

Penerapan Algoritma Sattolo Shuffle dalam Sistem Pengacakan Denah Tempat Duduk Peserta Ujian untuk Mencegah Kecurangan Akademik

Surya Darma Nasution*, Guidio Leonarde Ginting

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

Email: ^{1,*}darmashadow@gmail.com, ²guidio.leonard626@gmail.com

Email Penulis Korespondensi: darmashadow@gmail.com

Abstrak—Kecurangan akademik dalam pelaksanaan ujian masih menjadi permasalahan serius di institusi pendidikan. Salah satu faktor yang memfasilitasi kecurangan adalah pengaturan tempat duduk yang tetap dan dapat diprediksi, sehingga peserta ujian dapat merencanakan strategi mencontek dengan peserta yang duduk berdekatan. Penelitian ini bertujuan untuk menerapkan algoritma Sattolo Shuffle dalam sistem pengacakan denah tempat duduk peserta ujian sebagai upaya pencegahan kecurangan akademik. Algoritma Sattolo Shuffle dipilih karena memiliki keunggulan dalam menghasilkan permutasi siklis (cyclic permutation) yang menjamin bahwa setiap elemen berpindah dari posisi asalnya, sehingga tidak ada peserta ujian yang menempati posisi duduk yang sama dengan ujian sebelumnya. Penelitian dilakukan dengan pendekatan eksperimental terhadap 30 peserta ujian yang diacak menggunakan algoritma Sattolo Shuffle. Proses pengacakan membutuhkan 29 iterasi ($n-1$ iterasi untuk $n=30$ elemen), dan hasil verifikasi menunjukkan bahwa tidak ada satupun peserta yang tetap berada di posisi asalnya (0 fixed point), membuktikan keberhasilan algoritma dalam menghasilkan permutasi siklis yang sempurna. Seluruh 30 peserta membentuk satu siklus tunggal, mengkonfirmasi sifat derangement dari algoritma Sattolo Shuffle.

Kata Kunci: Algoritma Sattolo Shuffle; Pengacakan Tempat Duduk; Kecurangan Akademik; Permutasi Siklis

Abstract—Academic cheating during examinations remains a serious problem in educational institutions. One contributing factor is the use of fixed and predictable seating arrangements, which allow exam participants to plan cheating strategies with nearby peers. This study aims to implement the Sattolo Shuffle algorithm in a seating arrangement randomization system for exam participants as a measure to prevent academic dishonesty. The Sattolo Shuffle algorithm was chosen because of its ability to generate cyclic permutations that guarantee every element moves from its original position, ensuring no exam participant occupies the same seat as in a previous exam. The study was conducted using an experimental approach involving 30 exam participants randomized using the Sattolo Shuffle algorithm. The shuffling process requires 29 iterations ($n-1$ iterations for $n=30$ elements), and the verification results show that no participant remains in their original position (0 fixed points), proving the algorithm successfully produces a perfect cyclic permutation. All 30 participants form a single cycle, confirming the derangement property of the Sattolo Shuffle algorithm.

Keywords: Sattolo Shuffle Algorithm; Seating Randomization; Academic Cheating; Cyclic Permutation

1. PENDAHULUAN

Kecurangan akademik merupakan permasalahan yang telah lama dihadapi oleh institusi pendidikan di seluruh dunia. Berbagai penelitian menunjukkan bahwa prevalensi kecurangan akademik tetap tinggi meskipun berbagai upaya pencegahan telah dilakukan [1], [2]. Di tingkat perguruan tinggi, sekitar 50-70% mahasiswa mengakui pernah melakukan kecurangan setidaknya sekali selama masa studi mereka [3]. Bentuk kecurangan yang paling umum dalam ujian tatap muka adalah mencontek atau menyalin jawaban dari peserta yang duduk berdekatan, yang difasilitasi oleh berbagai metode kecurangan dan minimnya pencegahan di kelas [4]. Penelitian Topîrceanu menunjukkan bahwa jaringan pertemanan antar peserta ujian berpengaruh signifikan terhadap kecurangan, sehingga pemisahan peserta yang saling mengenal melalui analisis jejaring sosial menjadi strategi penting dalam pengaturan tempat duduk [5].

Pengaturan tempat duduk dalam ujian memainkan peran penting dalam menciptakan lingkungan ujian yang berintegritas. Penelitian yang dilakukan oleh Levitt dan Lin menunjukkan bahwa penempatan acak tempat duduk secara signifikan mengurangi insiden kecurangan dibandingkan dengan membiarkan peserta memilih tempat duduk sendiri [6]. Teori perilaku terencana (planned behavior theory) mendukung temuan ini dengan menegaskan bahwa kecurangan terjadi sebagai hasil dari kesempatan dan niat untuk berbuat curang, sehingga upaya pencegahan harus bertindak pada kedua faktor situasional dan perilaku [7], [8].

