Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.2025129381
Vol. 12, No. 2, April 2025, hlm. 457-468 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

PENINGKATAN KEAMANAN PENGACAKAN SOAL UJIAN COMPUTER ASSISTED TEST (CAT) PENERIMAAN MAHASISWA UIN IMAM BONJOL

Ozzy Secio Riza[*1], Rosalina[2], Rendi Saputra[3], Aulia Arham[4]

[1,2,3,4]Universitas Islam Negeri Imam Bonjol, Padang
Email: [1]ozzysecioriza@uinib.ac.id, [2]rosalina@uinib.ac.id, [3]rendi.saputra@uinib.ac.id, [4]auliaarham@uinib.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 25 Februari 2025, diterima untuk diterbitkan: 14 April 2025)

Abstrak

Teknologi informasi telah membuka akses luas terhadap sumber belajar dan berperan signifikan dalam pelaksanaan tes ujian masuk perguruan tinggi, termasuk adopsi ujian berbasis komputer. Proses penerimaan mahasiswa baru merupakan tugas yang krusial dan menantang. Penelitian ini bertujuan untuk meningkatkan keamanan sistem ujian masuk mahasiswa baru jalur mandiri di UIN Imam Bonjol Padang dengan memperbaiki teknik pengacakan soal pada *Computer Assisted Test* (CAT). Pengacakan soal dalam sistem CAT yang ada saat ini masih menggunakan prosedur pengacakan standar, yang secara kriptografi tidak aman karena prediktabilitasnya. Untuk mengatasi masalah ini, penelitian mengusulkan penggunaan kombinasi algoritma Fisher-Yates Shuffle dan Chaos Map. Fisher-Yates Shuffle menawarkan efektivitas pengacakan tinggi dengan kompleksitas optimal (O(n)), sementara Chaos Map sangat sensitif terhadap nilai awal, sehingga cocok digunakan untuk pengacakan yang lebih aman. Dengan algoritma ini, soal dan pilihan jawaban diacak berdasarkan nomor peserta, sehingga setiap peserta mendapat soal yang berbeda, mengurangi kemungkinan kecurangan. Hasil penelitian menunjukkan bahwa sistem CAT dengan teknik pengacakan baru ini lebih aman dan efektif dimana terjadi penurunan kemiripan soal sebesar 67,85% dalam tingkat kemiripan soal dari 0,339 menjadi 0,109, yang menunjukkan bahwa algoritma yang diusulkan menghasilkan pengacakan soal yang lebih merata dan lebih sedikit pengulangan soal di antara peserta ujian.

Kata kunci: *Computer Based Test (CBT), Kriptografi, Fisher-Yates Shuffle, Chaos Map*

ENHANCING THE SECURITY OF RANDOMIZED EXAM QUESTIONS IN THE COMPUTER-ASSISTED TEST (CAT) FOR STUDENT ADMISSIONS AT UIN IMAM BONJOL

Abstract

Information technology has provided wide access to learning resources and played a significant role in the implementation of college entrance exams, including the adoption of computer-based exams. The process of admitting new students is a crucial and challenging task. This study aims to enhance the security of the independent student entrance exam system at UIN Imam Bonjol Padang by improving the question randomization technique in the Computer-Assisted Test (CAT). The current CAT system still uses a standard randomization procedure, which is cryptographically insecure due to its predictability. To address this issue, the study proposes a combination of the Fisher-Yates Shuffle and Chaos Map algorithms. Fisher-Yates Shuffle offers high randomization effectiveness with optimal complexity (O(n)), while Chaos Map is highly sensitive to initial values, making it ideal for more secure randomization. With this approach, questions and answer choices are randomized based on participant numbers, ensuring each participant receives different questions, thus reducing the possibility of cheating. The results show that the CAT system with this new randomization technique is more secure and effective where there is a 67.85% decrease in the question similarity rate from 0.339 to 0.109, which indicates that the proposed algorithm results in more even question randomization and less question repetition among examinees.

Keywords: *Computer Based Test (CBT), Cryptography, Fisher-Yates Shuffle, Chaos Map*

1. PENDAHULUAN

Perkembangan teknologi informasi telah menghasilkan peningkatan pesat dalam kemampuan komputasi. Prosesor yang lebih kuat, penyimpanan data yang lebih besar, dan kemampuan komputasi yang lebih tinggi telah memungkinkan pengembangan aplikasi dan sistem yang lebih canggih (Savitri, 2019). Ini memungkinkan pemrosesan data yang lebih cepat dan lebih efisien, memungkinkan pengolahan dan analisis data yang lebih kompleks, serta mempercepat inovasi dalam berbagai bidang. Secara garis besar perkembangan teknologi informasi telah memiliki dampak yang signifikan dalam berbagai aspek kehidupan, tidak terkecuali dunia pendidikan. Teknologi informasi telah membuka pintu akses yang luas terhadap informasi dan sumber belajar, mendorong pembelajaran jarak jauh, serta berdampak signifikan dalam pelaksanaan tes ujian masuk perguruan tinggi (Agustina et al., 2023).

Dalam dunia pendidikan, proses penerimaan mahasiswa baru merupakan tugas yang krusial dan menantang. Salah satu komponen kunci dari proses ini adalah tes penerimaan, yang berfungsi sebagai sarana untuk mengukur dan mengevaluasi pengetahuan dan keterampilan calon mahasiswa (Suwarna, 2016). Tradisionalnya, ujian dilakukan secara manual dengan menggunakan kertas dan pena, yang memerlukan banyak waktu dan sumber daya dalam pengelolaannya. Hasil ujian seringkali butuh waktu lama untuk diproses, dikoreksi, dan dilaporkan. Selain itu, proses pengoreksian manual memiliki potensi kesalahan manusia dan ketidakakuratan dalam penilaian.

Salah satu perkembangan utama dalam pelaksanaan tes ujian masuk perguruan tinggi adalah adopsi ujian berbasis komputer, atau yang biasa dikenal dengan Computer Assisted Test (CAT) (Bahari, 2021), pendekatan dalam pengukuran dan evaluasi pendidikan mengalami perubahan signifikan. Sistem ujian berbasis komputer memungkinkan calon mahasiswa untuk menjawab pertanyaan ujian dengan menggunakan perangkat komputer, dan sistem ini secara otomatis mengoreksi jawaban mereka. Hal ini memungkinkan hasil ujian untuk tersedia dengan cepat dan akurat. Keuntungan ini mengurangi beban, waktu dan biaya pekerjaan panitia seleksi.

Selain efisiensi dalam pengelolaan ujian, sistem ujian berbasis komputer juga memberikan manfaat lainnya. Keamanan ujian dapat ditingkatkan dengan teknologi ini. Risiko kecurangan ujian dapat diminimalkan dengan pengawasan yang lebih efisien (Akram, Kurniati and Salim, 2020). Data ujian dapat lebih aman disimpan dan dikelola, mengurangi risiko kehilangan atau kerusakan data. Lebih lanjut, sistem ujian berbasis komputer memungkinkan adanya ujian jarak jauh, di mana siswa dapat mengikuti ujian dari lokasi yang berbeda, tanpa harus hadir fisik di lokasi ujian. Ini memberikan fleksibilitas dalam pendidikan, terutama bagi mereka yang memiliki keterbatasan mobilitas atau yang berada di lokasi yang jauh dari pusat pendidikan.

