

Implementasi Algoritma *Linear Congruent Method* Pada Aplikasi Belajar Mandiri Siswa

Taqrim Ibadi^{1*}, Rudi Krisulta²
Universitas Bina Darma^{1,2}

Jalan Jenderal Ahmad Yani No.3 Palembang
Sur-el : taqrimibadi@binadarma.ac.id¹, rudi12krisulta@gmail.com²

*) Corresponden Author

DOI : <https://doi.org/10.33557/jurnalatrik.v26i2.3328>

Abstract : This research aims to implement the Linear Congruent Method (LCM) algorithm in the application of student self-learning in the INET Computer Course Institute. The LCM algorithm is used to generate random numbers required in learning features, such as practice questions and random tests. This study evaluates the effectiveness of the algorithm in improving the quality of self-learning through applications. The research method involves application development, and algorithm integration. The results of the study show that the implementation of the LCM Algorithm has succeeded in improving the variety and quality of the learning content presented. The app has proven to be efficient in providing random materials, which supports student interaction and improves their understanding of the material. This research contributes to the development of more effective and adaptive educational applications, as well as opens up opportunities for further research in the field of educational algorithms and technology.

Keywords: LCM, Self-Learning Application, Educational Technology

Abstrak : Penelitian ini bertujuan untuk mengimplementasikan Algoritma Linear Congruent Method (LCM) pada aplikasi belajar mandiri siswa di Lembaga Kursus INET Komputer. Algoritma LCM digunakan untuk menghasilkan angka acak yang diperlukan dalam fitur pembelajaran, seperti soal latihan dan tes acak. Studi ini mengevaluasi efektivitas algoritma dalam meningkatkan kualitas pembelajaran mandiri melalui aplikasi. Metode penelitian melibatkan pengembangan aplikasi, dan integrasi algoritma. Hasil penelitian menunjukkan bahwa implementasi Algoritma LCM berhasil meningkatkan variasi dan kualitas konten pembelajaran yang disajikan. Aplikasi ini terbukti efisien dalam penyediaan materi acak, yang mendukung interaksi siswa dan meningkatkan pemahaman mereka terhadap materi. Penelitian ini memberikan kontribusi pada pengembangan aplikasi pendidikan yang lebih efektif dan adaptif, serta membuka peluang untuk penelitian lanjutan di bidang algoritma dan teknologi pendidikan.

Kata kunci: LCM, Aplikasi Belajar Mandiri, Teknologi Pendidikan

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah membawa perubahan signifikan dalam bidang pendidikan. Penggunaan aplikasi belajar mandiri menjadi salah satu inovasi yang semakin populer dalam mendukung proses pembelajaran yang lebih fleksibel dan personal. Penggunaan media pembelajaran yang melibatkan siswa secara aktif dalam proses belajar telah terbukti dapat meningkatkan

pemahaman dan motivasi siswa [1]. Media pembelajaran dapat dibuat dan dirancang sesuai dengan perkembangan teknologi saat ini [2]. Aplikasi ini memungkinkan siswa untuk belajar secara mandiri sesuai dengan kebutuhan dan kemampuan mereka, tanpa batasan waktu dan tempat. Dalam konteks ini, algoritma untuk mengelola dan menyajikan konten secara dinamis dan acak sangat dibutuhkan untuk meningkatkan efektivitas dan kualitas pembelajaran.

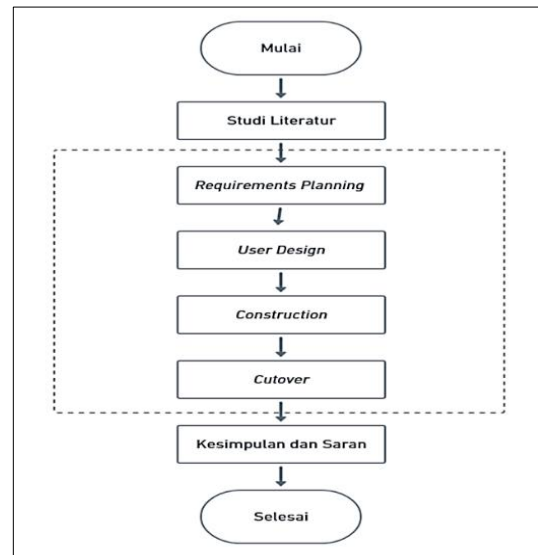
Algoritma Linear Congruent Method (LCM) adalah salah satu algoritma pembangkit bilangan acak yang sering digunakan dalam berbagai aplikasi, termasuk aplikasi pembelajaran. Algoritma ini memiliki keunggulan dalam hal kesederhanaan implementasi dan efisiensi dalam menghasilkan angka acak yang diperlukan untuk variasi konten belajar. Dengan menerapkan LCM pada aplikasi belajar mandiri, diharapkan aplikasi ini dapat menyediakan konten yang lebih bervariasi dan adaptif, sehingga meningkatkan keterlibatan dan hasil belajar siswa.

Penelitian ini bertujuan untuk mengimplementasikan Algoritma LCM pada aplikasi belajar mandiri siswa di Lembaga Kursus dan Pelatihan (LKP) INET Komputer. Studi ini akan mengevaluasi efektivitas penerapan algoritma tersebut dalam meningkatkan kualitas pembelajaran dan mengoptimalkan pengalaman belajar siswa. Melalui penelitian ini, diharapkan dapat diperoleh wawasan mengenai penerapan algoritma dalam aplikasi pendidikan serta memberikan kontribusi pada pengembangan teknologi pendidikan.

