

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pembelajaran adalah suatu proses yang rumit dan melibatkan berbagai elemen, seperti perencanaan, pelaksanaan, dan evaluasi, guna mencapai hasil yang maksimal. Evaluasi dalam pembelajaran memiliki peranan yang krusial dalam dunia pendidikan, karena dapat memberikan gambaran mengenai efektivitas metode yang diterapkan serta tingkat pemahaman siswa. Melalui evaluasi yang efektif, pendidik dapat mengenali kekuatan dan kelemahan dalam proses pembelajaran, sehingga perbaikan dan inovasi dapat dilakukan untuk meningkatkan mutu pendidikan. (Rangga Putera Boroallo et al., 2025). Evaluasi biasanya dilakukan melalui kegiatan ujian, baik ujian harian, tengah semester, maupun akhir semester.

Di era saat ini dengan adanya perkembangan teknologi telah membuat kemudahan dalam mengakses teknologi dan informasi terutama dalam bidang pendidikan (Agustin et al., 2025). Pendidikan di Indonesia semakin diarahkan pada pemanfaatan teknologi digital dalam mendukung proses pembelajaran dan juga evaluasi. Dalam pemanfaatan teknologi ini, metode pelaksanaan ujian yang awalnya dilaksanakan secara konvensional berbasis kertas, kini mengalami transformasi menuju sistem digital yang lebih praktis, yaitu pelaksanaan ujian dengan sistem *Computer Based Test* (CBT) berbasis android.

Di SMP Negeri 1 Karimun, sistem ujian *online* telah mulai diterapkan

menggunakan platform *Google Form*. Platform ini memberikan kemudahan bagi guru dalam membuat soal, mendistribusikan tautan ujian, serta memonitor hasil pengerjaan siswa secara langsung. Selain itu, siswa juga dapat mengerjakan soal dengan cepat hanya melalui perangkat *mobile* yang mereka miliki. Namun, meskipun penerapan *Google Form* memberikan banyak kemudahan, pada praktiknya ditemukan beberapa permasalahan yang cukup krusial terkait aspek integritas akademik dan keamanan ujian.

Salah satu permasalahan utama adalah banyaknya kasus siswa mengerjakan ujian lebih dari satu kali. *Google Form* secara *default* memberikan opsi untuk mengizinkan pengisian ulang apabila tidak dikonfigurasi secara ketat. Dalam beberapa kejadian, siswa melakukan *submit* jawaban berkali-kali untuk mencari kombinasi jawaban yang benar. Selain itu, fitur hasil otomatis *Google Form* sering kali menampilkan kunci jawaban setelah ujian selesai, terutama jika pengaturan tidak diatur secara khusus oleh guru. Kondisi ini menyebabkan sebagian siswa dapat melihat jawaban benar setelah percobaan pertama, kemudian membagikan kunci tersebut kepada teman-temannya.

Masalah ini berdampak langsung pada objektivitas hasil penilaian karena memungkinkan terjadinya kecurangan secara sistematis. Guru juga mengalami kesulitan untuk memastikan bahwa setiap siswa benar-benar mengerjakan ujian secara mandiri tanpa adanya pertukaran jawaban. Dengan demikian, diperlukan sistem ujian yang mampu mengurangi potensi kecurangan dan meningkatkan keamanan distribusi soal.

Salah satu pendekatan teknis yang dapat digunakan adalah penerapan

algoritma *Fisher–Yates Shuffling*, yaitu algoritma pengacakan yang mampu menghasilkan urutan acak sempurna (*perfect shuffle*) tanpa pola repetitif. Dengan menerapkan *Fisher–Yates* dalam aplikasi ujian berbasis *mobile*, setiap siswa dapat menerima urutan soal yang berbeda satu sama lain. Hal ini membuat praktik saling berbagi jawaban menjadi jauh lebih sulit karena urutan soal tidak lagi sama antar perangkat