Dalam konteks pengacakan, terdapat berbagai algoritma yang dapat digunakan untuk menghasilkan permutasi acak. Algoritma Fisher-Yates Shuffle, yang juga dikenal sebagai Knuth Shuffle, merupakan algoritma standar yang menghasilkan permutasi acak seragam (uniformly distributed) dengan kompleksitas waktu $O(n)$ [9]. Namun, algoritma Fisher-Yates memiliki kelemahan dalam konteks pengacakan tempat duduk ujian, yaitu memungkinkan satu atau lebih elemen tetap berada di posisi asalnya (fixed point), yang berarti ada kemungkinan peserta ujian tetap menempati tempat duduk yang sama dengan ujian sebelumnya [10]. Linear Congruent Method (LCM) juga sering digunakan namun memiliki kelemahan utama berupa hasil bilangan acak yang cenderung berpola dan berulang pada periode tertentu [11].

Algoritma Sattolo Shuffle, yang diperkenalkan oleh Sandra Sattolo pada tahun 1986, merupakan varian dari Fisher-Yates Shuffle yang secara khusus dirancang untuk menghasilkan permutasi siklis acak (random cyclic permutation) [12]. Perbedaan fundamental antara kedua algoritma ini terletak pada mekanisme pemilihan indeks acak, di mana pada Sattolo Shuffle indeks acak dipilih secara eksklusif dari rentang yang lebih kecil dari indeks saat ini, sehingga menghasilkan permutasi siklis di mana setiap elemen dipastikan berpindah dari posisi asalnya [12], [13].

Keunggulan utama algoritma Sattolo Shuffle adalah jaminan bahwa setiap elemen akan berpindah dari posisi asalnya, menghasilkan apa yang dikenal sebagai derangement. Dalam konteks pengacakan tempat duduk ujian, properti ini sangat berharga karena menjamin bahwa tidak ada peserta yang akan menempati kursi yang sama dengan penempatan sebelumnya, sehingga memutus pola kedekatan yang dapat dimanfaatkan untuk kecurangan [13], [14].

Penelitian terkait penerapan algoritma pengacakan pada berbagai konteks telah banyak dilakukan. Ibiyola dan Olu mengembangkan simulasi peningkatan algoritma Fisher-Yates Shuffle untuk pengacakan pada game kartu virtual [15]. Yusfrizal dkk. menerapkan algoritma Fisher-Yates Shuffle pada game edukasi mencocokkan gambar monumen dunia [16]. Hasan dkk. mengimplementasikan algoritma Fisher-Yates untuk mengacak soal ujian online penerimaan mahasiswa baru [10]. Febriani dkk. menggunakan Fisher-Yates Shuffle untuk pengacakan soal matematika pada ujian berbasis komputer [17]. Christopher dan Iswari menerapkan algoritma Sattolo Shuffle pada aplikasi pengenalan hewan berbasis gamifikasi [18]. Nasution dan Suginam melakukan modifikasi algoritma Fisher-Yates Shuffle menggunakan Linear Congruent Method untuk pembangkitan bilangan acak [19]. Angelina dan Wowor mengusulkan optimasi pembangkit bilangan acak menggunakan fungsi polinomial dan kombinasi metode iterasi [20]. Yohanna dkk. menerapkan algoritma Fisher-Yates Shuffle pada aplikasi ujian try out [21]. Arviansyah dkk. mengimplementasikan Fisher-Yates Shuffle pada aplikasi TOEFL preparation berbasis web [22]. Diharjo dkk. menggunakan metode Fisher-Yates Shuffle pada game edukasi bahasa Indonesia berbasis puzzle [23].

Meskipun berbagai penelitian telah mengeksplorasi penerapan algoritma pengacakan, penerapan spesifik algoritma Sattolo Shuffle pada sistem pengacakan denah tempat duduk peserta ujian untuk pencegahan kecurangan akademik belum banyak dikaji. Penelitian ini bertujuan untuk mengimplementasikan algoritma Sattolo Shuffle pada pengacakan 30 peserta ujian sekaligus mengevaluasi efektivitasnya dalam menghasilkan permutasi siklis yang optimal untuk mencegah kecurangan akademik.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini termasuk ke dalam jenis penelitian eksperimental yang bertujuan untuk menerapkan dan mengevaluasi algoritma Sattolo Shuffle pada sistem pengacakan denah tempat duduk ujian. Tahapan penelitian meliputi:

1. Studi literatur tentang algoritma pengacakan, permutasi siklis, dan pencegahan kecurangan akademik;
2. Perancangan sistem yang mendukung representasi 30 peserta ujian dalam array dan pemetaan ke denah ruang ujian;
3. Implementasi algoritma Sattolo Shuffle untuk pengacakan posisi 30 peserta ujian; dan
4. Pengujian serta evaluasi hasil pengacakan melalui verifikasi fixed point, analisis siklus, dan distribusi penempatan.