Dengan demikian, penelitian ini tidak hanya berfokus pada aspek teknis implementasi algoritma, tetapi juga pada dampaknya terhadap kualitas pembelajaran siswa. Hal ini diharapkan dapat membuka peluang untuk eksplorasi lebih lanjut tentang bagaimana teknologi dapat diintegrasikan secara efektif ke dalam sistem pendidikan untuk mendukung proses pembelajaran yang lebih baik.

2. METODOLOGI PENELITIAN

Berdasarkan gambar 1, terdapat beberapa tahapan penelitian yang akan dilakukan agar penelitian lebih terarah dan terukur hasil yang diharapkan.



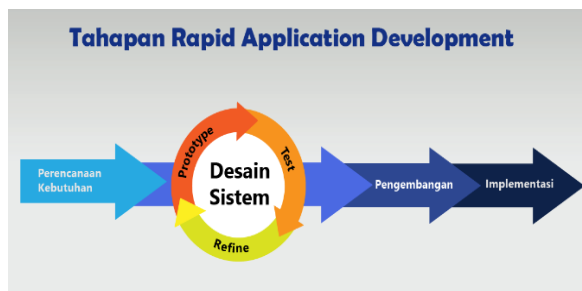
Gambar 1. Tahapan Penelitian

2.1. Studi Literatur

Studi literatur adalah serangkaian kegiatan yang berkenaan dengan metode pengumpulan data pustaka, membaca, dan mencatat, serta mengelola bahan penelitian [3]. Tahap ini dilakukan untuk mendapatkan teori-teori yang valid terkait dengan penelitian yang dilakukan. Sumber literatur yang digunakan untuk dijadikan referensi antara lain buku, jurnal, artikel, dan situs internet. Tujuan utama melakukan kajian literatur adalah menemukan variabel-variabel yang akan diteliti, membedakan hal-hal yang sudah dilakukan, dan menentukan hal-hal yang perlu dilakukan, melakukan sintesis dan memperoleh perspektif baru, serta menentukan makna dan hubungan antar variabel.

2.2. Metode Pengembangan Aplikasi

Metode yang digunakan dalam pengembangan aplikasi ini adalah *Rapid Application Development (RAD)*. Metode ini terkenal karena dapat menghemat waktu dalam proses pengembangan, karena fokusnya pada pengembangan sistem yang cepat melalui siklus pengulangan berbasis umpan balik yang berkelanjutan [4], [5], [6]. RAD dapat menghasilkan perangkat lunak yang dinamis, fleksibel, dimana metode RAD yang secara langsung melibatkan antara pengguna dengan perancang sistem secara langsung [7].



Gambar 2. Metode RAD (*Rapid Application Development*)

Metode RAD memiliki 4 tahapan yaitu sebagai berikut:

1. *Requirements Planning* (Perencanaan Kebutuhan)

Tahap ini melibatkan semua pemangku kepentingan (*stakeholders*) untuk mengidentifikasi dan memahami kebutuhan bisnis dan teknis

2. *Design Workshop* (Desain Sistem)

Pada tahap ini, pengembang dan pengguna bekerja sama secara erat untuk membuat prototipe awal dari sistem yang akan dikembangkan. Aktivitas utama meliputi: pembuatan prototipe, pengujian dan validasi, iterasi berulang

3. *Development* (Pengembangan)

Setelah prototipe disetujui, tahap pengembangan dimulai dengan fokus pada pengembangan sistem sebenarnya. Aktivitas pada tahap ini meliputi: pengembangan cepat, integrasi komponen, pengujian sistem, umpan balik berkelanjutan.

4. *Implementation* (Penerapan)

Tahap akhir dari metode RAD adalah penerapan sistem ke lingkungan produksi. Aktivitas utama meliputi: pelatihan pengguna, konversi data, instalasi sistem, dukungan pasca-implementasi.

2.3. Kesimpulan dan Saran

Pada tahap terakhir ini, akan dikemukakan kesimpulan dan saran yang didapat guna untuk membangun sistem yang dibuat menjadi lebih baik lagi, dan tentu saja pada kesimpulan akan dijabarkan sebuah hasil yang memang diharapkan oleh pengguna dan peneliti agar dapat menjadi sebuah referensi ke depannya bagi para peneliti selanjutnya.

2.4. Algoritma Linear Congruent Method (LCM)

Salah satu algoritma yang telah terbukti efisien dalam menghasilkan angka acak adalah *Linear Congruent Method (LCM)*. Dalam penelitian [8] penerapan Algoritma LCM mampu menghasilkan pengacakan soal yang efektif, di mana tidak ada perulangan nilai yang sama jika nilai variabel yang dideklarasikan memenuhi parameter penentuan dari Algoritma LCM yang ada yaitu nilai *increment* dan modulus harus prima relatif.

Algoritma LCM memiliki sifat-sifat yang

membuatnya ideal untuk digunakan dalam konteks aplikasi pendidikan, termasuk kemudahan implementasi, periodisitas yang baik, dan distribusi angka acak yang cukup merata. Algoritma LCM (*Linear Congruent Method*) pertama kali diperkenalkan Thomson pada tahun 1958. Alasan digunakannya metode LCM selain populer adalah mudah dan efektif dalam mengacak soal, metode ini hanya memerlukan memori yang kecil (satu bilangan modulo- m , umumnya 32 atau 64 bit) untuk penyimpanan sementara bilangan yang dihasilkan. Salah satu sifat sekaligus kelemahan dari metode ini adalah terjadi perulangan pada periode waktu tertentu atau setelah sekian kali pembangkitan. Untuk mengatasi masalah pengulangan dalam beberapa kali kebangkitan maka penggunaan Konstanta a (konstanta pengali), c (*increment*/konstanta penambah) dan m (modulus pembagi) harus tepat sehingga soal benar-benar acak.

