

BAB 2

TINJAUAN PUSTAKA

Pada bab II ini akan dijelaskan isian dari karya tulis berdasarkan teori yang dipakai untuk menunjang penulisan dalam melakukan penerapan metode LCG (*Linear Congruential Generator*) pada aplikasi ujian online berbasis web.

2.1 Aplikasi

Sebuah aplikasi merujuk pada program komputer atau ponsel yang dirancang untuk menjalankan tugas atau fungsi tertentu, seperti yang disebutkan oleh (Dewi et al., 2021). Menurut (Voutama & Novalia, 2021) mendefinisikan aplikasi sebagai perangkat lunak yang digunakan oleh pengguna untuk melakukan berbagai tugas dalam sistem komputer.

Aplikasi adalah program perangkat lunak yang mencakup fungsionalitas khusus dan menggunakan metode yang dapat diakses oleh semua pengguna. Istilah "aplikasi" berasal dari "Aplikasi," seperti yang dijelaskan dalam Kamus Komputer Eksekutif, yang menunjukkan proses pemecahan masalah menggunakan teknik pemrosesan data perangkat lunak, umumnya melibatkan perhitungan atau pengolahan data yang diinginkan (Rusdiyanto et al., 2022).

2.2 Ujian Online

Ujian online pada dasarnya serupa dengan ujian yang sudah diterapkan di bidang pendidikan, yang mana masih berjalan dengan cara yang konvensional dengan menggunakan kertas. Proses ujian dimulai ketika guru membuat soal yang akan diujikan, kemudian guru juga membuat jawaban dari soal ujian yang sudah dibuat dengan menyertakan kunci jawaban dari soal ujian tersebut. Setelah soal ujian selesai dibuat oleh guru, selanjutnya dilakukan proses ujian dimana soal ujian yang telah dibuat oleh guru akan dibagikan kepada siswa sesuai dengan jadwal ujian beserta kelas dari soal ujian yang telah dibuat. Yang membedakan dengan ujian konvensional yang selama ini telah dilakukan dengan ujian online adalah

penggunaan dari media dan aturan dalam melaksanakan ujian online dimana ujian online, siswa akan mengerjakan soal ujian menggunakan komputer (Saefudin et al., 2020).

2.3 Website

Sebuah situs web merupakan halaman informasi yang dapat diakses melalui internet, memungkinkan akses dari mana saja dan kapan saja, selama terhubung dengan internet. Situs web ini terdiri dari berbagai komponen, seperti teks, gambar, suara, dan animasi, yang menyajikan informasi menarik bagi pengguna untuk dilihat dan dikunjungi (Rochman et al., 2020).

(Andriyan et al., 2020) menjelaskan bahwa situs web adalah rangkaian halaman yang saling terhubung yang menampilkan beragam informasi dalam bentuk teks, gambar diam atau bergerak, animasi, suara, video, atau kombinasi dari elemen-elemen tersebut. Situs web ini dapat memiliki halaman statis atau dinamis, membentuk jaringan terorganisir dengan baik dari struktur-struktur yang saling terhubung.

2.4 *Unified Modeling Language* (UML)

Unified Modelling Language (UML) merupakan sebuah bahasa grafis yang berbentuk meta-model yang digunakan untuk menggambarkan dan merancang sistem perangkat lunak, pemrograman berorientasi objek. Penggunaan UML memungkinkan definisi masalah dilakukan dengan notasi grafis, sehingga mempermudah pemahaman terhadap sistem-sistem kompleks. Penjelasan diagram UML yang digunakan dibawah ini: (Rosa & Shalahuddin, 2019).