UIN Imam Bonjol merupakan salah satu Perguruan Tinggi Keagamaan Islam Negeri yang telah memanfaatkan sistem ujian berbasis komputer dalam penerimaan mahasiswa baru sejak tahun 2019. Kehadiran sistem ini sangat membantu UIN Imam Bonjol dalam proses seleksi penerimaan mahasiswa baru jalur mandiri, baik dalam hal efisiensi pengelolaan ujian seperti persiapan soal, distribusi soal, dan pengawasan ujian, pengoreksian otomatis yang dapat mengurangi kesalahan manusia dan memastikan akurasi penilaian, kemudahan akses, keamanan dalam pengawasan ujian dan penyimpanan data ujian, analisis data ujian, dan dapat memudahkan proses perbaikan dan penyempurnaan dalam sistem evaluasi di UIN Imam Bonjol Padang. Oleh karena itu, inisiatif pengembangan sistem ujian berbasis komputer di UIN Imam Bonjol Padang memiliki signifikansi penting dalam meningkatkan mutu pendidikan, efisiensi administratif, dan memberikan peluang yang lebih adil bagi mahasiswa. Pengacakan soal ujian yang dilakukan dalam sistem CAT di UIN Imam Bonjol saat ini masih menggunakan prosedur pengacakan standar yang merupakan fungsi bawaan dari sistem, dimana fungsi ini secara kriptografi tidak aman, dimana jika penyerang dapat menentukan atau menebak nilai seed yang digunakan untuk menginisialisasi pengacakan, mereka dapat memprediksi seluruh rangkaian angka yang akan dihasilkan. Prediktabilitas ini dapat merusak keamanan pengacakan sistem CAT yang ada di UIN Imam Bonjol Padang, sehingga soal yang muncul pada peserta ujian dapat diprediksi. Sistem pengacakan yang diterapkan di CAT UIN Imam Bonjol Padang tidak menggunakan kunci awal pengacakan seperti menggunakan nomor peserta ujian atau lain sebagainya, sehingga terdapat kemungkinan calon mahasiswa baru mendapat soal yang sama dengan peserta lainnya. Kemudian dalam sistem CAT UIN Imam Bonjol Padang, yang dilakukan pengacakan hanya pada kemunculan soal ujian saja tidak sampai pada pengacakan pilihan jawaban pada soal ujian tersebut.

Untuk mencapai hal ini, algoritma Fisher-Yates Shuffle dapat digunakan. Algoritma fisher-yates shuffle adalah algoritma untuk menghasilkan permutasi acak dari suatu himpunan berhingga, dengan kata lain untuk mengacak suatu himpunan (Saokani et al., 2023). Fisher-yates shuffle dikenal memberikan performance yang baik dalam menghasilkan permutasi acak (Risnasari et al., 2021) yang telah diterapkan dalam beberapa topik pengamanan data seperti image encryption(Hazra et al., 2015; Alarood et al., 2022a; Madouri, Said and Pacha, 2022) serta pengacakan soal ujian (Duha, Dewi and Rahayu, n.d.; Febriani et al., 2021; Risnasari et al., 2021; Sugihartono and Putra, 2021; Saokani et al., 2023).

Penelitian sebelumnya (Sugihartono and Putra, 2021) menggunakan algoritma fisher yates shuffle untuk pengacakan soal pada sistem ujian kompetensi wartawan. Hasil uji coba pengacakan soal terhadap tiga soal pertama yang ditampilkan kepada responden menunjukkan bahwa algoritma Fisher-Yates Shuffle berhasil dalam 100% kasus dalam mengacak urutan soal yang disajikan kepada responden. (Duha, Dewi and Rahayu, n.d.) menggunakan algoritma fisher yates shuffle untuk pengacakan soal pada sistem tes standar online berbasis HOTS pada mahasiswa tingkat akhir di Universitas Negeri Medan. Hasil pengujian menunjukkan bahwa sistem yang dibangun berjalan dengan baik sesuai proses yang ada dalam sistem ujian tes standar online berbasis HOTS.

Algoritma Fisher-Yates Shuffle adalah metode yang terkenal dan banyak digunakan untuk mengacak daftar. Algoritma ini memastikan bahwa setiap elemen dalam daftar dikocok dan memiliki kesempatan yang sama untuk dipilih. Chaos map adalah fungsi matematika yang menunjukkan perilaku kacau, sehingga ideal untuk menghasilkan urutan yang tidak dapat diprediksi. Fungsi pemetaan yang mengandung unsur-unsur kekacauan sering dijumpai dalam studi sistem dinamik. Chos map biasanya digunakan dalam meningkatkan keamanan data dengan mengacak data dengan meningkatkan kekacauan data (Li et al., 2019a; Sahid et al., 2019; Alarood et al., 2022a; Madouri, Said and Pacha, 2022) (Dhoruri et al., 2019; Alarood et al., 2022; Li et al., 2019; Madouri et al., 2022). Penggunaan chaos map dalam pengacakan soal ujian diharapkan dapat meningkatkan randomization soal dan jawaban dalam sistem ujian berbasis CBT.

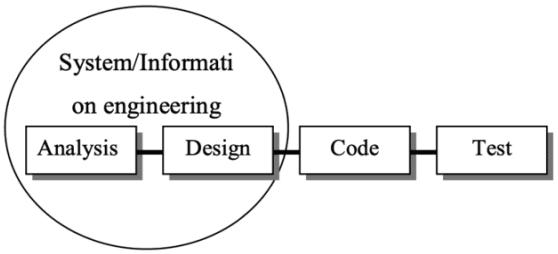
Penelitian ini mengusulkan kombinasi algoritma fisher yates shuffle dan chaos map untuk pengacakan soal dan pilihan jawaban penerimaan mahasiswa baru UIN Imam Bonjol Padang. Penambahan chaos map dalam penelitian ini diharapkan dapat meningkatkan hasil pengacakan soal sehingga meningkatkan keamanan soal ujian karena urutan soal dan pilihan jawaban yang teracak akan menyulitkan peserta untuk melakukan tindakan curang.

2. METODE PENELITIAN

2.1. Tahapan SDLC (*Systems Development Live Cycle*) Model *Waterfall*

Metode pengembangan perangkat lunak yang digunakan adalah metode sekuensial linier (*waterfall*) seperti pada Gambar 1. Tahapan-tahapan dalam metode ini meliputi:

1) Analisis Perangkat Lunak: Melakukan analisis terhadap data penelitian serta alat dan bahan yang akan digunakan. Tahap ini juga mencakup penentuan fitur-fitur yang akan dikembangkan dalam perangkat lunak.
2) Desain: Merancang antarmuka dan struktur perangkat lunak. Proses perancangan ini menggunakan Data Flow Diagram (DFD).



Gambar 1. Tahapan-tahapan metode *Linear Sequence (Waterfall)*

3) Code Generation/Coding: Pengembangan perangkat lunak dilakukan dengan menggunakan bahasa pemrograman PHP.
4) Testing/Pengujian: Pengujian perangkat lunak dilakukan oleh penulis untuk memastikan seluruh fitur yang dikembangkan berfungsi dengan baik melalui serangkaian uji coba.

2.2. Algoritma Pengacakan *Fisher-Yates*

Algoritma *Fisher-Yates Shuffle*, atau sering disebut juga dengan metode *Knuth Shuffle*, merupakan sebuah algoritma yang digunakan untuk mengacak urutan dari sejumlah item secara acak. Algoritma ini pertama kali diperkenalkan oleh Ronald Fisher dan Frank Yates pada tahun 1938, tetapi kemudian popularitasnya melejit setelah Donald Knuth menyertakannya dalam bukunya "*The Art of Computer Programming*".

Cara kerja algoritma *Fisher-Yates Shuffle* secara umum adalah:

1. Inisialisasi Array atau Daftar Item: Algoritma dimulai dengan sejumlah item yang ingin diacak, yang biasanya diwakili dalam bentuk array atau daftar.
2. Iterasi Melalui Setiap Item: Algoritma ini melakukan iterasi melalui setiap elemen dalam array, dimulai dari elemen pertama hingga elemen terakhir.
3. Penukaran Elemen: Setiap langkah iterasi melibatkan pemilihan elemen acak dari sisa elemen yang belum teracak (misalnya, elemen pada indeks yang lebih besar). Elemen yang dipilih secara acak akan ditukar dengan elemen pada posisi iterasi saat ini.
4. Penyelesaian Iterasi: Proses ini terus berlanjut hingga semua elemen dalam array telah terpilih atau teracak. Setelah selesai, array akan berisi urutan elemen yang diacak dengan distribusi yang merata dan acak.