Algoritma LCM bekerja dengan menggunakan formula rekursif untuk menghasilkan sebuah deret bilangan acak. Rumus dasar dari algoritma ini adalah:

$$X_{n+1} = (a X_n + c) \text{ mod } m \quad (1)$$

Di mana:

1. X_{n+1} adalah bilangan acak berikutnya dalam urutan.
2. X_n adalah bilangan acak saat ini atau bilangan awal (*seed*).
3. a adalah konstanta pengali (*multiplier*).
4. c adalah konstanta penambah (*increment*).
5. m adalah *modulus* (batas atas).

Komponen Utama Algoritma LCM

1. *Seed* X_0 Nilai awal atau seed yang digunakan untuk memulai proses pengacakan. Pemilihan seed yang berbeda akan menghasilkan urutan bilangan acak yang berbeda, namun urutan tersebut akan selalu sama jika seed yang sama digunakan kembali.
2. *Multiplier* (a): Sebuah konstanta yang digunakan untuk mengalikan nilai saat ini. Pemilihan nilai yang tepat untuk multiplier sangat penting untuk memastikan bahwa urutan bilangan acak memiliki periode yang panjang dan distribusi yang baik.
3. *Increment* (c): Sebuah konstanta penambah yang ditambahkan ke hasil perkalian. Penggunaan nilai increment yang tepat juga mempengaruhi kualitas dan periode urutan bilangan acak yang dihasilkan.
4. *Modulus* (m): Nilai modulus menentukan rentang bilangan acak yang dihasilkan. Modulus biasanya dipilih sebagai bilangan prima besar atau kekuatan dari dua, tergantung pada kebutuhan spesifik dari aplikasi.

Penerapan Algoritma LCM terbukti efektif dalam menghasilkan soal yang berbeda atau secara acak untuk setiap siswa yang menggunakan aplikasi [9], [10]. Dalam konteks aplikasi belajar mandiri, Algoritma LCM digunakan untuk menghasilkan angka acak yang dapat menentukan variasi soal latihan, urutan pembelajaran, dan tes. Ini membantu memastikan bahwa setiap siswa mendapatkan pengalaman belajar yang unik dan tidak monoton, sehingga meningkatkan keterlibatan

dan hasil belajar siswa. Dengan menggunakan LCM, aplikasi dapat menyediakan konten yang acak namun tetap terstruktur dan relevan sesuai dengan kebutuhan pembelajaran siswa.

Berikut ini contoh penerapan algoritma LCM, ada 50 soal yang tersimpan dalam komputer. Setiap ujian bisa terdiri dari 10, 20, 30, atau 40 soal. Setiap soal punya nomor unik yang digunakan untuk mengacak urutannya. Agar soal tidak terulang saat diacak. Cara ini menggunakan angka-angka tertentu (a=11, c=7, X₀=1, dan m=50) untuk menghasilkan urutan soal yang acak setiap kali. Contohnya, soal pertama yang muncul akan dihitung dengan Persamaan (1).

Tabel 1. Contoh Perhitungan Algoritma LCM

No	Rumus	Hasil Pengacakan
1.	$X_1 = (11 (1) + 7) \text{ mod } 50$	18
2.	$X_2 = (11 (18) + 7) \text{ mod } 50$	5
3.	$X_3 = (11 (5) + 7) \text{ mod } 50$	12
4.	$X_4 = (11 (12) + 7) \text{ mod } 50$	39
5.	$X_5 = (11 (39) + 7) \text{ mod } 50$	36
6.	$X_6 = (11 (36) + 7) \text{ mod } 50$	3
7.	$X_7 = (11 (3) + 7) \text{ mod } 50$	40
8.	$X_8 = (11 (40) + 7) \text{ mod } 50$	47
9.	$X_9 = (11 (47) + 7) \text{ mod } 50$	24
10.	$X_{10} = (11 (24) + 7) \text{ mod } 50$	21
11.	$X_{11} = (11 (21) + 7) \text{ mod } 50$	38
12.	$X_{12} = (11 (38) + 7) \text{ mod } 50$	24
13.	$X_{13} = (11 (24) + 7) \text{ mod } 50$	32
14.	$X_{14} = (11 (32) + 7) \text{ mod } 50$	9
15.	$X_{15} = (11 (9) + 7) \text{ mod } 50$	6
16.	$X_{16} = (11 (6) + 7) \text{ mod } 50$	23
17.	$X_{17} = (11 (23) + 7) \text{ mod } 50$	10
18.	$X_{18} = (11 (10) + 7) \text{ mod } 50$	17
19.	$X_{19} = (11 (17) + 7) \text{ mod } 50$	44
20.	$X_{20} = (11 (44) + 7) \text{ mod } 50$	42

Setelah di hitung menggunakan rumus yang sudah ditentukan, akan didapatkan sederet nomor soal secara acak, yaitu: 18, 5, 12, dan seterusnya sampai 42. Angka-angka yang dipilih untuk

perhitungan (a, c, dan m) ternyata sudah tepat, karena semua nomor soal muncul hanya sekali dan tidak ada yang berulang. Ini artinya, setiap kali ujian, soal yang keluar akan selalu berbeda.

Angka awal untuk perhitungan (X_n) akan berubah-ubah tergantung pada berapa kali ujian. Misalnya, pada ujian pertama, X_n adalah 1. Tapi, kalau ujian kedua, X_n menjadi 2, dan seterusnya.