Tabel 2. 1. Simbol-simbol *Use Case* Diagram

(Sumber:(Ayu & Permatasari, 2018))

No	Simbol	Keterangan
1	<p data-bbox="459 387 587 421"><i>Use Case</i></p> 	<p data-bbox="699 365 1337 562">Sistem menggunakan kata kerja di awal nama “use case” sebagai indikator fungsionalitasnya, berperan sebagai unit pertukaran pesan antara unit atau aktor</p>
2	<p data-bbox="432 656 608 689">Aktor / <i>Actor</i></p>  <p data-bbox="432 819 600 853">Nama Aktor</p>	<p data-bbox="699 602 1337 799">Aktor merepresentasikan entitas (orang, proses, atau sistem) yang berinteraksi dengan sistem informasi. Nama aktor biasanya berawalan kata benda.</p>
3	<p data-bbox="448 931 592 1021">Asosiasi / <i>Association</i></p> 	<p data-bbox="699 931 1337 1070">Terjadi komunikasi antara aktor dan <i>use case</i> yang terlibat, termasuk yang berinteraksi dengan aktor.</p>
4	<p data-bbox="419 1111 632 1144">Ekstensi/<i>Extend</i></p>  <p data-bbox="443 1249 603 1283"><<extend>></p>	<p data-bbox="699 1104 1337 1355"><i>Use case</i> tambahan dapat berfungsi secara independen tanpa <i>use case</i> asli. Ini menyerupai konsep pewarisan dalam pemrograman berorientasi objek dan biasanya menggunakan nama awal yang sama dengan <i>use case</i> asli</p>
5	<p data-bbox="419 1379 616 1469">Generalisasi / <i>Generalization</i></p> 	<p data-bbox="699 1379 1337 1518">Dua <i>use case</i> dapat terhubung melalui hubungan generalisasi dan spesialisasi, di mana satu <i>use case</i> lebih umum dibandingkan yang lain</p>
6	<p data-bbox="419 1641 632 1765">Menggunakan / <i>Include / uses</i></p>  <p data-bbox="443 1839 619 1872"><<include>></p>	<p data-bbox="699 1574 1337 1771">Penambahan <i>use case</i> bergantung pada <i>use case</i> aslinya, memerlukan eksekusi <i>use case</i> asli. Hubungan “<i>include</i>” memastikan <i>use case</i> tambahan selalu dipanggil bersama <i>use case</i> asli.</p>

2.4.1 Activity Diagram

Diagram aktivitas menggambarkan beragam urutan aktivitas dalam sistem yang sedang dirancang, termasuk titik awal, titik keputusan yang mungkin terjadi, dan bagaimana setiap urutan berakhir. Penjelasan tentang simbol-simbol yang digunakan dalam diagram aktivitas terdapat pada tabel 2.2.

Tabel 2. 2. Simbol Activity Diagram

(Sumber:(Ayu & Permatasari, 2018))

No	Gambar	Keterangan
1.	 Activity	Kegiatan yang dilakukan oleh sistem umumnya dimulai dengan menggunakan kata kerja
2.	 Decision	Asosiasi cabang terjadi ketika terdapat lebih dari satu pilihan kegiatan yang bisa dipilih
3.	 Status Awal	Diagram aktivitas memiliki kondisi awal untuk menunjukkan titik awal dari kegiatan sistem
4.	 Status Akhir	Diagram aktivitas juga memiliki kondisi akhir untuk menandakan akhir dari kegiatan sistem
5.	 Join	Asosiasi penggabungan terjadi ketika beberapa kegiatan digabung menjadi satu kesatuan

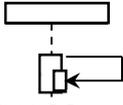
2.4.2 Sequence Diagram

Diagram urutan atau *Sequence Diagram* menggambarkan bagaimana objek-objek dalam sistem berinteraksi dengan objek-objek di sekitarnya, termasuk pengguna dan tampilan, melalui pertukaran pesan yang dijelaskan sepanjang waktu.

Diagram urutan memiliki dua dimensi, yaitu dimensi vertikal yang mencerminkan urutan waktu, dan dimensi horizontal yang mencerminkan hubungan antara objek-objek tersebut. Setiap pesan yang dikirimkan dapat menimbulkan respons atau balikan yang sesuai dengan skenario yang telah dirancang dalam diagram use case. Interaksi yang terjadi dapat berupa instansiasi objek atau pemanggilan metode statis objek dari suatu kelas. Berikut ini adalah daftar simbol yang digunakan dalam sequence diagram yang dijelaskan pada tabel 2.3.