Algoritma ini dianggap efisien karena kinerjanya tidak bergantung pada teknik pemilihan angka acak yang mahal secara komputasi. Sebaliknya, ia menggunakan algoritma pemilihan acak yang sederhana dengan kompleksitas waktu $O(n)$ untuk melakukan pengacakan terhadap n elemen.

Fisher-Yates Shuffle dikenal karena kemampuannya dalam menghasilkan urutan acak yang benar-benar acak, meminimalkan pola, dan memastikan bahwa setiap kemungkinan urutan elemen adalah sama kemungkinannya. Algoritma ini sering digunakan dalam pengacakan, pengujian, permainan, dan aplikasi lain yang membutuhkan urutan acak yang andal dan merata.

2.3. Bilangan Acak

Bilangan acak merupakan bilangan yang kemunculannya tidak dapat di-prediksi. Bilangan acak dapat berwujud sebagai bilangan bulat, bilangan riil antara 0 dan 1, atau string biner. Hingga kini, belum ada metode komputasi yang mampu menghasilkan deret bilangan acak secara sempurna (*truly random*). Bilangan acak yang dihasilkan oleh komputer disebut bilangan acak semu (*pseudo-random*) karena dihasilkan melalui algoritma matematika, sehingga pembangkitan bilangan tersebut dapat diulang kembali. Alat yang digunakan untuk menghasilkan bilangan acak semu ini dikenal dengan nama *Pseudo-random Number Generator* (PRNG). PRNG bersifat deterministik, yang berarti bahwa proses pembangkitan bilangan acak dapat direproduksi jika menggunakan umpan (*seed*) yang sama.

Bilangan acak semu digunakan dalam berbagai aplikasi, termasuk dalam pengacakan soal ujian, simulasi Monte Carlo, enkripsi data, dan permainan komputer. Meskipun tidak sepenuhnya acak, PRNG cukup memadai untuk banyak keperluan praktis, terutama karena kecepatan dan efisiensinya dalam menghasilkan deret bilangan yang tampak acak. Untuk meningkatkan kualitas bilangan acak semu, berbagai metode PRNG telah dikembangkan, termasuk metode Logistic Map yang berasal dari teori chaos. Metode ini mampu menghasilkan urutan bilangan yang kompleks dan sulit diprediksi, menjadikannya salah satu pilihan yang efektif untuk berbagai aplikasi yang membutuhkan bilangan acak.

Dalam kriptografi, bilangan acak memiliki peran yang sangat krusial. Beberapa penerapan bilangan acak ini di antaranya adalah:

1. Kunci Enkripsi: Bilangan acak digunakan untuk menghasilkan kunci enkripsi yang aman. Kunci yang benar-benar acak sulit diprediksi, sehingga meningkatkan keamanan data yang dienkripsi.
2. *Nonce: Nonce* (*number used once*) adalah bilangan acak yang digunakan hanya sekali dalam protokol komunikasi. *Nonce* membantu mencegah serangan *replay* dengan memastikan bahwa setiap sesi komunikasi menggunakan bilangan unik.
3. Vektor Inisialisasi (IV): IV adalah bilangan acak yang digunakan untuk menginisialisasi algoritma enkripsi, terutama dalam mode enkripsi blok. IV memastikan bahwa enkripsi data yang sama dengan kunci yang sama akan menghasilkan ciphertext yang berbeda.
4. *Salt*: Dalam penyimpanan kata sandi, salt adalah bilangan acak yang ditambahkan ke kata sandi sebelum di-*hash*. *Salt* membantu melindungi kata sandi dari serangan kamus dan serangan tabel pelangi (*rainbow table*).
5. Pembangkitan Kunci Asimetris: Bilangan acak digunakan dalam algoritma pembangkitan kunci asimetris (seperti RSA atau ECC) untuk menghasilkan pasangan kunci publik dan privat yang aman.

2.4. *Cryptographically Secure Pseudo-random Number Generator* (CSPRNG)

Cryptographically Secure Pseudo-random Number Generator (CSPRNG) adalah jenis pembangkit bilangan acak semu (PRNG) yang dirancang untuk memenuhi standar keamanan kriptografi. CSPRNG menghasilkan bilangan acak yang tidak hanya tampak acak tetapi juga sulit diprediksi, bahkan dengan pengetahuan se- bagian dari keluaran atau keadaan internal pembangkit (Zang et al., 2016; Gupta, Gupta and Singh, 2019). Karakteristik utama dari CSPRNG meliputi:

1. Keamanan: Bilangan acak yang dihasilkan tidak dapat diprediksi, meskipun sebagian dari nilai atau keadaan internal diketahui.
2. Uji Kekuatan: Lulus berbagai uji statistik untuk memastikan bahwa pola atau prediktabilitas dalam urutan bilangan acak tidak dapat ditemukan.
3. Resistensi terhadap Serangan: Tahan terhadap berbagai jenis serangan kriptografi, termasuk serangan analisis statistik dan serangan yang mencoba merekonstruksi keadaan internal.

Sebagian besar protokol kriptografi memerlukan pembuatan dan penggunaan nilai-nilai rahasia yang harus dijaga kerahasiaannya dari pihak yang tidak berwenang. Angka acak diperlukan untuk menciptakan *number used once* (*nonce*), tantangan, nilai blinding, dan byte padding (Pandit, Kumar and Mishra, 2023).

CSPRNG digunakan dalam berbagai aplikasi keamanan, termasuk generasi kunci enkripsi, nonce, salt, dan berbagai protokol keamanan lainnya. Algoritma umum yang digunakan dalam CSPRNG termasuk algoritma berbasis *hash*, *cipher block chaining* (CBC), dan teknik lainnya yang memastikan keamanan dan ketid- akpastian keluaran bilangan acak.

2.5. Teori Chaos

Teori chaos adalah cabang dari matematika dan fisika yang mempelajari sistem dinamis yang sangat sensitif terhadap kondisi awal, sebuah fenomena yang sering dikenal sebagai "efek kupu-kupu". Teori ini menunjukkan bahwa sistem- sistem yang tampaknya acak dan tidak dapat diprediksi sebenarnya memiliki pola, hukum, dan keteraturan yang mendasarinya

(Zang et al., 2016). Karakteristik utama dari teori chaos meliputi:

1. Sensitivitas terhadap Kondisi Awal: Perubahan kecil dalam kondisi awal dapat menghasilkan hasil yang sangat berbeda, membuat prediksi jangka panjang menjadi sangat sulit.
2. Dinamika Non-linear: Sistem chaos sering kali non-linear, artinya perubahan output tidak proporsional dengan perubahan input.
3. Fraktal: Banyak sistem chaos menunjukkan struktur fraktal, yang berarti mereka memiliki pola yang berulang pada skala yang berbeda.
4. Tak Terduga tetapi Teratur: Meskipun perilaku jangka panjang dari sistem chaos tampak acak, mereka mengikuti hukum matematis tertentu dan memiliki keteraturan dalam ketidakpastian.