2.2 Denah Ruang Ujian

Ruang ujian yang digunakan memiliki kapasitas 30 kursi yang ditata dalam format 5 kolom dan 6 baris. Setiap kursi diidentifikasi dengan nomor posisi 1 hingga 30, dimana penomoran dilakukan dari kiri ke kanan dan dari atas ke bawah. Peserta ujian berjumlah 30 orang yang diidentifikasi sebagai peserta 1 hingga 30. Pada konfigurasi awal (sebelum pengacakan), peserta ke- i menempati posisi ke- i .

Tabel 1. Denah Ruang Ujian Dengan Konfigurasi 5 Kolom $\times$ 6 Baris (30 Kursi)

Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5
Posisi 1	Posisi 2	Posisi 3	Posisi 4	Posisi 5
Posisi 6	Posisi 7	Posisi 8	Posisi 9	Posisi 10
Posisi 11	Posisi 12	Posisi 13	Posisi 14	Posisi 15
Posisi 16	Posisi 17	Posisi 18	Posisi 19	Posisi 20
Posisi 21	Posisi 22	Posisi 23	Posisi 24	Posisi 25
Posisi 26	Posisi 27	Posisi 28	Posisi 29	Posisi 30

2.3 Algoritma Sattolo Shuffle

Algoritma Sattolo Shuffle merupakan modifikasi dari algoritma Fisher-Yates Shuffle yang diperkenalkan oleh Sandra Sattolo pada tahun 1986. Perbedaan utama terletak pada rentang pemilihan indeks acak: pada Fisher-Yates indeks j dipilih dari 0 hingga i (inklusif), sedangkan pada Sattolo dari 0 hingga $i-1$ (eksklusif terhadap i). Perbedaan ini menghasilkan permutasi siklis dari $(n-1)!$ kemungkinan, di mana setiap elemen dipastikan berpindah dari posisi asalnya [12], [13], [14], [24], [25].

Langkah-langkah algoritma Sattolo Shuffle:

1. Input Data: Masukkan 30 peserta ke dalam array $A[0..29]$.
2. Inisialisasi: Set i = panjang array (30).
3. Pemilihan Bilangan Acak: Pilih bilangan acak r antara 0 dan $i-1$ (eksklusif terhadap i).
4. Pertukaran Elemen: Tukar elemen $A[i-1]$ dengan $A[r]$.
5. Iterasi: Kurangi i dengan 1. Jika $i > 1$, ulangi langkah 3; jika tidak, proses selesai.

3. HASIL DAN PEMBAHASAN

Pada penelitian ini, peserta ujian yang digunakan berjumlah 30 orang. Setiap peserta direpresentasikan dengan bilangan integer dari 1 sampai 30, di mana peserta 1 menempati posisi 1, peserta 2 menempati posisi 2, dan seterusnya. Array awal sebelum pengacakan adalah $A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]$ dengan indeks 0 sampai 29.

3.1 Proses Iterasi Pengacakan 30 Peserta Ujian

Berikut ini adalah proses pengacakan lengkap 30 peserta ujian menggunakan algoritma Sattolo Shuffle. Proses pengacakan membutuhkan 29 iterasi ($n-1$ iterasi untuk $n=30$ elemen). Pada setiap iterasi, dipilih bilangan acak r dari rentang 0 sampai $i-1$ (eksklusif terhadap i), kemudian dilakukan pertukaran elemen pada posisi i dengan elemen pada posisi r .

Diketahui:

1. Array $A = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]$
2. Panjang array $n = 30$, sehingga dilakukan 29 iterasi.
3. Proses iterasi dibagi menjadi 2 tabel yang setiap tabel masing-masing terdapat 15 atau 14 proses iterasi, hal ini dilakukan karena proses yang dilakukan terlalu panjang untuk disajikan dalam 1 tabel dalam 1 halaman.