Tabel 2. 3. Simbol *Sequence Diagram*

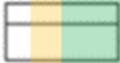
(Sumber:(Ayu & Permatasari, 2018))

No	Gambar	Keterangan
1.	 Aktor	Aktor dalam sistem adalah pengguna, yang bisa berupa manusia, mesin, atau sistem lain dari berbagai model yang berinteraksi dengan sistem dari batas sistem
2.	 <i>Lifeline</i>	Peserta individual dalam interaksi adalah jalur kehidupan yang tidak dapat memiliki ketergantungan ganda.
3.	 <i>Message</i>	Diagram urutan menggambarkan aliran informasi atau kontrol transaksi antara elemen-elemen
4.	 <i>Self-Message</i>	Diagram urutan mencerminkan proses baru atau pemanggilan operasi metode jalur <i>lifeline</i>

2.4.3 Class Diagram

Diagram kelas menggambarkan keterhubungan antara objek-objek yang terlibat dalam sistem, selain itu, diagram kelas juga dapat menampilkan operasi dan properti yang ada di dalam sebuah objek. Tabel 2.4 berisi penjelasan mengenai simbol-simbol yang digunakan dalam diagram kelas.

Tabel 2. 4. Simbol *Class Diagram*
(Sumber:(Ayu & Permatasari, 2018))

Gambar	Keterangan
 <i>Generalization</i>	Suatu relasi dimana objek anak mengadopsi perilaku dan struktur data dari objek yang menjadi induknya
 <i>Navy Association</i>	Berusaha untuk tidak mengaitkan lebih dari 2 objek
 <i>Class</i>	Sebuah kumpulan objek yang memiliki atribut dan operasi yang serupa
 <i>Collaboration</i>	Penjelasan mengenai urutan tindakan yang dilakukan oleh sistem yang menghasilkan hasil yang dapat diukur untuk seorang aktor.
 <i>Realization</i>	Tindakan yang benar-benar dilakukan oleh sebuah objek
 <i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada elemen independen akan mempengaruhi elemen-elemen yang bergantung padanya yang tidak independen

 <i>Association</i>	Suatu koneksi yang menghubungkan satu objek dengan objek lainnya
---	--

2.5 *Linear Congruential Generator (LCG)*

Algoritma *Linear Congruential Generator (LCG)* merupakan suatu pendekatan untuk menghasilkan berbagai angka dengan cara-cara beragam. Angka-angka acak dapat diperoleh melalui berbagai metode, seperti melalui pelemparan dadu, pembacaan tabel angka acak, serta teknik-teknik lainnya. Angka-angka acak yang dihasilkan dari pendekatan ini dikenal sebagai angka acak semu atau *pseudo-random* (Amarta.S et al., 2021).

LCG adalah salah satu algoritma yang umum digunakan untuk *Pseudo Random Number Generator (PRNG)*. Teori dari algoritma ini mudah dipahami dan dapat diterapkan dengan efisien untuk menghasilkan serangkaian nilai acak atau mengacak sekelompok nilai. Keuntungan utama dari LCG adalah kecepatan operasinya yang tinggi. Metode LCG digunakan untuk menghasilkan angka-angka acak dengan distribusi yang merata. Angka acak, juga dikenal sebagai *pseudo-acak*, dihasilkan melalui proses di mana hasilnya tidak dapat diprediksi, dan angka yang sama tidak dapat dihasilkan secara berurutan. Angka-angka acak yang dihasilkan oleh LCG digunakan untuk menentukan posisi angka 1 hingga n dalam setiap baris (Fanani, 2021).

LCG adalah *generator* angka acak yang sederhana dan mudah dipahami, baik dari segi teori maupun implementasinya. LCG didefinisikan dalam hubungan iteratif berikut: (Amarta.S et al., 2021).

$$X_n = (aX_{n-1} + b) \bmod m$$

Keterangan:

X_n = bilangan acak ke-n dari deretnya

X_{n-1} = bilangan acak sebelumnya

a = faktor pengali

b = increment

m = modulus LCG memiliki periode penuh jika:

1. b relatif prima terhadap m.
2. $a - 1$ dapat dibagi dengan setiap faktor prima dari m.
3. $a - 1$ adalah kelipatan 4 jika m kelipatan 4. X_0 adalah kunci pembangkit atau disebut juga umpan (seed).