Teori chaos memiliki aplikasi luas dalam berbagai bidang, termasuk meteorologi, ekologi, ekonomi, dan bahkan kriptografi. Dalam kriptografi, konsep

2.5. Chaos Map

Chaos map adalah fungsi matematika yang menggambarkan sistem dinamis yang menunjukkan perilaku chaos, yaitu perilaku yang sangat sensitif terhadap kondisi awal. Salah satu contoh paling terkenal dari chaos map adalah *Logistic Map*, yang digunakan untuk memodelkan populasi biologis tetapi juga diterapkan dalam kriptografi dan teori chaos untuk menghasilkan bilangan acak. *Logistic Map* adalah contoh dari fungsi polinomial derajat dua yang sering digunakan untuk menunjukkan kompleksitas sifat chaos yang dapat muncul dari persamaan yang sangat sederhana. Persamaan ini dipopulerkan oleh ahli biologi Robert May pada tahun 1976, yang melanjutkan karya sebelumnya dari Pierre Francois Verhulst yang mengembangkan persamaan logistik. Persamaan matematikanya adalah:

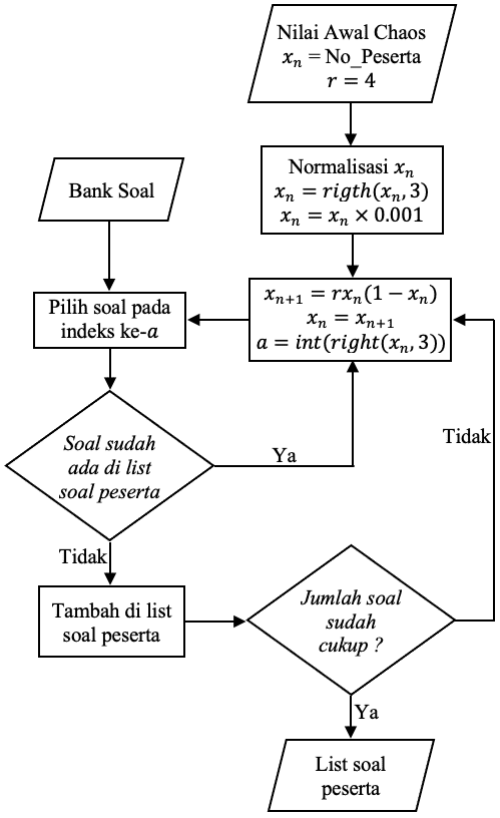
$$x_{n+1} = r x_n (1 - x_n)$$

Di mana:

- x_n adlaah nilai pada iterasi ke-n dengan $0 \leq x_n \leq 1$
- r adalah parameter yang menentukan perilaku sistem, dengan $0 \leq r \leq 4$

Periaku Logistic Map

- Untuk $0 \leq r \leq 1$: Nilai x_n akan menuju nol terlepas dari nilai awal x_0.
- Untuk $1 \leq r \leq 3$: Nilai x_n akan menuju nilai tetap yang bergantung pada r.
- Untuk $3 \leq r \leq 3.57$: Nilai x_n akan berosilasi antara beberapa nilai tetap.
- Untuk $3.57 \leq r \leq 4$: Sistem menunjukkan perilaku chaos, dimana nilai x_n tampak acak dan sangat sensitif terhadap kondisi awal.



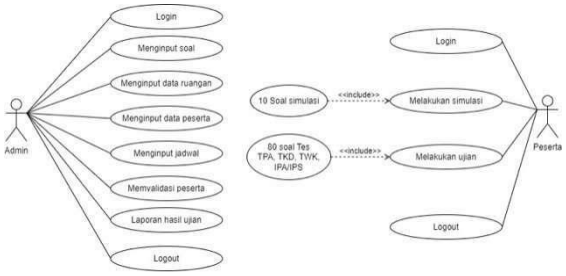
Gambar 2. Algoritma Pengacakan Soal

3. HASIL DAN PEMBAHASAN

3.1. Algoritma Pengacakan Soal

Penelitian ini mengusulkan kombinasi algoritma fisher-yates-shuffle dan chaos map untuk pengacakan soal dan pilihan jawaban penerimaan mahasiswa baru UIN Imam Bonjol Padang. Penambahan chaos map dalam penelitian ini bertujuan untuk meningkatkan hasil pengacakan soal sehingga meningkatkan keamanan soal ujian karena urutan soal dan pilihan jawaban yang teracak akan menyulitkan peserta untuk melakukan tindakan curang. Teknik pengacakan soal dijabarkan secara terperinci pada Gambar 2.

Pada teknik pengacakan, nilai awal (x_n) yang digunakan untuk pengacakan menggunakan algoritma Chaos map adalah nomor peserta ujian, kemudian parameter r yang digunakan adalah 4. Nilai x_n kemudian dinormalisasi agar rentang nilai berada antara 0 sampai dengan 1 ($0 \leq x_n \leq 1$) dan dilakukan perhitungan dengan algoritma chaos map. Nilai chaos map yang dihasilkan kemudian dinormalisasi kembali agar mendapatkan nilai dalam bilangan bulat yaitu dengan mengambil 3 digit terakhir, kemudian nilai tersebut dijadikan untuk pemilihan soal. Proses ini dilakukan sampai dengan jumlah soal sudah terpenuhi.



Gambar 3. *Use Case* Diagram Pengguna

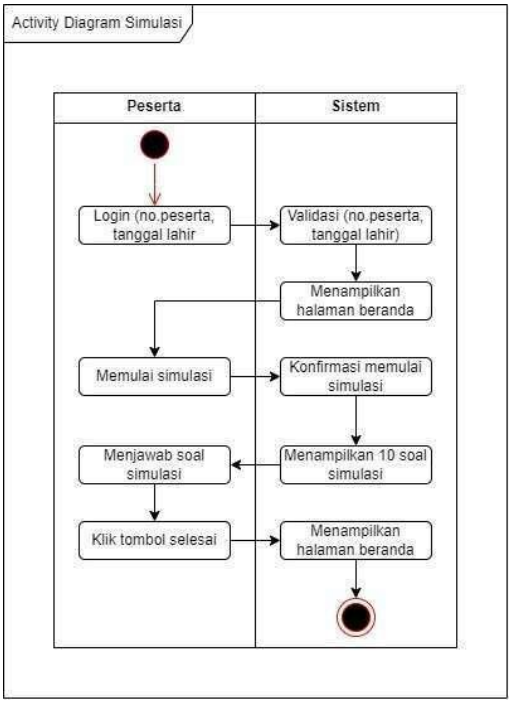
3.2. Perancangan Sistem

a. *Use Case* Diagram

Gambar 3 merupakan *use case* diagram dari aplikasi CAT yang dikembangkan. Terdapat 2 aktor dan 12 *use case* yang saling terhubung. Pada admin terdapat 8 aktifitas yaitu Login, menginput soal, menginput data ruangan, menginput data peserta, menginput jadwal, memvalidasi peserta, membuat laporan hasil ujian, dan logout. Pada peserta terdapat 4 aktifitas yaitu *login*, melakukan simulasi, melakukan ujian, dan *logout*.

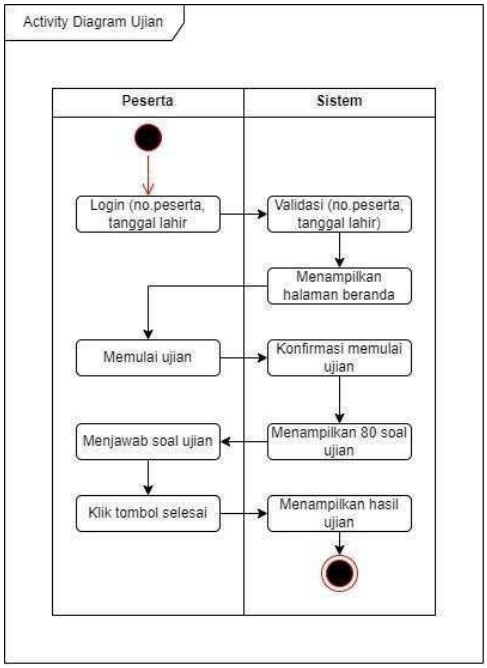
b. *Activity* Diagram

Gambar .4 merupakan *activity* diagram simulasi yang dilakukan oleh peserta ujian, dimana peserta ujian harus melakukan login terlebih dahulu menggunakan No. Peserta dan tanggal lahir, kemudian jika berhasil login maka peserta dapat melakukan semulasi dengan masuk ke halaman simuliasi dan menyelesaikan 10 soal simulasi yang disediakan, jika telah selesai maka peserta akan kembali kehalaman *dashboard*.