Beberapa penelitian menggunakan algoritma *Fisher-Yates Shuffle* sebagai metode pengacakan angka pada kasus yang sama yaitu dalam pembuatan aplikasi kuis. (Akram et al., 2020a) membuktikan bahwa algoritma ini mampu menghasilkan susunan soal yang bervariasi dan adil, sehingga meminimalkan peluang kecurangan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang mendasari penulis menguraikan permasalahan yang akan diselesaikan pada penelitian ini, antara lain adalah sebagai berikut:

1. Bagaimana merancang aplikasi ujian *online* untuk siswa menengah pertama berbasis android *mobile* menggunakan Dart Flutter?
2. Bagaimana menerapkan metode *Fisher-Yates* untuk mengacak nomor soal ujian dan mengimplementasikan pada aplikasi ujian untuk siswa sekolah menengah pertama ?

1.3 Batasan Masalah

Berdasarkan uraian di atas maka didapatkan batasan masalah adalah sebagai berikut:

1. Aplikasi ini dibangun hanya untuk platform ujian siswa menengah pertama
2. Aplikasi ini hanya digunakan untuk siswa kelas 8.1 di SMP Negeri 1 Karimun yang terdiri dari 33 siswa.
3. Aplikasi ini hanya terdiri dari 11 mata pelajaran dengan masing-masing ujian memiliki 25 soal.
4. Hasil/*output* yang dihasilkan oleh sistem berupa hasil nilai ujian siswa menengah pertama
5. Aplikasi dibangun dengan pemrograman berbasis *mobile* dengan menggunakan bahasa pemrograman Dart Flutter.

1.4 Tujuan Penelitian

Adapun yang menjadi tujuan penelitian dalam membangun aplikasi adalah sebagai berikut:

1. Untuk merancang aplikasi ujian siswa menengah pertama berbasis android *mobile* menggunakan *Flutter*.
2. Untuk menerapkan metode algoritma *Fisher-Yates* dalam proses merancang aplikasi ujian sebagai metode untuk pengacakan nomor soal ujian untuk siswa menengah pertama.

1.5 Manfaat Penelitian

Penulis mengharapkan adanya dampak positif dari dilakukannya penelitian ini dengan memberikan manfaat:

1. Menambah wawasan penulis dalam pengembangan aplikasi berbasis android *mobile* dengan pemrograman dart flutter.
2. Membantu dan mempermudah guru dalam memberikan materi ujian serta menghemat waktu dalam mendapatkan hasil skor ujian siswa secara otomatis
3. Mempermudah siswa dalam melaksanakan ujian secara online dengan menggunakan handphone sebagai media ujian

1.6 Sistematika Penulisan

Agar mempermudah penyusunan skripsi maka penulis menyajikan sistematika penulisan sebagai berikut:

BAB I: PENDAHULUAN

Dalam bab ini dijelaskan dan diuraikan tentang latar belakang, perumusan masalah, Batasan masalah, tujuan dan manfaat penelitian. metode penelitian dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA

Dalam bab ini membahas materi yang mendukung terkait penelitian dalam penyusunan skripsi, mulai dari teori-teori tentang android dan aplikasi android, metode yang diterapkan dalam menyelesaikan permasalahan yang diteliti oleh penulis dalam penelitian.

BAB III: METODE PENELITIAN

Dalam bab ini penulis membahas metode penelitian yang akan dilakukan didasarkan pada fokus penelitian yang menitik beratkan pada perancangan, implementasi, dan pengujian aplikasi ujian berbasis *mobile*

BAB IV: HASIL DAN PEMBAHASAN

Bab ini membahas dan memaparkan hasil dari rancangan sistem, yang terdiri dari *input* serta *output* dari aplikasi yang telah dibangun.

BAB V: KESIMPULAN DAN SARAN

Bab akhir ini memaparkan kesimpulan dari keseluruhan pada bab-bab yang telah dibahas sebelumnya, juga terdapat saran atau masukan dalam pengembangan aplikasi ini.