2.6 Agile

Metode *Agile* merupakan suatu pendekatan dalam pengembangan perangkat lunak yang sangat efektif dan fleksibel. Metode ini tidak memberikan prosedur terperinci untuk menciptakan jenis model tertentu, walaupun terdapat cara-cara untuk menjadi *modeler* yang terampil (Larasati et al., 2021).

Agile Modelling adalah sebuah pendekatan yang digunakan untuk mendokumentasikan dan membuat model sistem perangkat lunak yang lebih praktis. Metode ini mencakup kumpulan nilai, prinsip, dan praktik yang dapat digunakan secara efisien dalam proyek pengembangan perangkat lunak. Metode ini mengandung beberapa langkah yang meliputi kebutuhan, perancangan, pembuatan, pengujian, penerapan, dan evaluasi. Keterangan tentang langkah-langkah pendekatan agile ini terlihat dalam ilustrasi yang ditunjukkan dalam Gambar 2.1.



Gambar 2. 1 Tahapan Agile Modelling

Penjelasan gambar diatas yaitu sebagai berikut:

1. *Requirements*

Tahap awal dalam membangun sistem ini melibatkan pengumpulan data mengenai keputusan pengguna dan persyaratan sistem. Data ini menjadi dasar untuk menciptakan aplikasi ujian online. Metode observasi, wawancara, dan studi literatur digunakan untuk mengumpulkan data yang diperlukan.

2. *Design*

Selanjutnya, peneliti akan mempelajari tahap-tahap sebelumnya dan merancang sistem yang akan dibuat. Proses desain dimulai dengan membuat alur sistem menggunakan UML (Diagram Use Case, Diagram Aktivitas, Diagram Urutan), serta merancang antarmuka pengguna untuk setiap halaman.

3. *Development*

Setelah itu, tahap pengembangan akan mengubah desain menjadi sistem canggih yang siap digunakan. Implementasi desain yang telah dibuat sebelumnya dilakukan untuk mendukung pembentukan sistem. Program akan dikodekan menggunakan bahasa pemrograman PHP dengan database MySQL.

4. *Testing*

Setelah sistem dibuat, dilakukan pengujian terlebih dahulu sebelum digunakan untuk mengetahui apakah masih terdapat malfungsi pada sistem agar

selanjutnya dapat diperbaiki oleh pengembang sebelum sampai ke pengguna. Blackbox Testing digunakan untuk pengujian sistem ini untuk menilai fungsionalitas dari sistem.

5. *Deployment*

Selanjutnya, tahap implementasi memungkinkan pengguna untuk menggunakan sistem yang telah dibuat. Selama tahap ini, pengembang aplikasi memberikan pelatihan kepada pengguna untuk penggunaan sistem. Umpan balik dari pengguna dikumpulkan untuk mengevaluasi kinerja sistem. Tanggapan ini berkontribusi dalam menilai nilai aplikasi yang telah dikembangkan.

6. *Review*

Tahap ini dilakukan setelah menerima hasil dari tahap-tahap sebelumnya dan melibatkan peninjauan ulang sistem berdasarkan tanggapan yang diperoleh. Ini berfungsi sebagai platform untuk pengembangan lebih lanjut dan evaluasi sistem secara menyeluruh.

2.7 *Blackbox Testing*

Blackbox Testing adalah pendekatan umum yang sering digunakan untuk mengevaluasi cara kerja internal aplikasi tanpa memerlukan pengetahuan pemrograman. Pendekatan ini sangat bermanfaat untuk pengujian fungsional, di mana input dan output dari aplikasi dievaluasi. Teknik pengujian ini ditujukan untuk para pengujian yang mungkin tidak memiliki pemahaman mendalam tentang pemrograman. Fokus utama dari pengujian *Blackbox Testing* adalah memahami input yang diberikan ke sistem, perilaku eksternal yang diharapkan, dan hasil aktual yang dihasilkan berdasarkan input yang diberikan ke program (Utomo et al., 2018).