Tabel 2. Proses Iterasi Tahap-1 Algoritma Sattolo Shuffle pada 30 Peserta Ujian

Iterasi	i	r	Operasi Tukar	Hasil Array A
1	29	7	$A[29] \leftrightarrow A[7]$	[1,2,3,4,5,6,7,30,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,8]
2	28	2	$A[28] \leftrightarrow A[2]$	[1,2,29,4,5,6,7,30,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,3,8]
3	27	15	$A[27] \leftrightarrow A[15]$	[1,2,29,4,5,6,7,30,9,10,11,12,13,14,15,28,17,18,19,20,21,22,23,24,25,26,27,16,3,8]
4	26	5	$A[26] \leftrightarrow A[5]$	[1,2,29,4,5,27,7,30,9,10,11,12,13,14,15,28,17,18,19,20,21,22,23,24,25,26,6,16,3,8]
5	25	9	$A[25] \leftrightarrow A[9]$	[1,2,29,4,5,27,7,30,9,26,11,12,13,14,15,28,17,18,19,20,21,22,23,24,25,10,6,16,3,8]
6	24	0	$A[24] \leftrightarrow A[0]$	[25,2,29,4,5,27,7,30,9,26,11,12,13,14,15,28,17,18,19,20,21,22,23,24,1,10,6,16,3,8]
7	23	10	$A[23] \leftrightarrow A[10]$	[25,2,29,4,5,27,7,30,9,26,24,12,13,14,15,28,17,18,19,20,21,22,23,11,1,10,6,16,3,8]
8	22	2	$A[22] \leftrightarrow A[2]$	[25,2,23,4,5,27,7,30,9,26,24,12,13,14,15,28,17,18,19,20,21,22,29,11,1,10,6,16,3,8]
9	21	18	$A[21] \leftrightarrow A[18]$	[25,2,23,4,5,27,7,30,9,26,24,12,13,14,15,28,17,18,22,20,21,19,29,11,1,10,6,16,3,8]
10	20	19	$A[20] \leftrightarrow A[19]$	[25,2,23,4,5,27,7,30,9,26,24,12,13,14,15,28,17,18,22,21,20,19,29,11,1,10,6,16,3,8]
11	19	16	$A[19] \leftrightarrow A[16]$	[25,2,23,4,5,27,7,30,9,26,24,12,13,14,15,28,21,18,22,17,20,19,29,11,1,10,6,16,3,8]
12	18	8	$A[18] \leftrightarrow A[8]$	[25,2,23,4,5,27,7,30,22,26,24,12,13,14,15,28,21,18,9,17,20,19,29,11,1,10,6,16,3,8]
13	17	10	$A[17] \leftrightarrow A[10]$	[25,2,23,4,5,27,7,30,22,26,18,12,13,14,15,28,21,24,9,17,20,19,29,11,1,10,6,16,3,8]
14	16	13	$A[16] \leftrightarrow A[13]$	[25,2,23,4,5,27,7,30,22,26,18,12,13,21,15,28,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
15	15	8	$A[15] \leftrightarrow A[8]$	[25,2,23,4,5,27,7,30,28,26,18,12,13,21,15,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
16	14	1	$A[14] \leftrightarrow A[1]$	[25,15,23,4,5,27,7,30,28,26,18,12,13,21,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
17	13	1	$A[13] \leftrightarrow A[1]$	[25,21,23,4,5,27,7,30,28,26,18,12,13,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
18	12	11	$A[12] \leftrightarrow A[11]$	[25,21,23,4,5,27,7,30,28,26,18,13,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
19	11	10	$A[11] \leftrightarrow A[10]$	[25,21,23,4,5,27,7,30,28,26,13,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
20	10	8	$A[10] \leftrightarrow A[8]$	[25,21,23,4,5,27,7,30,13,26,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
21	9	4	$A[9] \leftrightarrow A[4]$	[25,21,23,4,26,27,7,30,13,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
22	8	1	$A[8] \leftrightarrow A[1]$	[25,13,23,4,26,27,7,30,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
23	7	3	$A[7] \leftrightarrow A[3]$	[25,13,23,30,26,27,7,4,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
24	6	1	$A[6] \leftrightarrow A[1]$	[25,7,23,30,26,27,13,4,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
25	5	4	$A[5] \leftrightarrow A[4]$	[25,7,23,30,27,26,13,4,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
26	4	3	$A[4] \leftrightarrow A[3]$	[25,7,23,27,30,26,13,4,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
27	3	0	$A[3] \leftrightarrow A[0]$	[27,7,23,25,30,26,13,4,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
28	2	1	$A[2] \leftrightarrow A[1]$	[27,23,7,25,30,26,13,4,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]
29	1	0	$A[1] \leftrightarrow A[0]$	[23,27,7,25,30,26,13,4,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]

Setelah semua proses iterasi selesai, maka hasil akhirnya adalah $A = [23,27,7,25,30,26,13,4,21,5,28,18,12,15,2,22,14,24,9,17,20,19,29,11,1,10,6,16,3,8]$. Hasil ini menunjukkan bahwa seluruh 30 peserta telah berpindah dari posisi asalnya. Sebagai contoh, peserta yang semula di posisi 1 kini berada di posisi lain, peserta yang semula di posisi 30 juga berpindah ke posisi yang berbeda, dan demikian seterusnya untuk seluruh peserta.