BAB II

TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi merupakan perangkat lunak yang dirancang untuk membantu pengguna dalam melaksanakan tugas tertentu sesuai dengan kebutuhan yang spesifik. Dalam konteks teknologi informasi, aplikasi menjadi jembatan antara pengguna dengan sistem komputer atau perangkat *mobile* agar dapat melakukan berbagai kegiatan secara efisien dan terstruktur

Menurut Yuliani, R. (2020) dalam penelitiannya berjudul “Pengembangan Aplikasi *Mobile Learning* Berbasis Android untuk Meningkatkan Efektivitas Pembelajaran” (Jurnal Teknologi Informasi dan Pendidikan, 13(1), 25–33), aplikasi *mobile* memiliki keunggulan dalam hal portabilitas dan kemudahan akses. Hal ini karena pengguna dapat menjalankan aplikasi kapan pun dan di mana pun melalui perangkat *smartphone*.

2.2 Android

Android merupakan sistem operasi berbasis Linux yang dikembangkan oleh Google untuk perangkat bergerak seperti *smartphone* dan *tablet*. Sistem ini bersifat *open source*, memungkinkan pengembang melakukan modifikasi sesuai kebutuhan aplikasi yang dibuat (Astuti & Wibowo, 2021). Android menyediakan lingkungan yang fleksibel bagi pengembang untuk menciptakan berbagai jenis aplikasi, termasuk aplikasi pendidikan seperti sistem ujian online.

Menurut Prasetya et al. (2020), Android didesain dengan prinsip *open*

architecture yang memungkinkan integrasi dengan berbagai layanan eksternal, seperti *Firebase*, *Supabase*, dan API pihak ketiga. Sifat terbuka ini menjadikan Android sebagai sistem operasi dominan pada perangkat mobile di seluruh dunia, termasuk di Indonesia. Android juga memiliki kemampuan multitasking, keamanan sistem yang terjaga melalui pembaruan rutin, dan dukungan penuh dari komunitas pengembang global.

Dalam pengembangan aplikasi ujian online, Android menjadi platform ideal karena mendukung akses jaringan, penyimpanan data lokal dan cloud, serta integrasi dengan basis data seperti PostgreSQL melalui Supabase. Menurut Hidayah dan Saputra (2024), Android memberikan dukungan penuh terhadap *real-time synchronization* dan *data encryption*, yang penting untuk menjaga validitas dan keamanan data hasil ujian.

Selain itu, Android menyediakan berbagai *Application Programming Interface (API)* yang dapat digunakan untuk membangun sistem interaktif, seperti timer ujian, notifikasi hasil, dan autentikasi pengguna. Dukungan ini menjadikan Android sebagai fondasi utama bagi sistem pembelajaran modern berbasis mobile di era digital.

2.3 Android Studio

Android Studio merupakan *Integrated Development Environment (IDE)* resmi yang dikembangkan oleh Google untuk membangun aplikasi berbasis mobile Android. IDE ini menggunakan dasar IntelliJ IDEA dari JetBrains yang dikenal karena kestabilan dan juga fitur-fitur canggih untuk pemrograman berorientasi objek (Sari et al., 2021). Dengan Android Studio, para pengembang dapat menulis, menguji serta mendistribusikan aplikasi secara

efisien karena seluruh proses pengembangan dilakukan dalam satu lingkungan terpadu. Dengan fitur seperti code completion, lint tools, dan real-time rendering pada layout editor yang membantu pengembang dalam mempercepat pembuatan antarmuka pengguna (user interface).

Sedangkan dikutip dari (Rianof et al., 2020), Android Studio merupakan *Integrated Development Environment* (IDE) resmi untuk pengembangan aplikasi Android yang dikembangkan oleh Google berbasis IntelliJ. Platform ini menyediakan berbagai fitur yang mendukung proses pembuatan aplikasi, termasuk bagi programmer pemula. Android Studio dilengkapi dengan beragam *library* bawaan yang siap digunakan, sehingga dapat mempercepat dan memudahkan dalam proses pengembangan aplikasi android.