Gambar 4. *Activity* Diagram Simulasi

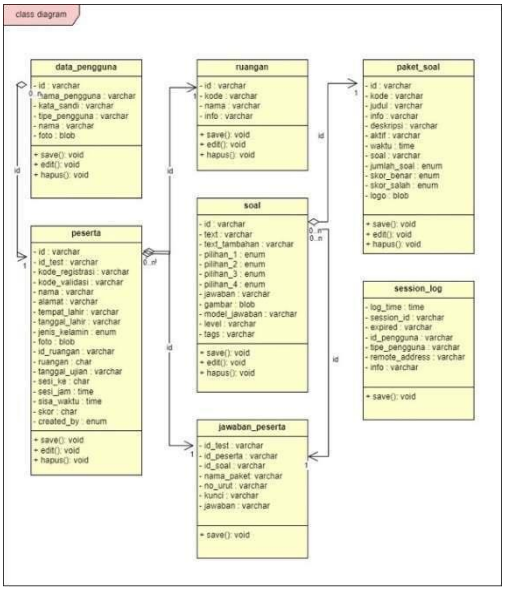
Gambar 5 merupakan *Activity* diagram ujian peserta. Setelah peserta berhasil login maka peserta dapat melakukan ujian setelah adminstrator memvalidasi peserta ujian. Pada halaman ujian, peserta akan menyelesaikan 80 soal ujian, kemudian setelah selesai menjawab semua soal akan tampil skor hasil ujian.



Gambar 5. *Activity* Diagram Ujian

c. Class Diagram

Class diagram untuk sistem *Computer Assisted Test* (CAT) penerimaan mahasiswa baru di UIN Imam Bonjol Padang menggambarkan struktur dari sistem tersebut, termasuk entitas utama dan hubungan antar-entitas. Pada Gambar 6 dapat dilihat terdapat beberapa komponen utama yang ada dalam class diagram untuk sistem CAT diantranya data_pengguna, peserta, ruangan, soal, paket_soal, jawaban_peserta, dan *session_log*.



Gambar 6. Class Diagram

Tabel 1. Hasil Percobaan Menggunakan *array_rand*

SK	Peserta Ujian																				KK	TK	PK
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20			
1	2	1	1	10	7	2	2	7	2	1	4	18	1	1	1	2	2	1	4	9	19	13	0,684
2	3	4	2	11	10	3	14	10	3	3	5	20	5	14	14	8	5	5	6	10	19	10	0,526
3	5	5	6	12	16	6	16	14	10	4	7	22	10	16	18	10	7	10	7	18	19	9	0,474
4	9	6	9	13	18	8	26	16	18	11	8	38	12	21	21	12	15	13	9	29	19	7	0,368
5	14	11	11	19	19	10	28	22	21	13	13	43	14	28	28	14	16	14	12	33	19	8	0,421
6	15	12	14	20	22	12	38	26	22	16	24	45	15	30	35	15	20	24	13	39	19	6	0,316
7	29	13	20	23	26	13	45	30	27	18	25	46	21	42	43	19	21	29	21	40	19	4	0,211
8	31	19	21	24	29	18	50	33	30	29	30	49	24	44	44	20	23	32	26	41	19	4	0,211
9	33	22	24	25	36	25	52	36	40	33	31	54	35	45	50	22	27	35	27	43	19	6	0,316
10	46	25	27	29	42	29	57	39	44	39	38	56	36	46	52	29	30	37	28	48	19	4	0,211
11	48	27	39	35	45	30	58	40	45	47	43	67	40	49	54	30	35	38	29	51	19	4	0,211
12	55	28	43	42	46	38	59	42	46	52	45	68	41	50	55	33	41	39	33	53	19	5	0,263
13	56	46	48	43	47	41	61	46	53	54	47	69	47	52	58	36	46	42	40	54	19	5	0,263
14	58	50	52	46	48	51	68	48	54	55	49	70	48	53	66	37	57	43	41	55	19	3	0,158
15	59	53	53	47	60	54	71	50	60	59	52	72	53	59	71	45	59	45	43	60	19	8	0,421
16	67	55	55	51	64	55	74	53	64	63	60	75	55	60	74	46	66	49	49	64	19	7	0,368
17	70	59	61	54	70	60	75	54	70	64	62	76	56	71	76	48	71	56	57	68	19	5	0,263
18	75	68	62	55	73	63	78	60	73	70	66	77	60	72	83	51	73	57	62	69	19	3	0,158
19	76	71	65	57	77	67	79	67	67	73	68	81	61	74	84	53	81	59	65	70	19	4	0,211
20	81	74	69	60	79	71	82	71	68	75	75	82	65	75	85	74	89	67	71	72	19	6	0,316
21	87	78	71	63	85	86	84	76	71	77	83	83	74	79	87	90	90	69	82	75	19	4	0,211
22	90	85	74	67	86	89	89	83	78	85	85	87	80	80	89	91	91	77	90	78	19	8	0,421
23	92	88	77	75	91	95	90	88	91	87	88	90	92	85	95	96	95	81	95	93	19	8	0,421
24	95	89	87	76	97	96	97	92	93	90	89	93	94	94	97	97	96	82	98	94	19	8	0,421
25	98	97	93	77	100	100	100	98	100	95	99	97	100	98	98	99	99	88	99	97	19	12	0,632
Rata-rata Tingkat kesamaan (0-1)																							0,339

SK = Soal Ke, KK = Kemungkinan Kesamaan, TK = Tingkat Kesamaan, PK = Persentase Kesamaan

Tabel 2. Hasil Percobaan Menggunakan Algoritma yang Diusulkan

SK	Peserta Ujian																				KK	TK	PK
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20			
1	11	7	95	75	47	10	66	13	53	84	7	23	30	29	20	2	77	44	2	53	19	3	0,158
2	91	43	22	100	100	91	91	67	34	55	43	58	9	55	100	100	58	25	100	44	19	9	0,474
3	69	85	96	84	94	96	87	73	97	43	14	91	48	90	4	63	7	92	25	58	19	1	0,053
4	16	97	91	14	77	3	61	30	8	68	91	62	26	44	6	4	29	48	20	66	19	1	0,053
5	95	82	73	99	12	68	70	31	100	86	68	88	59	84	35	67	3	99	27	75	19	2	0,105
6	79	27	42	90	72	18	13	19	12	9	12	57	65	37	76	73	12	69	41	91	19	2	0,105
7	20	2	63	12	92	60	100	25	5	28	65	42	69	98	73	61	47	77	11	85	19	0	0,000
8	88	75	12	49	66	26	45	29	44	64	18	72	53	87	91	58	33	93	58	93	19	1	0,053
9	38	22	75	92	14	8	62	49	36	83	32	26	55	19	33	59	46	19	33	14	19	3	0,158
10	60	57	52	60	29	14	68	93	83	19	78	55	91	3	36	35	65	95	81	76	19	1	0,053
11	14	10	18	52	53	4	57	66	4	90	84	85	43	54	23	3	38	20	34	32	19	1	0,053
12	2	73	48	61	91	11	63	37	56	24	85	90	82	58	29	15	30	49	23	49	19	1	0,053
13	94	46	67	98	3	39	92	36	16	49	62	29	15	45	59	96	61	36	93	7	19	1	0,053
14	28	1	61	54	39	89	39	27	72	54	69	14	95	46	49	38	27	32	38	79	19	4	0,211
15	96	48	27	85	70	90	28	63	20	100	28	68	87	60	71	65	50	8	49	21	19	1	0,053
16	82	19	94	30	75	100	1	87	24	51	19	78	35	89	85	60	70	80	60	45	19	2	0,105
17	48	67	100	26	86	87	10	10	63	88	47	66	83	28	30	8	79	79	1	3	19	2	0,105
18	81	36	16	77	43	59	74	39	32	99	81	76	61	71	94	26	13	41	12	54	19	1	0,053
19	18	25	8	70	48	93	73	62	38	46	79	70	1	26	84	76	100	73	21	90	19	2	0,105
20	13	11	36	94	42	38	46	92	18	66	83	94	71	35	93	93	92	29	16	52	19	3	0,158
21	19	41	29	38	50	67	30	82	48	41	64	84	51	21	40	11	68	68	7	18	19	2	0,105
22	23	5	79	35	10	46	72	94	69	7	73	12	93	100	7	81	78	81	97	68	19	3	0,158
23	61	20	58	46	13	6	56	48	77	32	63	6	52	99	89	80	98	1	77	63	19	3	0,158
24	87	98	65	33	19	57	9	58	68	33	27	3	64	14	56	95	51	28	67	56	19	2	0,105
25	92	24	41	69	96	30	11	71	39	2	8	19	81	50	50	1	85	91	74	62	19	1	0,053
Rata-rata Tingkat kesamaan (0-1)																							0,109