3.2 Verifikasi Hasil Pengacakan

Verifikasi hasil pengacakan merupakan tahapan kritis untuk membuktikan bahwa algoritma Sattolo Shuffle benar-benar menghasilkan permutasi siklis (cyclic permutation). Dalam teori permutasi, sebuah permutasi dikatakan siklis apabila tidak terdapat satupun elemen yang tetap berada di posisi asalnya setelah proses transformasi, yang dalam terminologi matematika disebut sebagai derangement atau fixed-point-free permutation [12], [26]. Verifikasi dilakukan melalui tiga mekanisme pengujian: (a) pengecekan fixed point pada seluruh 30 elemen, (b) analisis pola perpindahan antar posisi, dan (c) pelacakan siklus permutasi.

Mekanisme pertama adalah pengecekan fixed point, yaitu membandingkan nilai pada setiap indeks sebelum dan sesudah pengacakan. Pada array awal, indeks ke- i berisi nilai $i+1$. Jika setelah pengacakan masih terdapat indeks ke- i yang berisi nilai $i+1$, maka elemen tersebut merupakan fixed point dan permutasi tidak bersifat siklis. Tabel 3 menunjukkan hasil verifikasi untuk seluruh 30 posisi dalam format dua kolom.

Tabel 3. Verifikasi Perpindahan Seluruh 30 Elemen (Fixed Point Analysis)

Idx	Awal	Akhir	Pindah?	Idx	Awal	Akhir	Pindah?
0	1	23	Ya	15	16	22	Ya
1	2	27	Ya	16	17	14	Ya
2	3	7	Ya	17	18	24	Ya
3	4	25	Ya	18	19	9	Ya
4	5	30	Ya	19	20	17	Ya
5	6	26	Ya	20	21	20	Ya
6	7	13	Ya	21	22	19	Ya
7	8	4	Ya	22	23	29	Ya
8	9	21	Ya	23	24	11	Ya
9	10	5	Ya	24	25	1	Ya
10	11	28	Ya	25	26	10	Ya
11	12	18	Ya	26	27	6	Ya
12	13	12	Ya	27	28	16	Ya
13	14	15	Ya	28	29	3	Ya
14	15	2	Ya	29	30	8	Ya

Berdasarkan Tabel 3, seluruh 30 elemen telah berhasil diverifikasi dan tidak ditemukan satupun fixed point (0 dari 30 elemen). Ini berarti probabilitas elemen tetap di posisi asal adalah 0%, yang secara matematis membuktikan bahwa hasil pengacakan merupakan permutasi siklis sempurna. Sebagai perbandingan, pada algoritma Fisher-Yates Shuffle standar, probabilitas terdapat minimal satu fixed point dalam permutasi acak n elemen mendekati $1 - 1/e$ (sekitar 63,2%), yang berarti sekitar 63% dari seluruh permutasi acak akan memiliki setidaknya satu elemen yang tidak berpindah [15], [26].

Mekanisme kedua adalah analisis pola perpindahan. Dilakukan penghitungan jarak perpindahan setiap elemen dari posisi asalnya, yaitu selisih absolut antara posisi awal dan posisi akhir. Variasi jarak perpindahan yang besar dan tidak seragam menunjukkan distribusi pengacakan yang merata dan tidak terpolo [11], [14].

Mekanisme ketiga adalah pelacakan siklus permutasi. Dalam permutasi siklis yang dihasilkan Sattolo Shuffle, seluruh elemen membentuk satu siklus tunggal. Jika dimulai dari elemen tertentu dan mengikuti rantai perpindahannya, seluruh 30 elemen akan melewati sebelum kembali ke elemen awal. Pada hasil ini, rantai siklus dimulai dari posisi 0 yang berisi nilai 23, kemudian posisi 22 berisi 29, dan seterusnya hingga kembali ke posisi 0 setelah melewati seluruh 30 posisi. Satu siklus tunggal yang mencakup semua elemen ini mengkonfirmasi keberhasilan algoritma [12], [13].

3.3 Distribusi Peserta Ke Denah Ruang Ujian

Setelah proses pengacakan selesai, 30 peserta yang telah teracak didistribusikan ke denah ruang ujian sesuai dengan konfigurasi 5 kolom $\times$ 6 baris. Tabel 4 menyajikan pemetaan hasil pengacakan ke denah ruang ujian.

Tabel 4. Distribusi Peserta Hasil Pengacakan Ke Denah Ruang Ujian

	Kolom 1	Kolom 2	Kolom 3	Kolom 4	Kolom 5
Baris 1	P23	P27	P7	P25	P30
Baris 2	P26	P13	P4	P21	P5
Baris 3	P28	P18	P12	P15	P2
Baris 4	P22	P14	P24	P9	P17
Baris 5	P20	P19	P29	P11	P1
Baris 6	P10	P6	P16	P3	P8

Keterangan: P = Peserta. Contoh: P23 berarti Peserta 23 menempati posisi tersebut.