Sementara itu, Android Studio terintegrasi langsung dengan *Software Development Kit* (SDK) Android yang terus diperbarui oleh Google. SDK ini menyediakan beragam pustaka dan antarmuka pemrograman aplikasi (API) untuk memanfaatkan kemampuan perangkat keras, seperti kamera, GPS, sensor, dan lain-lain (Utami et al., 2022). Dengan adanya SDK ini, pengembang dapat membuat aplikasi yang responsif terhadap berbagai versi sistem operasi Android dan ukuran layar perangkat.

Android Studio juga menyediakan *Gradle Build System* yang berfungsi mengatur dependensi, otomatisasi kompilasi, dan juga membangun proyek dalam berbagai varian. Sistem ini sangat penting dalam proyek aplikasi ujian berbasis mobile dikarenakan memungkinkan proses kompilasi yang lebih cepat dan terstruktur. Berdasarkan studi oleh Putra dan Santoso (2022) menjelaskan bahwa penggunaan *Gradle* mempercepat integrasi berbagai *library* eksternal,

termasuk untuk koneksi dengan database maupun server ujian online.

2.4 Algoritma

Secara umum, menurut Kamus Besar Bahasa Indonesia, implementasi berarti pelaksanaan atau penerapan. Istilah ini biasanya digunakan untuk menggambarkan suatu kegiatan yang dilakukan guna mencapai tujuan tertentu. Implementasi adalah proses mengubah ide, konsep, kebijakan, atau inovasi menjadi tindakan nyata yang menghasilkan dampak, seperti perubahan dalam pengetahuan, keterampilan, serta nilai dan sikap. (ULFATIMAH HERNITA, 2020)

Menurut (Rambe Periyanti Purnamasari, 2021) Algoritma adalah serangkaian langkah logis yang disusun secara sistematis untuk menyelesaikan suatu masalah. Setiap langkah dalam algoritma harus dapat ditentukan kebenarannya, sehingga proses penyelesaian masalah dapat dilakukan secara terstruktur dan terukur.

Sementara itu, dikutip dari artikel (Nabila et al., n.d.) mengatakan bahwa Algoritma pemrograman merupakan rangkaian instruksi yang diberikan kepada komputer untuk dijalankan dalam rangka menyelesaikan sebuah masalah.

Algoritma dan pemrograman merupakan dua konsep fundamental yang saling terkait dalam bidang ilmu komputer. Algoritma adalah urutan langkah atau instruksi yang dibuat untuk menyelesaikan suatu masalah dengan cara yang efisien. Dalam hal ini, algoritma berperan sebagai dasar dari setiap solusi komputasi, yang mengatur proses pemecahan masalah secara sistematis dan terorganis.

2.5 Algoritma *Fisher-Yates Shuffling*

Algoritma *Fisher-Yates* merupakan metode yang digunakan untuk menghasilkan permutasi acak dari suatu himpunan terbatas, atau dengan kata lain, mengacak elemen-elemen di dalamnya (Asih et al., 2020). Proses ini sering disebut juga *Fisher-Yates Shuffle*, yaitu teknik pengacakan yang mampu menyusun ulang urutan data dalam sebuah array secara adil dan terdistribusi merata.

Algoritma *Fisher-Yates* ini banyak diterapkan dalam berbagai aplikasi permainan yang memerlukan proses pengacakan, misalnya pada permainan kartu maupun *puzzle*. Keunggulannya terletak pada kemampuannya menghasilkan hasil acak yang minim bias, sehingga kemungkinan munculnya urutan atau posisi yang sama menjadi sangat kecil.