Karena soal yang dimiliki sebanyak 50 soal, maka hasil perhitungan yang dilakukan akan menghasilkan angka antara 0 sampai 49. Tapi, karena tidak ada nomor soal 0, maka jika keluar angka 0, akan diganti dengan angka 50.

3. HASIL DAN PEMBAHASAN

Penggunaan algoritma *Linear Congruent Method* dilakukan langsung pada Aplikasi Belajar Mandiri Siswa dengan mengimplementasikan ke dalam kode program yang telah dibuat ke dalam sistem. Berikut ini merupakan kode program implementasi algoritma LCM dalam Aplikasi Belajar Mandiri Siswa:

```
function
linearCongruentialGenerator($seed,
$modulus, $multiplier, $increment, $n)
{
    $randomNumbers = [];
    $previous = $seed;

    for ($i = 0; $i < $n; $i++) {
        $next = ($multiplier *
$previous + $increment) % $modulus;
        $randomNumbers[] = $next;
        $previous = $next;
    }

    return $randomNumbers;
}
```

```

}

// implement the Linear Congruent
Method (LCM) algorithm to generate a
random number

$seed = time(); // Use current
timestamp as seed

$modulus = 10007; // A large prime
number

$multiplier = 17; // Any number

$increment = 43; // Any number

$numberOfQuestions = count($quiz-
>questions);

// Generate random numbers

$randomIndices =
linearCongruentialGenerator($seed,
$modulus, $multiplier, $increment,
$numberOfQuestions);

$questions = $quiz->questions;
$details = [];
foreach ($randomIndices as $index) {
    $question = $questions[$index %
$questions->count()];
    $details[] = [
        'question_id' => $question-
>id,
        'answer_id' => null,
        'is_correct' => false,
        'correct_answer_id' =>
$question->answers-
>where('is_correct', true)->first()-
>id,
        'score' => 0
    ];
}

```

Dalam penerapan Algoritma *Linear Congruent Method* ini, soal yang muncul akan ditampilkan secara acak dengan menggunakan

Algoritma LCM. Berikut ini pembahasan bagaimana LCM menampilkan kesepuluh soal. LCM mempunyai periode penuh ($m - 1$) jika memenuhi syarat berikut.

1. b relative prima terhadap m
2. $a - 1$ dapat dibagi dengan semua faktor prima dari m
3. $a - 1$ adalah kelipatan 4 jika m adalah kelipatan 4.
4. $m > m(a, b, X_0)$
5. $a > 0, b > 0$

Sesuai dari syarat diatas untuk mendapatkan periode penuh dari LCM, tentukan:

$$a = 2,$$

$$b = 5$$

$$m = 11$$

$$X_0 = 4 \text{ (diambil melalui proses acak)}$$

Sehingga proses pengacakan soal latihan dalam Aplikasi Belajar Mandiri Siswa dapat dilihat pada tabel 2.

Tabel 2. Hasil Perhitungan Algoritma LCM

No	Rumus	Hasil Pengacakan
1.	X_0	4
2.	$X_1 = (2(4) + 5) \text{ mod } 11$	2
3.	$X_2 = (2(2) + 5) \text{ mod } 11$	9
4.	$X_3 = (2(9) + 5) \text{ mod } 11$	1
5.	$X_4 = (2(1) + 5) \text{ mod } 11$	7
6.	$X_5 = (2(7) + 5) \text{ mod } 11$	8
7.	$X_6 = (2(8) + 5) \text{ mod } 11$	10
8.	$X_7 = (2(10) + 5) \text{ mod } 11$	3
9.	$X_8 = (2(3) + 5) \text{ mod } 11$	0
10.	$X_9 = (2(0) + 5) \text{ mod } 11$	5
11.	$X_{10} = (2(4) + 5) \text{ mod } 11$	4
12.	$X_{11} = (2(2) + 5) \text{ mod } 11$	2

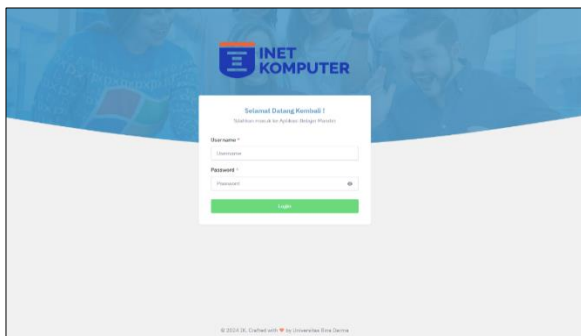
Dari Tabel 1 dengan parameter $a = 2, b = 5, m = 11$, dan $X_0 = 4$, diperoleh urutan angka acak $X^1 = 2$ dan seterusnya hingga periode ke-10. Dalam aplikasi game, angka-angka acak ini akan digunakan untuk menampilkan 10 lagu secara

acak. Lagu pertama yang akan muncul adalah lagu keempat, sesuai dengan nilai awal $X^0 = 4$. Kemudian, hasil perhitungan LCM dengan parameter yang sama menghasilkan angka $X^1 = 2$, $X^2 = 9$, $X^3 = 1$, dan seterusnya. Setelah periode ke-10, angka kembali ke nilai awal $X^0 = 4$, menunjukkan adanya pola pengulangan. Hal ini disebabkan oleh sifat algoritma LCM yang menghasilkan urutan acak yang berulang pada titik tertentu.