Berdasarkan Tabel 4, terlihat bahwa seluruh peserta telah diacak ke posisi yang berbeda dari posisi asalnya. Tidak ada peserta yang menempati posisi dengan nomor yang sama dengan nomor pesertanya, mengkonfirmasi properti derangement dari algoritma Sattolo Shuffle.

3.4 Analisis Efektivitas Algoritma

Untuk mengevaluasi efektivitas algoritma Sattolo Shuffle secara komprehensif dalam konteks pengacakan tempat duduk ujian, analisis dilakukan dari empat perspektif:

1. Perbandingan dengan algoritma pengacakan lainnya,
2. Analisis kompleksitas komputasi,

3. Pengujian konsistensi melalui pengacakan berulang, dan
4. Evaluasi kesesuaian untuk pencegahan kecurangan akademik.

3.4.1 Perbandingan Dengan Algoritma Lain

Tabel 5. Perbandingan Algoritma Sattolo Shuffle dengan Metode Lainnya

Aspek	Sattolo Shuffle	Fisher-Yates Shuffle	LCM
Jenis Permutasi	Siklis	Acak Umum	Deterministik
Jumlah Kemungkinan	$(n-1)! = 29!$	$n! = 30!$	Terbatas (periodik)
Elemen Tetap di Posisi Asal	Tidak mungkin (0%)	Mungkin (~63%)	Tidak relevan
Pola Berulang	Tidak ada	Tidak ada	Ada (periodik)
Kompleksitas Waktu	$O(n)$	$O(n)$	$O(n)$
Kompleksitas Ruang	$O(1)$ in-place	$O(1)$ in-place	$O(n)$ baru
Jumlah Iterasi ($n=30$)	29	29	30+
Derangement	Selalu (guaranteed)	Tidak selalu	Tidak dijamin
Kesesuaian Pengacakan Ujian	Sangat Baik	Baik	Kurang Baik

Berdasarkan Tabel 5, sifat derangement Sattolo Shuffle menjamin 100% elemen berpindah posisi. Pada Fisher-Yates, probabilitas seluruh elemen berpindah hanya sekitar 36,8% (1/e), sehingga rata-rata terdapat 1 peserta yang tetap di posisi asal pada setiap pengacakan. Dibandingkan LCM yang bersifat deterministik dan menghasilkan pola periodik, Sattolo Shuffle menghasilkan distribusi yang lebih tidak terduga [9], [11], [12], [19].

3.4.2 Analisis Kompleksitas Komputasi

Algoritma Sattolo Shuffle memiliki kompleksitas waktu $O(n)$ dan kompleksitas ruang $O(1)$. Untuk 30 peserta, dilakukan tepat 29 operasi pertukaran, masing-masing terdiri dari: (1) pembangkitan bilangan acak r dalam rentang $[0, i-1]$, (2) penyimpanan sementara nilai $A[i]$, dan (3) pertukaran nilai $A[i]$ dengan $A[r]$. Pengacakan bersifat in-place tanpa alokasi memori tambahan selain variabel sementara, menjadikannya lebih efisien dibandingkan LCM yang memerlukan array terpisah [11], [12].

3.4.3 Pengujian Konsistensi Pengacakan Berulang

Untuk memastikan konsistensi kinerja, dilakukan tiga kali pengujian berulang terhadap 30 peserta yang sama. Tabel 6 menunjukkan hasilnya.

Tabel 6. Hasil Pengujian Konsistensi Pengacakan Berulang

Parameter	Pengujian 1	Pengujian 2	Pengujian 3
Jumlah Iterasi	29	29	29
Fixed Point	0 (0%)	0 (0%)	0 (0%)
Pola Berulang	Tidak ditemukan	Tidak ditemukan	Tidak ditemukan
Jenis Permutasi	Siklis (1 siklus)	Siklis (1 siklus)	Siklis (1 siklus)
Distribusi Posisi Merata	Ya	Ya	Ya
Hasil Identik dgn Lainnya	Tidak	Tidak	Tidak

Hasil pengujian pada Tabel 6 menunjukkan bahwa seluruh tiga pengujian menghasilkan 0 fixed point, permutasi siklis dengan 1 siklus, dan tidak ada pola berulang. Ketiga hasil pengacakan juga tidak identik satu sama lain, membuktikan bahwa algoritma menghasilkan konfigurasi yang berbeda pada setiap eksekusi.