Dikatakan oleh (Akram et al., 2020b) bahwa sejumlah penelitian menunjukkan penerapan algoritma *Fisher-Yates Shuffle* dalam berbagai konteks. Misalnya, di tingkat sekolah kejuruan (SMKN), algoritma ini digunakan untuk menekan biaya pelaksanaan ujian sekaligus mengurangi potensi kecurangan dalam ujian *online*. Selain itu, algoritma ini juga diterapkan pada sistem tes *online* untuk seleksi penerimaan mahasiswa baru, sehingga setiap peserta secara bersamaan dapat memperoleh soal ujian dengan susunan yang berbeda.

Berikut merupakan formula/rumus matematis dari bentuk algoritma *fisher-yates*:

Rumus Umum Algoritma *Fisher-Yates Shuffle*

$$J = \text{rand}(0,i) \dots\dots\dots (2.1)$$

Dan

$$\text{Swap} (S [i], S [j]) \dots\dots\dots(2.2)$$

Keterangan simbol:

- S = Array atau daftar elemen yang akan diacak
- n = Jumlah elemen dalam array S
- i = Indeks iterasi yang berjalan dari n – 1 ke 1
- j = Indeks acak yang dipilih dari rentang 0 hingga i
- rand (a,b) = Fungsi penghasil bilangan acak bulat antara a dan b (inklusif)
- Swap (S [i], S [j]) = Menukar posisi dua elemen dalam array S

Rumus Lengkap dalam Bentuk Pseudomatematis

Input: $S=[S_1,S_2,\dots,S_n]$

for i = n – 1 down to 1:

$j=\text{rand}(0,i)$

$S[i],S[j]=S[j],S[i]$ (2.3)

2.6 Flutter

Flutter adalah *framework open-source* yang dikembangkan oleh Google untuk membangun antarmuka (*user interface*) aplikasi pada Android maupun iOS. Berdasarkan penelitian (Setiawan et al., 2022) , sejumlah perusahaan besar seperti Google, Alibaba Group, dan Grab telah mengadopsinya dalam pengembangan aplikasi mereka. Salah satu keunggulan utama *Flutter* adalah kemampuannya mendukung *cross-platform*, yaitu memungkinkan aplikasi dijalankan di berbagai sistem operasi dengan basis kode yang sama.

Menurut Maulana dan Rachmat (2022), *Flutter* memiliki keunggulan

utama berupa fitur *hot reload*, yang memungkinkan pengembang melihat perubahan kode secara langsung tanpa harus melakukan *build* ulang seluruh aplikasi. Fitur ini mempercepat proses pengembangan dan *debugging*, terutama ketika membangun antarmuka atau menguji logika algoritma seperti *Fisher-Yates Shuffling* untuk mengacak urutan soal ujian. *Flutter* juga memiliki pustaka *widget* bawaan yang sangat lengkap, baik untuk tampilan bergaya *Material Design* (Android) maupun *Cupertino* (iOS), sehingga pengembang dapat membuat tampilan profesional dengan cepat dan konsisten.

Flutter memiliki ekosistem yang terus berkembang dengan dukungan berbagai *package* seperti *supabase_flutter* untuk koneksi langsung ke basis data *Supabase*, serta *http* dan *getx* untuk pengelolaan API dan *state management*. Menurut Wibisono dan Hidayah (2024), integrasi *Flutter* dengan *Supabase* mempermudah pengembang dalam membangun aplikasi edukasi yang aman, *real-time*, dan ringan. Dengan menggunakan *Flutter*, aplikasi ujian *online* dapat dikembangkan secara efisien untuk digunakan di berbagai perangkat siswa tanpa perlu menulis ulang kode untuk setiap sistem operasi.

2.7 Database

Database adalah sebuah sistem yang merupakan wadah untuk menyimpan, mengelola, dan mengakses data secara efisien dan terstruktur. Dalam pengembangan sebuah aplikasi ujian berbasis *mobile* database merupakan bagian penting karena database sendiri berfungsi untuk wadah bagi informasi pengguna, hasil ujian, dan aktivitas sistem (Handayani et al., 2020). Dalam pengembangan sebuah aplikasi *mobile*, pemilihan jenis database sangat penting

untuk mempertimbangkan efisiensi penyimpanan, keamanan data, serta kemudahan untuk integrasi dengan *backend*.