Dengan menggunakan metode *blackbox*, dapat menghasilkan serangkaian kasus uji dengan kriteria berikut (Fitriyanto et al., 2022):

1. Kasus uji ini mengurangi jumlah kasus uji tambahan yang diperlukan untuk melakukan pengujian yang dapat diandalkan dengan lebih efisien.

2. Kasus uji ini mencerminkan keberadaan atau ketiadaan kategori kesalahan tertentu, bukan hanya terbatas pada kesalahan yang terkait dengan uji tertentu.

2.8 User Acceptance Testing (UAT)

UAT adalah suatu praktik umum, dilakukan untuk menilai sistem dengan kebutuhan dari pengguna. Pengujian ini memvalidasi sistem yang dikembangkan dari segi fungsionalitas dan tampilan antarmuka pengguna (Al Fahmi 2019). Pengujian *Acceptance Testing* dilakukan dengan menggunakan data, dan skenario yang mirip atau hampir sama dengan sistem yang akan beroperasi, dengan fokus pada pengujian kasus penggunaan produk tertentu (Fahrizal, Fikry and Ramadhani 2023). Untuk menentukan nilai UAT, lebih mudah dipahami dengan persamaan berikut (Mariyani, Setyaningsih and Agustina 2022):

$$\text{Nilai UAT} = \left(\frac{\text{Jumlah jawaban} \times \text{Bobot Jawaban}}{\text{Jumlah responden}} \right) \quad (2.1)$$

Langkah selanjutnya adalah membagi nilai UAT dengan skor ideal kemudian mengalikannya dengan seratus persen untuk mendapatkan nilai persentase untuk melihat keuntungan dari sistem koreksi otomatis.

$$\text{Persentase UAT} = \frac{\text{Total skor}}{x} \times 100 \% \quad (2.2)$$

Keterangan:

Total skor = Total skor keseluruhan jawaban responden

x = Skor tertinggi x (total pertanyaan x jumlah responden)

Tabel 2. 5 Kategori Kelayakan

No	Skor	Kategori Kelayakan
1	< 21%	Sangat Tidak Layak
2	21 – 40 %	Tidak Layak
3	41 – 60 %	Cukup Tidak Layak

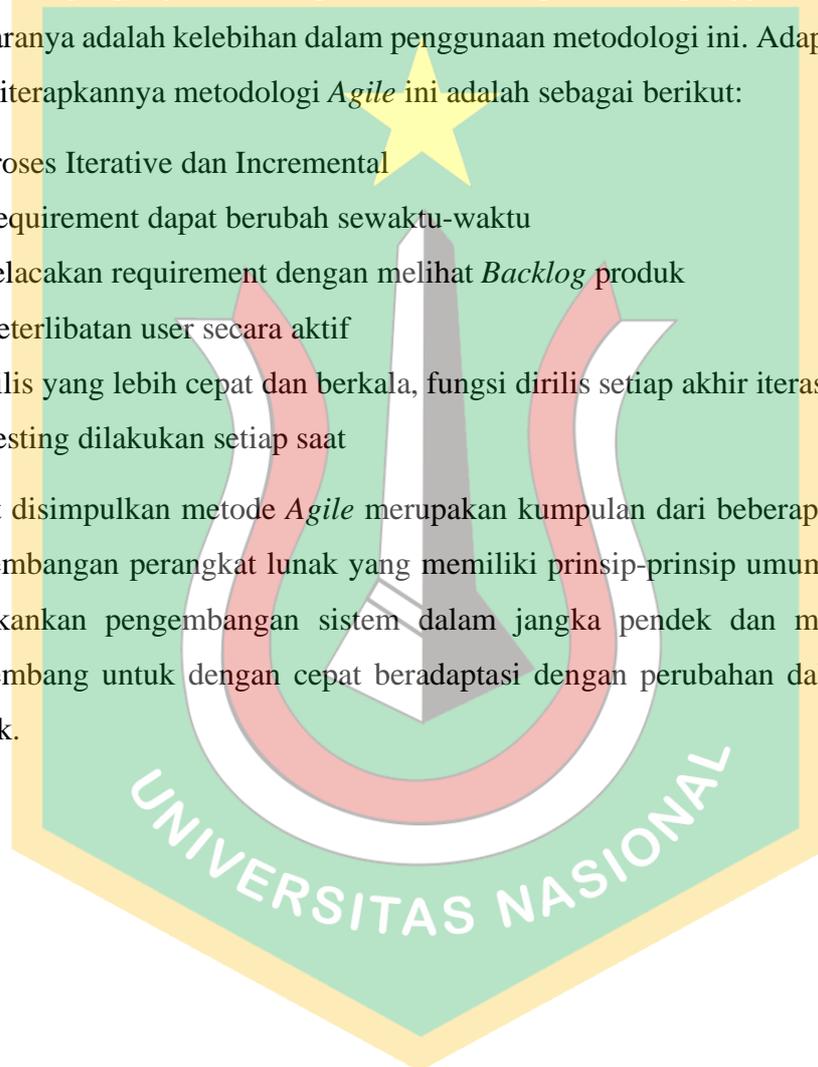
4	60 – 80 %	Layak
5	81 – 100 %	Sangat Layak