SK = Soal Ke, KK = Kemungkinan Kesamaan, TK = Tingkat, Kesamaan, PK = Persentase Kesamaan

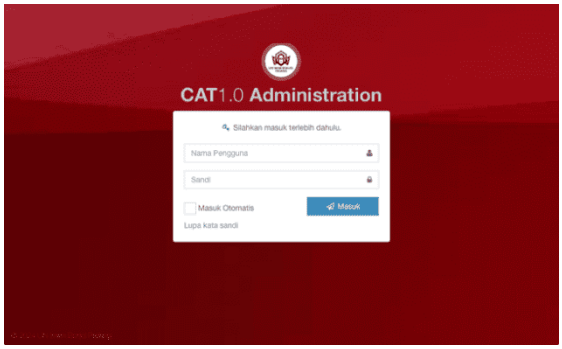
3.3. Pengujian Algoritma

Pengujian dilakukan menggunakan bank soal yang terdiri dari 100 soal, di mana 25 soal akan dipilih secara acak untuk diberikan kepada 20 peserta ujian. Proses pengacakan dilakukan menggunakan algoritma *array_rand* dan algoritma yang diusulkan, kemudian frekuensi kemunculan setiap soal pada masing-masing peserta dihitung guna mengevaluasi distribusi soal yang dihasilkan oleh masing-masing algoritma. Hasil pengacakan soal menggunakan algoritma *array_rand* disajikan pada Tabel 1, sedangkan hasil pengacakan soal menggunakan algoritma yang diusulkan disajikan pada Tabel 2.

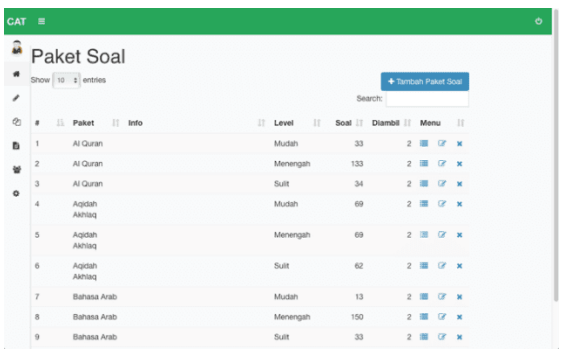
Hasil uji coba menunjukkan bahwa algoritma array_rand pada Tabel 1 menghasilkan tingkat kesamaan soal yang cukup tinggi, dengan rata-rata kemiripan sebesar 33,9% dan kesamaan tertinggi pada soal ke-1 sebesar 68,4%. Sebaliknya, algoritma yang diusulkan pada Tabel 2 memiliki rata-rata kemiripan lebih rendah, yaitu hanya 10,9%, dengan kesamaan tertinggi sebesar 47,4% pada soal ke-2, sementara sebagian besar soal memiliki tingkat kesamaan di bawah 2%. Algoritma yang diusulkan terbukti lebih efektif dalam mengurangi kemiripan soal di antara peserta ujian, sehingga menghasilkan pengacakan soal yang lebih merata dibandingkan array_rand.

3.4. Implementasi Sistem

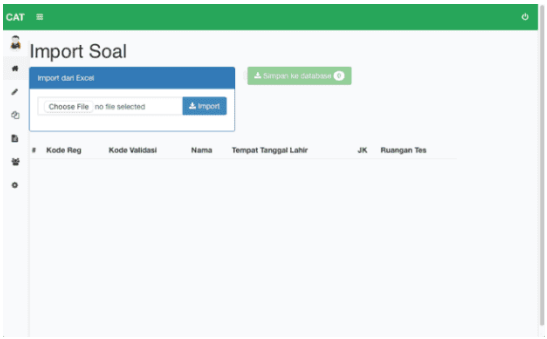
Gambar 7 merupakan halaman login adminstrator. Pada halaman ini admin harus memasukan nama pengguna dan kata sandi, apabila berhasil maka akan tampil halaman *dashboard*.



Gambar 7. Halaman Login Adminstator



Gambar 8. Halaman Peket Soal

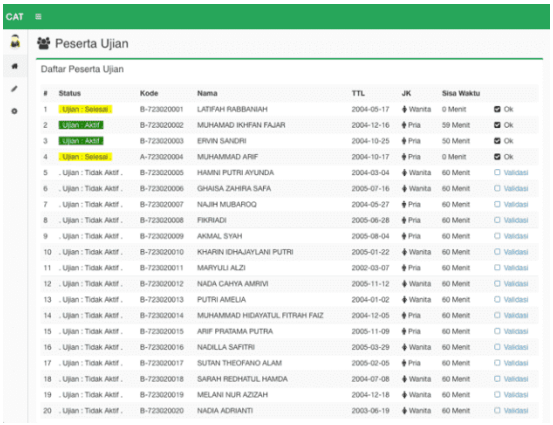


Gambar 9. Halaman Import Soal

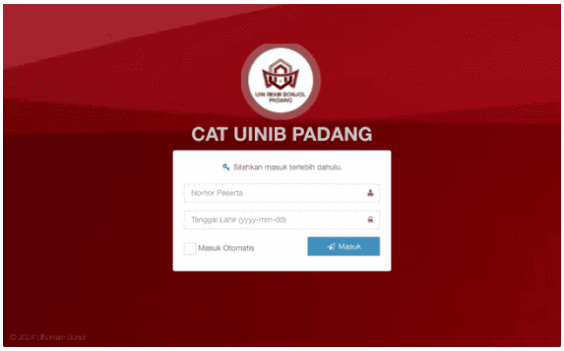
Gambar 8 merupakan halaman paket soal. Pada halaman ini admin dapat mengelola paket soal seperti menimport dan menambah paket soal.

Halaman Import soal seperti pada Gambar 9 merupakan fitur untuk yang dapat digunakan oleh admin untuk mengimport soal yang sudah tersedia dengan format excel sehingga memberikan kemudahan dalam mengelola soal ujian.

Halaman peserta ujian pada gambar 10 merupakan halaman yang dikelola oleh admin untuk melakukan validasi peserta ujian. Validasi dilakukan untuk mengaktifkan ujian peserta, jika sudah tervalidasi maka peserta dapat melakukan ujian. Pada saat validasi ujian, soal peserta ujian akan di alokasikan secara acak untuk masing-masing peserta menggunakan algoritma *fisher yates shuffle* dan *chaos map*, sehingga masing-masing peserta mendapatkan soal yang berbeda.