Secara umum, penggunaan database pada aplikasi Android terbagi menjadi dua kategori: lokal yang merupakan database *offline* dan *cloud-based* yang merupakan database *online*. Untuk penyimpanan lokal, *SQLite* merupakan pilihan yang paling umum karena sudah terintegrasi langsung ke dalam sistem Android dan tidak memerlukan server eksternal. Menurut Suharto et al (2022) *SQLite* cocok digunakan untuk aplikasi dengan kebutuhan penyimpanan data sementara. Sementara itu, untuk aplikasi dengan banyak pengguna dan kebutuhan sinkronisasi data secara *real-time*, solusi yang disarankan adalah dengan menggunakan database berbasis *cloud*.

Salah satu solusi backend as a service modern yang populer saat ini adalah *Supabase*. *Supabase* merupakan platform *open source* yang dibangun dengan menggunakan PostgreSQL sebagai basis utamanya, serta menyediakan fitur autentikasi, penyimpanan file, dan API otomatis (Hidayat & Maulana, 2023). Berbeda dengan *Firebase* yang bersifat tertutup (*proprietary*), *Supabase* bersifat terbuka dan memberikan fleksibilitas penuh kepada pengembang untuk mengelola server mereka sendiri. Menurut Wulandari dan Pratama (2024), *Supabase* sangat cocok untuk aplikasi ujian *online* karena mendukung *real-time data synchronization*, *row-level security*, dan integrasi mudah dengan Android Studio melalui *RESTful API*.

Keunggulan lain *Supabase* adalah kemampuannya dalam memberikan *auto-generated API* untuk setiap tabel yang dibuat di PostgreSQL. Hal ini memungkinkan komunikasi langsung antara aplikasi Android dan database

tanpa perlu menulis kode *backend* tambahan. Selain itu, *Supabase* juga menyediakan *dashboard* berbasis web untuk memantau data, autentikasi pengguna, serta mengelola *storage bucket* untuk media seperti gambar atau dokumen hasil ujian (Ramadhan & Yuliani, 2023). Dengan arsitektur berbasis PostgreSQL, *Supabase* mampu memberikan performa yang tinggi serta mendukung *query* kompleks yang tidak dimiliki oleh *Firebase Realtime Database*.

2.8 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa pemodelan standar yang digunakan untuk menggambarkan, merancang, serta mendokumentasikan sistem perangkat lunak berbasis berorientasi objek. Dengan UML, pengembang dapat menggambarkan hubungan antar komponen sistem baik dari sisi fungsionalitas maupun alur data.

Menurut Setiawan dan Nurhayati (2022), UML tidak hanya membantu dalam memvisualisasikan rancangan sistem, tetapi juga memfasilitasi komunikasi antara pengembang, analis sistem, dan pemangku kepentingan (*stakeholder*). Melalui diagram UML, proses pengembangan aplikasi seperti sistem ujian *online* menjadi lebih terstruktur dan terdokumentasi dengan baik.

Dalam konteks pengembangan aplikasi ujian *online* berbasis Android, UML berperan dalam merancang interaksi antara pengguna (siswa dan guru), sistem ujian, serta server database *Supabase* yang digunakan untuk menyimpan data soal dan hasil ujian. Menurut Pratama dan Fadilah (2023), penggunaan UML dapat mempercepat proses implementasi karena memberikan panduan



visual bagi pengembang dalam menulis kode program sesuai kebutuhan sistem.


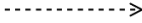
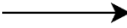

2.8.1 Use Case Diagram

Use Case Diagram merupakan jenis diagram UML yang digunakan untuk menggambarkan hubungan antara aktor (pengguna sistem) dan fungsionalitas utama sistem. Diagram ini menjelaskan apa yang dapat dilakukan oleh pengguna terhadap sistem tanpa menjelaskan bagaimana sistem tersebut bekerja secara teknis (Ramdani & Wulandari, 2020).