3.1. Tampilan Antarmuka Admin

1. Halaman Login Admin

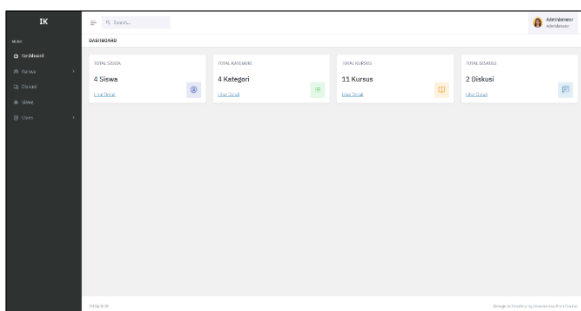
Halaman *login* ini berfungsi untuk memeriksa hak akses pengguna ketika akan masuk ke sistem. Tampilan halaman *login* admin dan pengelola penginapan pada gambar 2.



Gambar 2. Tampilan Login Admin

2. Halaman Dashboard Admin

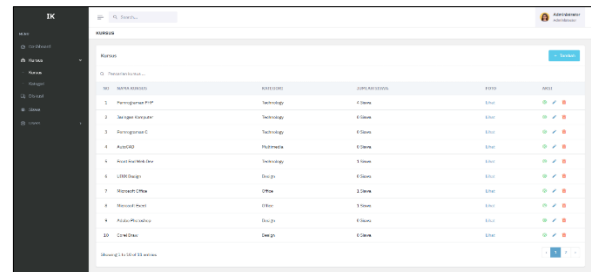
Halaman *dashboard* berfungsi untuk merupakan tampilan utama ketika login admin. Tampilan halaman *login* admin pada gambar 3.



Gambar 3. Tampilan Dashboard Admin

3. Halaman Kursus

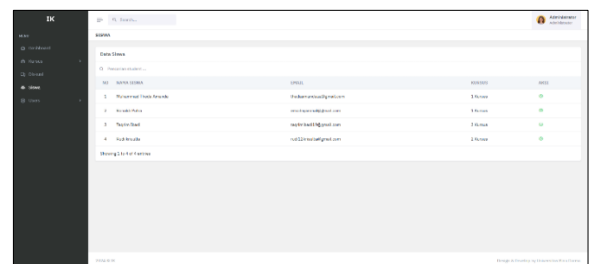
Halaman kursus ini berfungsi untuk menambah, mengubah atau menghapus daftar kursus yang ada di INET Komputer. Tampilan halaman kursus terlihat pada gambar 4.



Gambar 4. Tampilan Kursus

4. Halaman Siswa

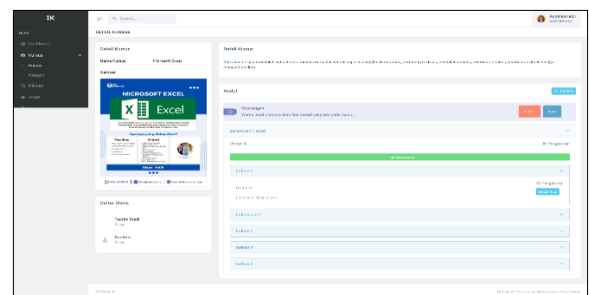
Halaman siswa ini berfungsi untuk melihat semua data siswa yang terdaftar pada aplikasi, Hal ini tentu saja sangat berguna untuk mengelola akun siswa jika suatu waktu mereka memerlukan bantuan.



Gambar 5. Tampilan Siswa

5. Halaman Detail Kursus

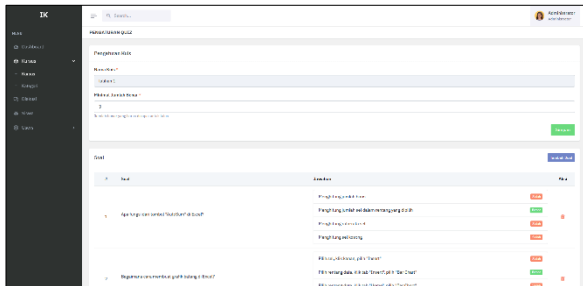
Pada halaman ini, admin bisa mengatur rincian dari kelas tersebut mulai dari deskripsi kelas, modul kelas, latihan, dan kuis dari kelas tersebut.



Gambar 6. Tampilan Detail Kursus

6. Halaman Pengaturan Kuis

Pada halaman ini, admin bisa mengatur berapa jumlah soal yang harus benar agar siswa bisa melanjutkan belajar ke sesi berikutnya. Admin juga bisa mengatur berapa jumlah soal yang tersedia dan apa saja soal yang tersedia.



Gambar 7. Tampilan Pengaturan Kuis

3.2. Tampilan Antarmuka Siswa

1. Halaman Login

Halaman *login* ini berfungsi untuk memeriksa hak akses pengguna ketika masuk ke aplikasi. Tampilan halaman *login* pengguna pada gambar 8.



Gambar 8. Tampilan Login Siswa

2. Halaman Daftar Akun

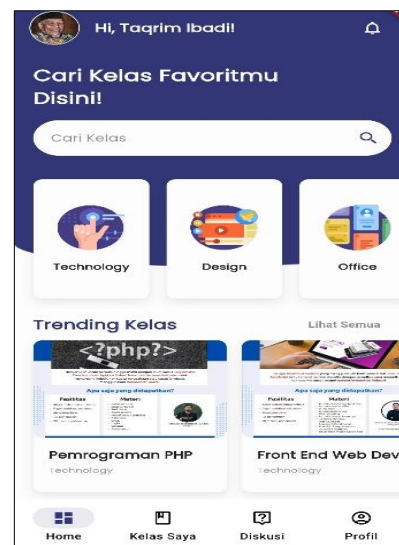
Halaman daftar akun ini berfungsi untuk melakukan pendaftaran akun ke sistem. Tampilan halaman daftar akun pada gambar 9.