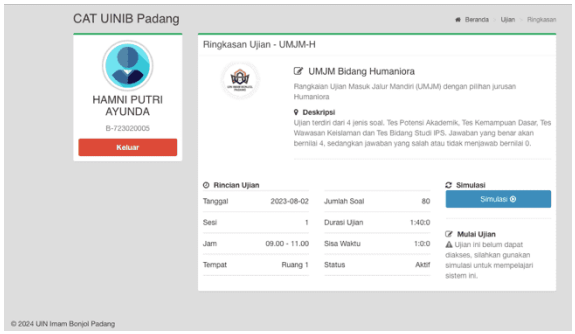


Gambar 10. Halaman Validasi Peserta Ujian



Gambar 11. Halaman Login Peserta Ujian

Gambar 11 merupakan halaman login peserta. Pada halaman ini peserta harus memasukan Nomor Peserta dan tanggal lahir, apabila berhasil maka akan tampil halaman *dashboard.*

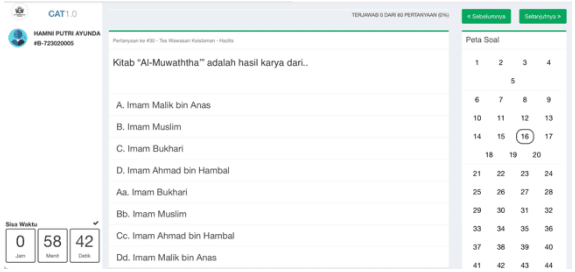


Gambar 12. Halaman Dashboard Peserta Ujian

Gambar 12 merupakan halaman *dashboard* peserta. Pada halaman ini ditampilkan informasi terkait identitas peserta, kategori ujian peserta beserta deskripsinya, dan rincian pelaksanaan ujian. Dihalaman ini juga terdapat dua buah tombol yaitu tombol simulasi dan mulai ujian. Tombol simulasi digunakan untuk masuk kehalaman simulasi. Tombol mulai ujian digunakan untuk masuk ke halaman ujian, tombol ini akan aktif apabila admin telah memvalidasi peserta ujian.

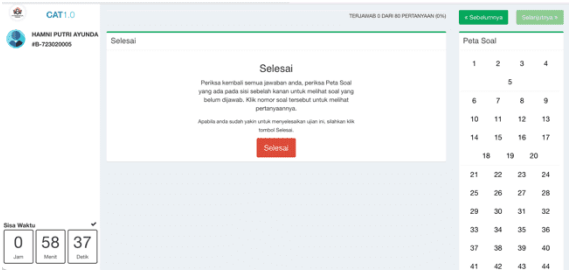
Gambar 13. Halaman Simulasi Peserta Ujian

Halaman simulasi peserta ujian pada Gambar 13 yang dapat digunakan oleh peserta ujian untuk melakukan ujicoba aplikasi. Pada halaman ini terdapat 10 soal simulasi yang dapat dikerjakan oleh peserta ujian, jika telah selesai maka peserta dapat mengklik tombol selesai dan sistem akan kembali ke halaman dahsboard peserta. Pada halaman ini juga terdapat dua tombol yaitu tombol sebelumnya untuk melihat soal sebelumnya dan tombol selanjutnya untuk melihat soal berikutnya.

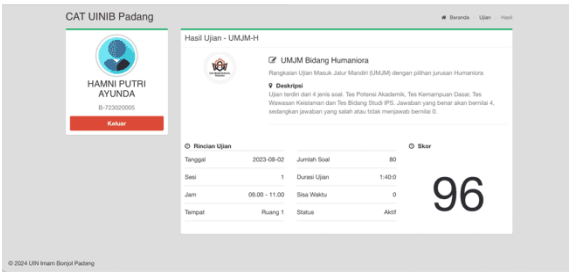


Gambar 14. Halaman Ujian Peserta

Gambar 14 merupakan halaman ujian peserta. Pada halaman ini terdapat beberapa informasi seperti identitas peserta, sisa waktu ujian dan daftar soal ujian. Sisa waktu ujian akan mulai menghitung mundur setelah peserta mengklik tombol mulai ujian. Setiap peserta selesai menjawab soal, maka nomor soal yang terdapat pada kanan layar akan berubah menjadi warna hijau. Dihalamn ini terdapat tiga tombol yaitu yaitu tombol sebelumnya untuk melihat soal sebelumnya, tombol selanjutnya untuk melihat soal berikutnya, dan tombol selesai untuk mengakhiri ujian seperti pada Gambar 15.



Gambar 15. Konfirmasi Peserta Selesai Ujian



Gambar 16. Halaman Skor Peserta Ujian

Gambar 16 merupakan halaman doashboar peserta saat perserta telah selesai melakukan ujian. Saat peserta telah selesai melaksanakan ujian dan ditampilkan skor yang diperoleh, tombol simulasi dan mulai ujian sudah tidak ada lagi.

3.3. Pengujian Sistem

Tujuan dari pengujian sistem ini adalah untuk memastikan bahwa sistem dapat beroperasi dengan baik dan tanpa kesalahan. Pengujian ini menggunakan metode *Blackbox Testing*, yang berfokus pada pengujian fungsionalitas perangkat lunak. Hasil pengujian yang dilakukan menggunakan metode tersebut disajikan pada Tabel 3 untuk bagian administrator dan Tabel 4 untuk bagian peserta.

4. KESIMPULAN

Berdasarkan penelitian yang dilakukan, program CBT yang dikembangkan telah berhasil mendukung pelaksanaan ujian penerimaan mahasiswa baru jalur mandiri di UIN Imam Bonjol Padang. Sistem ini dapat diakses melalui jaringan, sehingga memudahkan pengelola dalam memeriksa hasil ujian dan mengelola database dengan lebih efisien dan aman.

Tabel 3. *Blackbox Testing* Adminstrator

No	Pengujian	Hasil	Ket
1	Membuka Aplikasi	Menampilkan Halaman *Login*	Berhasil
2	*Login*	Menampilkan Halaman *Dashboard*	Berhasil
3	Menu Jadwal Ujian	Menampilkan halaman jadwa ujian	Berhasil
4	Klik Absensi	Menampilkan dan download presensi peserta ujian	Berhasil
5	Klik Berita Acara	Menampikan dan *download* berita acara	Berhasil
6	Klik Tambah jadwal	Menampilkan *form* dan menyimpan jadwal ke *database* jadwal	Berhasil
7	Master Paket Soal	Menampilkan halaman paket soal	Berhasil
8	Klik Tambah Paket Soal	Menampilkan *form* dan menyimpan paket soal ke *database* paket_soal	Berhasil
9	Master Bank Soal	Menampilkan halaman bank soal	Berhasil
10	Submenu Kelola Soal	Menampilkan halaman pengelolaan soal	Berhasil
11	Klik *import* soal	Menampilkan form dan mengimport soal ke *database* soal	Berhasil
12	Klik tambah soal	Menampilkan form dan menyimpan soal ke *database* soal	Berhasil
13	Submenu cadangan dan *import* soal	Menampilkan halaman membuat cadangan data soal dan *import* soal	Berhasil
14	Master Pengguna	Menampilkan halaman pengguna	Berhasil
15	Klik tambah soal	Menampilkan form dan menyimpan data pengguna ke database pengguna	Berhasil
16	Menu Peserta Ujian	Menampilkan daftar peserta ujian	Berhasil
17	Klik validasi ujian	Mengaktifkan ujian peserta dan menggenerati secara acak soal ujian	Berhasil

Aplikasi ini juga memberikan kemudahan bagi peserta ujian, karena tidak memerlukan kertas maupun alat tulis. Sistem ini mampu menangani input data peserta, pemrosesan data, hingga penyajian ujian dengan lebih cepat dibandingkan metode manual, sehingga secara signifikan menghemat waktu.