3.4.4 Evaluasi Kesesuaian Untuk Pencegahan Kecurangan Akademik

Dari perspektif pencegahan kecurangan akademik, algoritma Sattolo Shuffle memberikan beberapa keuntungan spesifik. Pertama, jaminan bahwa setiap peserta berpindah dari posisi asalnya memastikan setiap sesi ujian memiliki konfigurasi tempat duduk yang benar-benar berbeda, mencegah peserta menghafal pola kedekatan tertentu untuk merencanakan strategi mencontek [6], [8]. Kedua, properti permutasi siklis tunggal memastikan bahwa seluruh peserta terhubung dalam satu rantai permutasi, sehingga tidak ada cluster peserta yang tetap berkelompok [12], [13]. Ketiga, jumlah kemungkinan konfigurasi yang sangat besar yaitu $(n-1)! = 29! \approx 8,84 \times 10^{30}$ menjadikan prediksi penempatan secara praktis mustahil [14], [27]. Keempat, efisiensi komputasi $O(n)$ waktu dan $O(1)$ ruang menjadikannya sangat cocok untuk implementasi pada berbagai platform termasuk perangkat mobile dengan sumber daya terbatas [9], [12].

Temuan ini sejalan dengan penelitian Levitt dan Lin [6] yang menunjukkan bahwa penempatan acak secara signifikan mengurangi kecurangan. Dalam konteks teori perilaku terencana [7], [8], sistem pengacakan Sattolo Shuffle secara efektif mengurangi opportunity (kesempatan) untuk kecurangan dengan menghilangkan prediktabilitas dan kedekatan yang konsisten antar peserta ujian. Dibandingkan dengan pengacakan Fisher-Yates biasa yang memiliki risiko fixed point (peserta tetap di posisi asal), Sattolo Shuffle memberikan jaminan matematis yang lebih kuat untuk pencegahan kecurangan berbasis kedekatan tempat duduk [9], [12].

4. KESIMPULAN

Penelitian ini berhasil mengimplementasikan algoritma Sattolo Shuffle pada sistem pengacakan denah tempat duduk untuk 30 peserta ujian sebagai upaya pencegahan kecurangan akademik. Proses pengacakan dilakukan melalui 29 iterasi, dan hasil verifikasi menunjukkan bahwa tidak ada satupun peserta yang tetap berada di posisi asalnya (0 fixed point), membuktikan keberhasilan algoritma dalam menghasilkan permutasi siklis yang sempurna. Algoritma Sattolo Shuffle terbukti efisien dengan kompleksitas waktu $O(n)$ dan kompleksitas ruang $O(1)$, menjadikannya cocok untuk implementasi pada berbagai platform pengelolaan ujian. Distribusi peserta ke denah ruang ujian menunjukkan penyebaran yang merata tanpa pola tertentu, sehingga secara efektif mengurangi peluang kecurangan berbasis kedekatan tempat duduk. Dibandingkan dengan algoritma Fisher-Yates Shuffle yang memiliki risiko fixed point (~63% kemungkinan ada peserta yang tidak berpindah) dan LCM yang bersifat deterministik, Sattolo Shuffle memberikan jaminan matematis 100% bahwa seluruh peserta akan berpindah posisi. Jumlah kemungkinan konfigurasi sebanyak $29! \approx 8,84 \times 10^{30}$ menjadikan prediksi penempatan secara praktis mustahil, sehingga memutus strategi kecurangan yang bergantung pada pengetahuan posisi duduk sebelumnya. Sebagai saran untuk pengembangan lebih lanjut, algoritma Sattolo Shuffle dapat dikombinasikan dengan constraint tambahan seperti pemisahan peserta dari program studi yang sama, diintegrasikan dengan sistem informasi akademik untuk otomatisasi pengelolaan ujian, serta dapat dikembangkan antarmuka pengguna berbasis web atau mobile yang memudahkan pengelola ujian dalam mengoperasikan sistem secara praktis.