2.9 Alasan Pemilihan Teori, Kerangka Kerja, atau Mekanisme

Sebagai penguat alasan penelitian dalam penentuan penggunaan aplikasi ini diantaranya adalah kelebihan dalam penggunaan metodologi ini. Adapun kelebihan dari diterapkannya metodologi *Agile* ini adalah sebagai berikut:

1. Proses Iterative dan Incremental
2. Requirement dapat berubah sewaktu-waktu
3. Pelacakan requirement dengan melihat *Backlog* produk
4. Keterlibatan user secara aktif
5. Rilis yang lebih cepat dan berkala, fungsi dirilis setiap akhir iterasi
6. Testing dilakukan setiap saat

Dapat disimpulkan metode *Agile* merupakan kumpulan dari beberapa metodologi pengembangan perangkat lunak yang memiliki prinsip-prinsip umum. Metode ini menekankan pengembangan sistem dalam jangka pendek dan menuntut para pengembang untuk dengan cepat beradaptasi dengan perubahan dalam berbagai bentuk.



2.10 Penelitian Terdahulu

Tabel 2. 6 Pembandingan Jurnal

No.	Author	Judul	Metode	Kelebihan	Kekurangan
1	(Amalia et al., 2021)	Analisis Metode Cosine Similarity Pada Aplikasi Ujian Online Esai Otomatis (Studi Kasus JTI Polinema)	Cosine Similarity	Cosine Similarity mudah diimplementasikan dan efisien dalam perhitungannya. Metode ini cukup akurat dalam membandingkan kemiripan antara jawaban peserta dengan kunci jawaban.	Hasil dari Cosine Similarity sangat tergantung pada representasi vektor yang digunakan. Metode ini tidak mempertimbangkan urutan dan konteks jawaban yang penting dalam menentukan jawaban yang benar. Metode ini juga hanya cocok untuk jawaban yang dapat direpresentasikan dalam vektor, sehingga tidak cocok untuk jawaban yang lebih kompleks atau abstrak.
2	(Siregar et al., 2021)	Aplikasi Ujian Online Untuk Siswa Sekolah Menengah Pertama Dengan	Rational Unified Process (RUP)	Metodeses ini fleksibel, fokus pada kualitas, mendorong kolaborasi tim pengembang dan pengguna, serta menyediakan	Kompleksitas yang tinggi, memerlukan sumber daya yang cukup, terlalu formal dan kurang