Gambar 9. Tampilan Daftar Akun Siswa

3. Halaman Beranda Siswa

Halaman home ini berfungsi untuk menampilkan semua menu yang tersedia pada aplikasi dan semua kelas yang tercatat pada sistem. Tampilan halaman home pada gambar 10.



Gambar 10. Tampilan Beranda

4. Halaman Kelas Saya

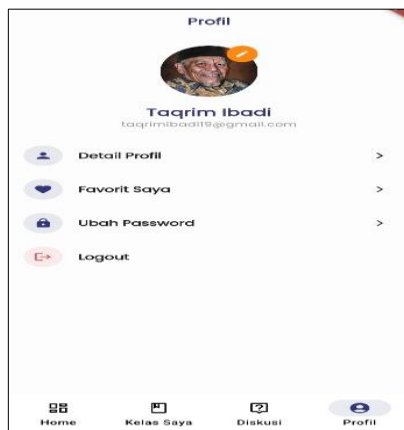
Halaman kelas saya ini berfungsi untuk mengumpulkan daftar kelas yang sudah diambil siswa untuk mereka pelajari. Tampilan halaman kelas pada gambar 11.



Gambar 11. Tampilan Kelas Saya

5. Halaman Profile Siswa

Halaman profile ini berfungsi untuk mengatur, mengubah dan menghapus data pribadi dari siswa. Tampilan halaman profile pada gambar 12.



Gambar 12. Tampilan Profile

6. Halaman Latihan Siswa

Pada halaman ini, siswa akan melihat soal latihan yang tersedia pada masing-masing kelas sebelum melanjutkan pelajaran berikutnya.



Gambar 13. Tampilan Latihan Siswa

3.3. Pengujian Aplikasi

Pada tahap pengujian aplikasi ini menggunakan metode *Blackbox Testing*. Pengujian ini dilakukan dengan cara menjalankan website dan aplikasi kemudian melihat *output* yang dihasilkan itu telah sesuai dengan hasil yang diharapkan. Hasil pengujian *blackbox testing* adalah sebagai berikut.

1. Unit Testing

Pengujian ini dilakukan pada unit terkecil dari program yang bertujuan untuk menguji aplikasi layak atau tidak untuk dipakai. Hasil *unit testing* disajikan pada tabel 2.

Tabel 2. Unit Testing

No	Test Case	Output	Actor	Hasil
1	Membuka website	Tampil halaman <i>login</i>	Admin, Kepala Sekolah	Sesuai
2	Pilih menu <i>dashboard</i>	Tampil halaman <i>dashboard</i>	Admin, Kepala Sekolah	Sesuai
3	Pilih menu data kelas	Tampil data kelas	Admin	Sesuai
4	Pilih <i>botton</i> tambah data kelas	Tampil <i>form input</i> data kelas	Admin	Sesuai
5	Pilih <i>botton</i> <i>edit</i> data kelas	Tampil <i>form edit</i> data kelas	Admin	Sesuai
6	Pilih <i>botton</i> hapus data kelas	Tampil <i>alert</i> data kelas berhasil dihapus	Admin	Sesuai
7	Pilih menu data kategori	Tampil data kategori	Admin	Sesuai
8	Pilih <i>botton</i> tambah data kategori	Tampil <i>form input</i> data kategori	Admin	Sesuai
9	Pilih <i>botton</i> <i>edit</i> data kategori	Tampil <i>form edit</i> data kategori	Admin	Sesuai
10	Pilih <i>botton</i> hapus data kategori	Tampil <i>alert</i> data kategori berhasil dihapus	Admin	Sesuai
11	Pilih menu data soal	Tampil data soal	Admin	Sesuai
12	Pilih <i>botton</i>	Tampil <i>form</i>	Admin	Sesuai

No	Test Case	Output	Actor	Hasil
13	tambah data soal	input data soal	Admin	Sesuai
	Pilih <i>button</i> edit data soal	Tampil <i>form</i> edit data soal		
14	Pilih <i>button</i> hapus data soal	Tampil <i>alert</i> data soal berhasil dihapus	Admin	Sesuai
	Pilih <i>button</i> tambah data soal	Tampil <i>form</i> input data soal		
16	Pilih <i>button</i> hapus data soal	Tampil <i>alert</i> data soal berhasil dihapus	Admin	Sesuai
	Pilih menu data user	Tampil data user		
18	Pilih menu data forum	Tampil data forum	Admin	Sesuai
	Pilih <i>button</i> detail data diskusi	Tampil halaman detail diskusi		
20	Pilih menu home	Tampil halaman home	Siswa	Sesuai
	Pilih menu kelas	Tampil halaman kelas		
22	Pilih <i>button</i> detail data kelas	Tampil halaman detail data kelas	Siswa	Sesuai
	Pilih <i>button</i> detail kelas data materi	Tampil halaman detail kelas data materi		
24	Pilih <i>button</i> detail kelas data latihan	Tampil halaman detail kelas data latihan	Siswa	Sesuai
	Pilih <i>button</i> ambil kelas	Tampil <i>alert</i> kelas berhasil diambil		
26	Pilih <i>button</i> konfirmasi kelas	Tampil <i>alert</i> kelas berhasil dikonfirmasi	Siswa	Sesuai
	Pilih <i>button</i> kerjakan kuis	Tampil data kuis		
28	Pilih <i>button</i> diskusi	Tampil modal form diskusi	Siswa	Sesuai
	Pilih <i>button</i> kirim diskusi	Tampil <i>alert</i> diskusi berhasil dibuat		
30	Pilih menu	Tampil	Siswa	Sesuai