REFERENCES

- [1] D. L. McCabe, K. D. Butterfield, And L. K. Treviño, *Cheating In College - Why Students Do It And What Educators Can Do About It*. The Johns Hopkins University Press, 2012.
- [2] T. P. Cronan, J. K. Mullins, And D. E. Douglas, "Further Understanding Factors That Explain Freshman Business Students' Academic Integrity Intention And Behavior: Plagiarism And Sharing Homework," *Journal Of Business Ethics*, Vol. 147, No. 1, Pp. 197–220, Jan. 2018, Doi: 10.1007/S10551-015-2988-3.
- [3] W. J. Bowers, *Student Dishonesty And Its Control In College*. New York: Bureau Of Applied Social Research, Columbia University, 1964. [Online]. Available: <https://eric.ed.gov/?id=Ed003834>
- [4] R. A. Bernardi, A. V. Baca, K. S. Landers, And M. B. Witek, "Methods Of Cheating And Deterrents To Classroom Cheating: An International Study," *Ethics Behav.*, Vol. 18, No. 4, Pp. 373–391, Oct. 2008, Doi: 10.1080/10508420701713030.
- [5] A. Topirceanu, "Breaking Up Friendships In Exams: A Case Study For Minimizing Student Cheating In Higher Education Using Social Network Analysis," *Comput. Educ.*, Vol. 115, Pp. 171–187, Dec. 2017, Doi: 10.1016/J.Compedu.2017.08.008.
- [6] S. D. Levitt And M.-J. Lin, "Catching Cheating Students," 2015. [Online]. Available: <http://www.nber.org/papers/W21628>
- [7] I. Ajzen, "The Theory Of Planned Behavior," 1991.
- [8] T. S. Harding, M. J. Mayhew, C. J. Finelli, And D. D. Carpenter, "The Theory Of Planned Behavior As A Model Of Academic Dishonesty In Engineering And Humanities Undergraduates," *Ethics Behav.*, Vol. 17, No. 3, Pp. 255–279, 2007, Doi: 10.1080/10508420701519239.
- [9] D. E. Knuth, "The Art Of Computer Programming."
- [10] 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.
- [11] S. Bulolo, "Implementasi Metode Linear Congruent Method (Lcm) Pada Simulasi Ujian Akhir Sekolah Menengah Kejuruan Lolomatua," In *Prosiding Seminar Nasional Teknologi Informatika*, 2019, Pp. 60–64.
- [12] S. Sattolo, "An Algorithm To Generate A Random Cyclic Permutation," 1986.
- [13] M. C. Wilson, "Overview Of Sattolo's Algorithm," 2002. [Online]. Available: <http://algo.inria.fr/seminars/>
- [14] H. Prodinger, "On The Analysis Of An Algorithm To Generate A Random Cyclic Permutation," *Ars Combinatoria*, 2002.
- [15] A. Olu, "A Simulated Enhancement Of Fisher-Yates Algorithm For Shuffling In Virtual Card Games Using Domain-Specific Data Structures," 2012.
- [16] Yusfrizal, D. Adhar, U. Indriani, E. Panggabean, A. Sabir, And H. Kurniawan, "Application Of The Fisher-Yates Shuffle Algorithm In The Game Matching The World Monument Picture," In *2020 2nd International Conference On Cybernetics And Intelligent System, Icoris 2020*, Institute Of Electrical And Electronics Engineers Inc., Oct. 2020. Doi: 10.1109/Icoris50180.2020.9320766.
- [17] 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.
- [18] 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," *Ijnm (International Journal Of New Media Technology)*, Vol. Iv, No. 1, 2017.
- [19] S. D. Nasution And S. Suginam, "Modifikasi Algoritma Fisher Yates Shuffle Menggunakan Linear Congruent Method Untuk Pembangkitan Bilangan Acak," *Jurnal Ilmu Komputer*, Vol. 12, No. 2, Pp. 101–106, 2019.
- [20] S. Angelina And A. D. Wowor, "Optimasi Pembangkit Bilangan Acak Dengan Fungsi Polinomial Dan Kombinasi Metode Iterasi," *Jiko (Jurnal Informatika Dan Komputer)*, Vol. 8, No. 2, P. 367, Sep. 2024, Doi: 10.26798/Jiko.V8i2.1313.
- [21] 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.

- [22] 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.
- [23] W. Diharjo, D. Ahkam Sani, And M. Firman Arif, “Game Edukasi Bahasa Indonesia Menggunakan Metode Fisher Yates Shuffle Pada Genre Puzzle Game,” *Integer: Journal Of Information Technology*, Vol. 5, No. 2, Pp. 23–35, 2020.
- [24] S. D. Nasution And G. L. Ginting, “Implementasi Algoritma Sattolo Shuffle Untuk Optimasi Pengacakan Pada Game Solitaire,” *Bulletin Of Artificial Intelligence*, Vol. 3, No. 2, Pp. 42–49, Oct. 2024, Doi: 10.62866/Buai.V3i2.200.
- [25] S. D. Nasution And G. L. Ginting, “Penerapan Algoritma Sattolo Shuffle Dalam Sistem Distribusi Kupon Undian Digital Pada Platform E-Commerce,” *Bulletin Of Artificial Intelligence*, Vol. 4, No. 1, Pp. 10–15, Apr. 2025, Doi: 10.62866/Buai.V4i1.221.
- [26] M. Hassani, “Derangements And Applications,” 2003.
- [27] H. M. Mahmoud, “Mixed Distributions In Sattolo’s Algorithm For Cyclic Permutations Via Randomization And,” *J. Appl. Probab.*, Vol. 40, No. 3, Pp. 790–796, 2003, [Online]. Available: [Http://About.Jstor.Org/Terms](http://About.Jstor.Org/Terms)