Pengacakan soal yang menggunakan kombinasi algoritma *Fisher-Yates Shuffle* dan *chaos map* terbukti efektif dalam memberikan soal secara acak kepada setiap peserta, sehingga meminimalisir potensi kecurangan. Hasil pengujian algoritma menunjukkan terjadi penurunan kemiripan soal sebesar 67,86% dalam tingkat kemiripan soal dari 0,339 menjadi 0,109.

Berdasarkan hasil perancangan, implementasi, dan pengujian, dapat disimpulkan bahwa:

1. Sistem yang dibangun mampu melakukan pengacakan soal secara efektif.
2. Sistem ini berfungsi dengan baik dan telah memenuhi ekspektasi berdasarkan hasil pengujian *Blackbox*.

Tabel 4. *Blackbox Testing* Peserta

No	Pengujian	Hasil	Ket
1	Membuka Aplikasi	Menampilkan Halaman *Login*	Berhasil
2	*Login*	Menampilkan Halaman *Dashboard*	Berhasil
3	Klik Simulasi	Menampilkan halaman simulasi	Berhasil
	Klik pertanyaan pertama (simulasi)	Menampilkan soal pertama	Berhasil
4	Klik soal sebelumnya (simlulasi)	Menampilkan soal sebelumnya	Berhasil
5	Klik soal selanjutnya (simulasi)	Menampilkan soal selanjutnya	Berhasil
6	Klik nomor soal simulasi	Menampilkan soal sesuai nomor soal	Berhasil
7	Klik selesai simulasi	Mengakhiri simulasi	Berhasil
8	Klik mulai ujian		Berhasil
9	Klik pertanyaan pertama ujian	Menampilkan soal pertama	Berhasil
10	Klik soal sebelumnya ujian	Menampilkan soal sebelumnya	Berhasil
11	Klik soal selanjutnya ujian	Menampilkan soal selanjutnya	Berhasil
12	Klik nomor soal ujian	Menampilkan soal sesuai nomor soal	Berhasil
13	Klik selesai ujian	Mengakhiri simulasi dan menampilakn skor	Berhasil

5. PENGHARGAAN

Penelitian ini didanai oleh Lembaga Penelitian dan Pengabdian Masyarakat Universitas Islam Negeri Imam Bonjol Padang.

DAFTAR PUSTAKA

AGUSTINA, R., RUKHMANA, T., PITRI, N. AND MEIRISA, S., 2023. *Sistem Pendidikan Digital*. Cendikia Mulia Mandiri.

AKRAM, M., KURNIATI, N. AND SALIM, Y., 2020. Penerapan Algoritma Fisher Yates Shuffle pada Sistem Pembelajaran Tes Online berbasis Aplikasi. *Jurnal Telekomunikasi dan Komputer*, 10(3), p.145. https://doi.org/10.22441/incomtech.v10i3.8753.

ALAROOD, A.A., ALSOLAMI, E., AL-KHASAWNEH, M.A., ABABNEH, N. AND ELMEDANY, W., 2022a. IES: Hyper-chaotic plain image encryption scheme using improved shuffled confusion-diffusion. *Ain Shams Engineering Journal*, 13(3), pp.101–583.

ALAROOD, A.A., ALSOLAMI, E., AL-KHASAWNEH, M.A., ABABNEH, N. AND ELMEDANY, W., 2022b. IES: Hyper-chaotic plain image encryption scheme using improved shuffled confusion-diffusion. *Ain*

Shams Engineering Journal, 13(3). https://doi.org/10.1016/j.asej.2021.09.010.

BAHARI, A., 2021. Computer-assisted language proficiency assessment tools and strategies. *Open Learning: The Journal of Open, Distance and e-Learning*, 36(1), pp.61–87. https://doi.org/10.1080/02680513.2020.1726738.

DUHA, M.W., DEWI, A.R. AND RAHAYU, E., 2022. Online Standard Test System Based on HOTS Using Fisher-Yates Shuffle Algorithm on Final Level Students in Medan State University. *CESS (Journal of Computer Engineering, System and Science)*, 7(2), pp.442–448. https://doi.org/10.24114/cess.v7i2.35657.

FEBRIANI, I., EKAWATI, R., SUPRIADI, U. AND ABDULLAH, M.I., 2021. Fisher-Yates shuffle algorithm for randomization math exam on computer based-test. In: *AIP Conference Proceedings*. AIP Publishing.

GUPTA, R., GUPTA, P. AND SINGH, J., 2019. Security and Cryptography. In: *Software Engineering for Embedded Systems*. Elsevier. pp.501–547.

HAZRA, T.K., GHOSH, R., KUMAR, S., DUTTA, S. AND CHAKRABORTY, A.K., 2015. File encryption using fisher-yates shuffle. In: *2015 International Conference and Workshop on Computing and Communication (IEMCON)*. IEEE. pp.1–7.

LI, C., QIAN, K., HE, S., LI, H. AND FENG, W., 2019a. Dynamics and optimization control of a robust chaotic map. *IEEE Access*, 7, pp.160072–160081.

LI, C., QIAN, K., HE, S., LI, H. AND FENG, W., 2019b. Dynamics and Optimization Control of a Robust Chaotic Map. *IEEE Access*, 7, pp.160072–160081. https://doi.org/10.1109/ACCESS.2019.2950457.

MADOURI, Z.B., HADJ SAID, N. AND ALI PACHA, A., 2022. Image encryption algorithm based on digital filters controlled by 2D robust chaotic map. *Optik*, 264. https://doi.org/10.1016/j.ijleo.2022.169382.

PANDIT, A.A., KUMAR, A. AND MISHRA, A., 2023. Lwr-based quantum-safe pseudo-random number generator. *Journal of Information Security and Applications*, 73, p.103431.

RISNASARI, M., EFFINDI, M.A., DELLIA, P., CAHYANI, L. AND AINI, N., 2021. Computer Based Test Using the Fisher-Yates Shuffle and Smith Waterman Algorithm. *KnE Social Sciences*, pp.353–360.

SAHID, A.D., LESTARI, D., SARI, E.R. AND FAUZAN, M., 2019. Sistem Kriptografi Stream Cipher Berbasis Fungsi Chaos Untuk Keamanan Informasi Stream Cipher Cryptography System Based On Chaotic Function For Information Security.

SAOKANI, U., IRFAN, M., MAYLAWATI, D.S., ABIDIN, R.J. AND TAUFIK, I., 2023. Comparison of the Fisher-Yates Shuffle and the Linear Congruent Algorithm for Randomizing Questions in Nahwu Learning Multimedia. *Khazanah Journal of Religion and Technology*, 1(1), pp.10–14.

SAVITRI, A., 2019. *Revolusi industri 4.0: mengubah tantangan menjadi peluang di era disrupsi 4.0*. Penerbit Genesis.

SUGIHARTONO, T. AND PUTRA, R.R.C., 2021. Penerapan Algoritma Fisher Yates untuk Pengacakan Soal Pada Sistem Ujian Kompetisi Wartawan. *Infotek: Jurnal Informatika dan Teknologi*, 4(2), pp.238–248.

SUWARNA, I.P., 2016. Pengembangan instrumen ujian komprehensif mahasiswa melalui computer based test pada program studi pendidikan fisika.

ZANG, X., IQBAL, S., ZHU, Y., LIU, X. AND ZHAO, J., 2016. Applications of chaotic dynamics in robotics. *International Journal of Advanced Robotic Systems*, 13(2). https://doi.org/10.5772/62796.

Halaman ini sengaja dikosongkan