		Menggunakan Metode Rational Unified Process (RUP)		dokumentasi yang lengkap.	sesuai untuk proyek kecil.
3	(Yulianto et al., 2020)	Pembangunan Aplikasi Ujian Online Menggunakan Akses Token & Algoritma <i>Simple Random Sampling</i>	<i>Simple Random Sampling</i>	Metode ini merupakan metode pemilihan acak yang mudah diimplementasikan dan dapat menghasilkan sampel yang representatif dari populasi yang lebih besar. Metode ini dapat mengurangi bias dalam pemilihan sampel dan dapat membantu meningkatkan validitas dan reliabilitas data.	Metode ini mungkin tidak cocok untuk digunakan pada populasi dengan ukuran yang sangat besar, karena membutuhkan waktu dan sumber daya yang lebih banyak dalam melakukan pemilihan sampel secara acak.
4	(Haryono et al., 2021)	Rancang Bangun Sistem Ujian Online Menggunakan Algoritma <i>Cosine Similarity</i>	<i>Cosine Similarity</i>	Metode ini tidak terpengaruh pada panjang pendeknya suatu dokumen dan memiliki tingkat akurasi yang tinggi. Algoritma ini juga dipilih	Metode ini tidak mempertimbangkan urutan dan konteks jawaban yang bisa penting dalam menentukan jawaban yang benar. Metode ini juga hanya

		Berbasis Web		karena pembangunan sistem yang lebih cepat dan sangat fleksibel dengan perubahan yang terjadi dengan proses pembangunan perangkat lunak.	cocok untuk jawaban yang dapat direpresentasikan dalam vektor, sehingga tidak cocok untuk jawaban yang lebih kompleks atau abstrak.
5	(Andesa et al., 2020)	Sistem Ujian Online Menggunakan Algoritma Fisher Yates Shuffle	<i>Fisher Yates Shuffle</i>	Merupakan metode pengacakan yang lebih baik atau bisa dikatakan cocok untuk mengacak angka, waktu pengerjaannya cepat, dan tidak memakan banyak waktu untuk pengacakannya.	Pengacakan yang dihasilkan tidak sepenuhnya acak jika pengacakan dilakukan pada populasi yang sangat besar atau jika pengacakan dilakukan secara terus-menerus pada populasi yang sama. Selain itu, penggunaan metode Fisher-Yates shuffle juga dapat mempengaruhi waktu respons pengguna karena harus melakukan pengacakan terlebih dahulu sebelum jawaban muncul.

6	(Prasetyadi et al., 2020)	Implementasi Metode <i>Multiplicative Random Number Generator (MRNG)</i> Pada Aplikasi Ujian Sekolah Berbasis Komputer	<i>Multiplicative Random Number Generator (MRNG)</i>	Menghasilkan bilangan acak yang cepat dengan menggunakan metode pengulangan sederhana. Selain itu, metode MRNG juga mudah diimplementasikan pada berbagai platform dan perangkat keras.	Menghasilkan pola pengulangan bilangan acak yang dapat membahayakan keamanan aplikasi dan ketidakmampuan untuk menghasilkan bilangan acak dengan distribusi yang sama dengan distribusi asli. Selain itu, metode MRNG juga dapat menghasilkan bilangan acak yang tidak sepenuhnya acak atau dapat diprediksi jika pengaturan awal atau variabel yang digunakan tidak dipilih dengan benar
7	(Fanani, 2021)	Pengacakan Soal Pada Sistem <i>Computer Based Test (CBT)</i> dengan Metode <i>Linear Congruential Generator (LCG)</i> di	<i>Linear Congruential Generator</i>	Operasinya yang sangat cepat. Metode LCG adalah metode yang digunakan untuk membangkitkan bilangan acak dengan distribusi uniform.	Kemampuannya dalam menghasilkan bilangan acak yang tidak sepenuhnya acak, karena metode ini menghasilkan bilangan acak dengan pola yang dapat diprediksi. Hal

