle and Smith Waterman Algorithm," in *International Conference on Art, Design, Education and Cultural Studies (ICADECS)*, Kne Social Sciences, Jun. 2021, pp. 353–360. doi: 10.18502/kss.v5i6.9224.
- [5] F. P. Juniawan and H. Hengki, "Pengacakan Soal Ujian Penerimaan Polri Menggunakan Algoritme Fisher Yates Shuffle," *Jurnal Telematika*, vol. 1, no. 1, 2019.
- [6] A. Alamsyah, I. M. Shofi, and H. B. Suseno, "Prototipe Sistem Computer Based Test dengan Pengacakan Soal Menggunakan Metode Fisher-Yates Shuffle," *Jurnal Ilmu Komputer Agri-Informatika*, vol. 8, no. 2, pp. 81–89, 2021.
- [7] I. Febriani, R. Ekawati, U. Supriadi, and M. I. Abdullah, "Fisher-Yates shuffle algorithm for randomization math exam on computer based-test," in *AIP Conference Proceedings*, American Institute of Physics Inc., Apr. 2021. doi: 10.1063/5.0042534.
- [8] A. Prakarsa, A. A. Sunarto, and P. Prajoko, "Model Pengacakan Soal Ujian Online SMA Menggunakan Metode Linear Congruential Generator dan Fisher Yates," *Progresif: Jurnal Ilmiah Komputer*, vol. 16, no. 2, pp. 133–142, 2020.
- [9] A. Suhazli, A. Atthariq, and A. Anwar, "Game Puzzle 'Numbers in English' Berbasis Android Dengan Metode Fisher Yates Shuffle Sebagai Pengacak Potongan Gambar," *Jurnal Infomedia*, vol. 2, no. 1, pp. 1–6, 2017.
- [10] N. F. Abdullah, "Pengembangan dan Pengujian Aplikasi Computer Based Test dengan Menggunakan Algoritme Fisher-Yates Shuffle sebagai Pengacakan Soal," *Journal of Computer Science an Engineering (JCSE)*, vol. 4, no. 1, pp. 50–60, 2023, doi: 10.36596/jese.v4i1.695.
- [11] M. I. Pohan, E. Hariyanto, and F. Izhari, "Pengacakan Soal Ujian Online Berbasis Web Menggunakan Algoritma Knuth Shuffle Pada SMK Gelora Jaya Nusantara Medan," *Bulletin of Information Technology (BIT)*, vol. 3, no. 4, pp. 294–300, 2022, doi: 10.47065/bit.v3i1.
- [12] S. D. Nasution and Suginam, "Modifikasi Algoritma Fisher Yates Shuffle Menggunakan Linear Congruent Method Untuk Pembangkitan Bilangan Acak," *Jurnal Ilmu Komputer*, vol. XII, no. 2, pp. 101–106, 2019.
- [13] G. Gunawan and D. Agung Prabowo, "Sistem Ujian Online Seleksi Penerimaan Mahasiswa Baru Dengan Pengacakan Soal Menggunakan Linear Congruent Method (Studi Kasus di Universitas Muhammadiyah Bengkulu)," *JURNAL INFORMATIKA UPGRIS*, vol. 3, no. 2, 2017.
- [14] A. Izzah and I. Mufarrihah, "Implementasi Metode Fisher Yates Shuffle Pada Computer Based Test Online Berbasis Website," *Inovate : Jurnal Ilmiah Inovasi Teknologi Informasi*, vol. 8, no. 1, pp. 83–90, 2023.
- [15] D. D. A. Suprpto, "Aplikasi Smart Entrance Exam Berbasis Android dengan Algoritma Fisher-Yates," *Jurnal Sistem dan Teknologi Informasi (Justin)*, vol. 9, no. 1, pp. 25–32, 2020, doi: 10.26418/justin.v9i1.42416.
- [16] A. P. Syalsabella and A. Prapanca, "Perancangan Website Online Course (KitaPTN) untuk Persiapan UTBK Menggunakan Algoritma Fisher-Yates Shuffle," *Journal of Informatics and Computer Science*, vol. 06, no. 02, pp. 581–588, 2024.
- [17] M. Ariandi and M. D. Ariyadi, "Penerapan Algoritma Fisher Yates Shuffle pada Game Edukasi Pembelajaran Untuk Pendidikan Anak Usia Dini (PAUD)," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 4, 2022, doi: 10.30865/mib.v6i4.4683.
- [18] I. Haditama, C. Slamet, and D. F. Rahman, "Implementasi Algoritma Fisher-Yates Dan Fuzzy Tsukamoto Dalam Game Kuis Tebak Nada Sunda Berbasis Android," *JOIN*, vol. 1, no. 1, pp. 51–58, 2016.
- [19] S. C. Santo and N. M. S. Iswari, "Design and Development of Animal Recognition Application Using Gamification and *Sattolo Shuffle* Algorithm on Android Platform Case Study: Kebun Binatang Ragunan," *46 IJNMT*, vol. IV, no. 1, pp. 46–53, 2017.
- [20] S. D. Nasution, "Penerapan Algoritma *Sattolo Shuffle* Pada Memory Matching Game," *KETIK : Jurnal Informatika*, vol. 02, no. 03, pp. 01–06, 2025.
- [21] S. C. Santo and N. M. S. Iswari, "Design and Development of Animal Recognition Application Using Gamification and *Sattolo Shuffle* Algorithm on Android Platform Case Study: Kebun Binatang Ragunan," *46 IJNMT*, vol. IV, no. 1, pp. 46–53, 2017.



- [22] M. Yohanna, F. G. N. Larosa, and D. F. Malau, “Aplikasi Ujian Try Out Dengan Menerapkan Algoritma Fisher Yates Shuffle,” *Jurnal Teknik Informatika*, vol. 14, no. 2, 2022.
- [23] S. D. Nasution, “Penerapan Algoritma *Sattolo Shuffle* Pada Memory Matching Game,” *KETIK : Jurnal Informatika*, vol. 02, no. 03, pp. 01–06, 2025.
- [24] Y. Arviansyah, N. Nurfaizah, and R. Waluyo, “Penerapan Algoritma Fisher Yates Shuffle Pada Aplikasi TOEFL Preparation Berbasis Web,” *Jurnal Buana Informatika*, vol. 11, no. 2, pp. 112–122, 2020.