No	Test Case	Output	Actor	Hasil
31	riwayat kelas	halaman riwayat kelas	Siswa	Sesuai
	Pilih menu profile	Tampil halaman profil		
32	Pilih menu ubah data profile	Tampil halaman ubah data profil	Siswa	Sesuai
	Pilih menu ubah password	Tampil halaman ubah password		
34	Pilih <i>button</i> logout	Tampil halaman <i>login</i>	Admin, Siswa	Sesuai

2. Integration Testing

Integration Testing dilakukan unit-unit atau komponen pada aplikasi untuk memverifikasi interaksi yang terjadi. Hasil *Integration Testing* disajikan pada tabel 4.34 berikut ini.

Tabel 3. Integration Testing (Admin)

No	Test Case	Output	Hasil
1	Buka Aplikasi	Tampil halaman <i>login</i>	Sesuai
2	Masukkan <i>username & password</i>	Jika login benar maka akan tampil halaman <i>dashboard</i>	Sesuai
		Jika login salah maka akan tampil <i>alert</i>	
3	Pilih menu <i>dashboard</i>	Tampil data <i>dashboard</i>	Sesuai
4	Pilih menu data kelas	Tampil data kelas	Sesuai
		4.1 Pilih <i>button</i> tambah data kelas	
	4.1.1 Pilih <i>button</i> simpan data kelas	Tampil <i>alert</i> berhasil disimpan dan tampil halaman data kelas	Sesuai
		4.1.2 Pilih <i>button</i> batal tambah data kelas	
	4.1.3 Pilih <i>button</i> reset data yang telah dimasukkan	Tampil <i>form</i> input data kelas yang kosong	Sesuai
		4.2 Pilih <i>button</i>	

No	Test Case	Output	Hasil	
5	edit data kelas	edit data kelas		
	4.3 Pilih <i>botton</i> hapus data kelas	Tampil <i>alert</i> data berhasil dihapus	Sesuai	
	Pilih menu data soal	Tampil data soal	Sesuai	
	5.1 Pilih <i>botton</i> tambah data soal	Tampil <i>form input</i> data soal	Sesuai	
	5.1.1 Pilih <i>botton</i> simpan data soal	Tampil <i>alert</i> berhasil disimpan dan tampil halaman data soal	Sesuai	
	5.1.2 Pilih <i>botton</i> batal tambah data soal	Tampil halaman data soal	Sesuai	
	5.1.3 Pilih <i>botton</i> reset data yang telah dimasukkan	Tampil <i>form input</i> data soal yang kosong	Sesuai	
	5.2 Pilih <i>botton</i> edit data soal	Tampil <i>form edit</i> data soal	Sesuai	
	5.3 Pilih hapus data soal	Tampil <i>alert</i> data berhasil dihapus	Sesuai	
	6	Pilih menu data kategori	Tampil data kategori	Sesuai
6.1 Pilih <i>botton</i> tambah data kategori		Tampil <i>form input</i> data kategori	Sesuai	
6.1.1 Pilih <i>botton</i> simpan data kategori		Tampil <i>alert</i> berhasil disimpan dan tampil halaman data kategori	Sesuai	
6.1.2 Pilih <i>botton</i> batal tambah data kategori		Tampil halaman data kategori	Sesuai	
6.1.3 Pilih <i>botton</i> reset data yang telah dimasukkan		Tampil <i>form input</i> data kategori yang kosong	Sesuai	
6.2 Pilih <i>botton</i> edit data kategori		Tampil <i>form edit</i> data kategori	Sesuai	
6.3 Pilih hapus data kategori		Tampil <i>alert</i> data berhasil dihapus	Sesuai	
7		Pilih menu data	Tampil data user	Sesuai

No	Test Case	Output	Hasil
8	user		
	Pilih menu data diskusi	Tampil data diskusi	Sesuai
9	8.1 Pilih <i>botton</i> detail diskusi	Tampil halaman detail diskusi	Sesuai
	Pilih <i>botton Logout</i>	Berhasil keluar aplikasi dan tampil halaman <i>login</i>	Sesuai

Tabel 4. Integration Testing (User)

No	Test Case	Output	Hasil
1	Buka Aplikasi	Tampil halaman <i>login</i>	Sesuai
2	Masukkan <i>username & password</i>	Jika login benar maka akan tampil halaman <i>home</i>	Sesuai
		Jika login salah maka akan tampil <i>alert</i>	
3	Pilih menu <i>home</i>	Tampil halaman <i>home</i>	Sesuai
	3.1 Pilih menu kelas	Tampil data penginapan	
3.1.1	Pilih <i>botton detail</i> kelas	Tampil informasi <i>detail</i> tentang penginapan	Sesuai
		Tampil informasi detail latihan tentang kelas	
3.1.3	Pilih <i>detail</i> latihan kelas	Tampil informasi detail materi tentang kelas	Sesuai
		Tampil informasi detail deskripsi kelas	
3.1.4	Pilih <i>detail</i> materi kelas	Tampil data latihan	Sesuai
		Tampil data materi	
3.1.5	Pilih <i>detail</i> deskripsi kelas	Tampil data ambil kelas	Sesuai
		Tampil informasi detail deskripsi kelas	
3.2	Pilih menu latihan	Tampil data ambil kelas	Sesuai
		Tampil konfirmasi ambil kelas	
3.3	Pilih menu ambil kelas	Tampil data ambil kelas	Sesuai
		Tampil konfirmasi submit latihan	
3.3.1	Pilih <i>botton</i> ambil kelas	Tampil data latihan	Sesuai
		Tampil konfirmasi submit latihan	
3.4	Pilih menu mengerjakan latihan	Tampil data ambil kelas	Sesuai
		Tampil konfirmasi submit latihan	
3.4.1	Pilih <i>botton</i> submit latihan	Tampil data ambil kelas	Sesuai
		Tampil konfirmasi submit latihan	