		SMA Negeri Jogoroto			ini dapat menimbulkan risiko keamanan pada aplikasi yang membutuhkan bilangan acak dengan tingkat keamanan yang lebih tinggi, seperti pada aplikasi keuangan atau keamanan data.
8	(Nainggolan, 2020)	Aplikasi Sistem Ujian Online Pada <i>Local Area Network</i> Dengan Metode Kriptografi <i>Government Standard</i>	<i>Kriptografi Government Standard</i>	Kelebihan dari metode GOST adalah kecepatannya yang cukup baik, walaupun tidak secepat Blowfish tetapi lebih cepat dari IDEA.	Algoritma yang relatif lambat dan kompleks dalam memproses data, dan kurangnya fleksibilitas dalam mengubah parameter kunci atau struktur algoritma jika ditemukan kelemahan atau kelemahan baru. Kelemahan ini dapat mempengaruhi kecepatan dan kinerja aplikasi ujian online, serta keamanan data pengguna dan kunci enkripsi.

9	(Akram et al., 2020)	Penerapan Algoritma <i>Fisher Yates Shuffle</i> pada Sistem Pembelajaran Tes Online Berbasis Aplikasi	<i>Fisher Yates Shuffle</i>	Merupakan metode pengacakan yang lebih baik atau dapat dianggap cocok untuk pengacakan angka, dengan waktu pelaksanaan yang cepat dan tidak memakan banyak waktu untuk melakukan pengacakan.	Pengacakan yang dihasilkan tidak sepenuhnya acak jika pengacakan dilakukan pada populasi yang sangat besar atau jika pengacakan dilakukan secara terus-menerus pada populasi yang sama. Selain itu, penggunaan metode Fisher-Yates shuffle juga dapat mempengaruhi waktu respons pengguna karena harus melakukan pengacakan terlebih dahulu sebelum jawaban muncul.
10	(Prakarsa et al., 2020)	Model Pengacakan Soal Ujian Online SMA Menggunakan Metode <i>Linear Congruential Generator</i> dan <i>Fisher Yates</i>	<i>Linear Congruential Generator</i> dan <i>Fisher Yates</i>	Metode LCG dapat menghasilkan bilangan acak dengan cepat dan efisien, sementara metode Fisher Yates dapat mengacak urutan soal secara acak dan merata, sehingga setiap peserta ujian	Metode ini menghasilkan bilangan acak yang tidak sepenuhnya acak, karena metode ini menghasilkan bilangan acak dengan pola yang dapat diprediksi. Selain itu, metode pengacakan ini

				<p>mendapatkan urutan soal yang berbeda-beda. Kombinasi kedua metode ini dapat menghasilkan urutan soal yang acak dan merata secara efisien.</p>	<p>juga kurang efektif jika digunakan pada jumlah soal yang sangat besar, karena dapat menghabiskan banyak sumber daya komputasi dan memperlambat kinerja aplikasi.</p>
--	--	--	--	--	---

Berdasarkan 10 penelitian terdahulu, disimpulkan bahwa terdapat berbagai metode pengacakan soal ujian online seperti *Cosine Similarity*, RUP, *Simple Random Sampling*, *Fisher Yates Shuffle*, dan LCG. Metode pengacakan seperti *Fisher Yates Shuffle* dan LCG dapat mengurangi kecurangan siswa dengan memberikan soal secara acak. Implementasi aplikasi ujian online berbasis web dengan pengacakan soal berhasil meningkatkan efisiensi, efektivitas, dan memberikan penilaian otomatis pada ujian esai. Beberapa penelitian mencatat tingkat akurasi yang menjanjikan dari metode pengacakan seperti *Cosine Similarity* dan *Fisher Yates Shuffle*. Penggunaan LCG dan *Fisher Yates Shuffle* dalam ujian online efektif mengatasi masalah nilai periodik. Secara keseluruhan, penerapan metode pengacakan seperti LCG dan *Fisher Yates Shuffle* pada ujian online dapat menciptakan proses ujian yang lebih adil, efisien, dan efektif dalam mengukur kemampuan siswa serta meningkatkan pengelolaan nilai.

Jadi, pada penelitian ini, algoritma yang akan digunakan adalah *Linear Congruential Generator (LCG)*.