No	Test Case	Output	Hasil
	3.5 Pilih menu diskusi	Tampil data diskusi	Sesuai
	3.5.1 Pilih button kirim diskusi	Tampil konfirmasi diskusi	Sesuai
4	Pilih menu profil	Tampil halaman profil <i>user</i>	Sesuai
5	Pilih menu informasi akun	Tampil informasi akun yang sedang <i>login</i>	Sesuai
	5.1 Pilih button ubah data profil	Tampil <i>form edit</i> data profil	Sesuai
	5.2 Pilih button ubah password	Tampil <i>form</i> ubah password	Sesuai
6	Pilih button Logout	Berhasil keluar aplikasi dan tampil halaman <i>login</i>	Sesuai

4. KESIMPULAN

Penelitian ini berhasil menerapkan Algoritma *Linear Congruent Method* (LCM) pada aplikasi belajar mandiri siswa di Lembaga Kursus INET Komputer. Berdasarkan hasil penelitian, disimpulkan bahwa: Implementasi Algoritma LCM: *Algoritma Linear Congruent Method* dapat diintegrasikan dengan baik ke dalam aplikasi belajar mandiri. Algoritma ini berfungsi efektif dalam menghasilkan angka acak yang diperlukan untuk fitur pembelajaran, seperti soal latihan yang bervariasi dan tes acak, yang meningkatkan interaksi siswa dengan aplikasi. Penggunaan Algoritma LCM dalam aplikasi belajar mandiri terbukti meningkatkan efektivitas pembelajaran. Siswa dapat mengakses konten pembelajaran yang lebih beragam dan acak, yang membantu mereka untuk memahami materi lebih baik dan mendorong pembelajaran yang lebih mendalam. Beberapa tantangan teknis ditemukan selama implementasi, seperti penyesuaian algoritma dengan sistem yang ada dan optimisasi performa aplikasi. Namun,

tantangan ini dapat diatasi melalui penyesuaian teknis yang tepat dan pengujian berulang.

DAFTAR PUSTAKA

- [1] M. R. Muhaimin dan E. Zumrotun, "Pengembangan Media Pembelajaran Berbasis Smart Apps Creator pada Materi Satuan Ukuran Kelas V Sekolah Dasar," *JURNALBASICEDU*, vol. 7, no. 3, pp. 1935-1950, 2023.
- [2] A. P. Jumala, R. Parluka dan F. A. Akbar, "Perancangan Sistem Ujian Online Dan Implementasi Algoritma Lcm Dalam Fitur Pengacakan Soal," *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, vol. 1, no. 2, pp. 302-311, 2020.
- [3] N. Anggita, Salsabiila and R. Novitasari, "Studi Literatur: Pengaruh Media Pembelajaran dalam Peningkatan Hasil Belajar Siswa Sekolah Dasar," *Jurnal Pendidikan Tambusai*, vol. 8, nr 1, pp. 5654-5658, 2024.
- [4] M. Tabrani dan H. Priyandaru, "Application of the Rapid Application Development Method to the BAZNAS Zakat Receipt Information System in Karawang," *Jurnal Teknologi Dan Open Source*, vol. 4, no. 1, pp. 78-84, 2021.
- [5] S. D. Ratna, W. Nurcahayo and N. Faizah, "Pengembangan Sistem Pelayanan Klinik Budhi Pratama Gedong Berbasis Web dengan Metode Rapid Application Development (RAD)," *JOURNAL DIGITAL TECHNOLOGY TREND*, vol. 3, nr 1, pp. 42-50, 2024.
- [6] A. C. Roso, A. Herdiansah, T. Handayani, N. Fitrawati and Najmuddin, "Penerapan Metode Rapid Application Development (RAD) Pengembangan Sistem Informasi Manajemen barang Reject," *JIKA (Jurnal of Informatics)*, vol. 8, nr 3, pp. 326-333, 2024.
- [7] S. Kosasi, "Penerapan Rapid Application Development Dalam Sistem Perniagaan Elektronik Furniture," *Citec Journal*, vol. 2, no. 4, p. 265-276, 2015.
- [8] B. A. Herlambang, S. Wibowo, Choirunnisa and V. A. Veria, "Implementasi Algoritma Linear Congruent

Method Pada Pengacakan Soal Kuis dalam Aplikasi Mobile Learning Anemia Berbasis Android (MobiliA),” *TRANSFORMATIKA*, pp. 182-160, 2021.

- [9] D. A. Prastiwi, A. M. Pardede och Z. Fatmaira, ”Penerapan Algoritma Linier Congruent Method Pada Pengacakan Soal Ujian Berbasis Online di SD Muhammadiyah Sei Cabang,” *Switch: Jurnal Sains dan Teknologi Informasi*, vol. 2, nr 4, pp. 124-133, 2024.
- [10] R. Pebrian, Fauziah och I. D. Sholihati, ”Algoritma Linear Congruent Method Dan Algoritma Fisher-Yates Shuffle Pada Kuis Ketangkasan Berbasis Android,” *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 6, nr 2, pp. 167-177, 2021.