Dalam sistem ujian online berbasis Android, aktor yang terlibat antara lain siswa, guru, dan administrator. Siswa dapat melakukan aktivitas seperti login, mengerjakan ujian, dan melihat hasil. Guru dapat membuat serta mengelola soal ujian, sedangkan administrator mengelola data pengguna serta pengaturan sistem. Dengan adanya *use case diagram*, batasan sistem dan peran pengguna dapat didefinisikan dengan jelas (Wijaya & Sari, 2021).

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

NO.	Simbol	Deskripsi
1	Use Case 	Menunjukkan fungsi utama yang dilakukan oleh sistem. Ditulis dalam bentuk kata kerja.
2	Actor 	Menunjukkan pengguna atau entitas luar yang berinteraksi dengan sistem.
3	Association	Garis penghubung antara aktor dan

		use case untuk menunjukkan interaksi.
4	Ekstensi 	Menunjukkan bahwa satu use case dapat memperluas use case lain jika kondisi tertentu terpenuhi.
5	Generalisasi 	Aktor atau use case dapat memiliki turunan.
6	Include 	Menunjukkan bahwa satu use case selalu memanggil use case lain.

2.8.2 Activity Diagram



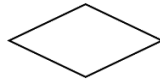


Activity Diagram menggambarkan aliran aktivitas atau proses bisnis dalam sistem secara logis. Diagram ini sering digunakan untuk menjelaskan bagaimana proses tertentu dimulai, berlangsung, hingga selesai (Nugroho & Rahmawati, 2023). Dalam konteks aplikasi ujian *online*, *activity diagram* dapat digunakan untuk menjelaskan alur mulai dari siswa *login* ke aplikasi, memilih ujian, menjawab soal, hingga mengirimkan hasil ujian ke database Supabase.

Menurut Aulia dan Kurniawan (2021), penggunaan *activity diagram* meningkatkan efisiensi dalam mendesain alur sistem karena memberikan gambaran yang jelas mengenai interaksi antar komponen dan pengguna.

Diagram ini membantu pengembang mengidentifikasi titik-titik keputusan (*decision point*), kondisi paralel (*parallel activity*), serta potensi kesalahan yang mungkin terjadi selama proses berlangsung. *Activity diagram* juga berfungsi

sebagai dokumentasi dinamis yang dapat membantu tim pengembang baru memahami proses bisnis sistem secara cepat. Dalam pengembangan aplikasi berbasis Android Studio, *activity diagram* sering diimplementasikan menjadi *flow control* dalam kode menggunakan *intent* dan *activity lifecycle* (Hartono & Yusuf, 2024).

Tabel 2. 2 Simbol-Simbol *Activity Diagram*

No	Simbol	Deskripsi
1	Status Awal 	Titik mulai aktivitas. Hanya ada satu di setiap diagram.
2	Aktivitas 	Proses yang dilakukan sistem atau pengguna.
3	Percabangan 	Menunjukkan percabangan kondisi (ya/tidak).
4	Penggabungan/Join 	Menggabungkan kembali alur paralel menjadi satu.
5	Status Akhir 	Titik akhir proses.

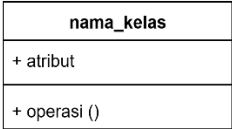
2.8.3 Class Diagram

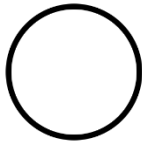

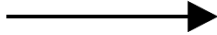
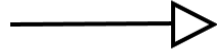

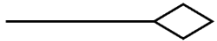
Class Diagram merupakan diagram UML yang menggambarkan struktur statis sistem dengan menunjukkan kelas, atribut, metode, serta hubungan antar kelas. Diagram ini menjadi inti dari desain berorientasi objek karena membantu dalam menentukan bagaimana data dan fungsi dikelompokkan (Gunawan & Lestari, 2020).

Dalam sistem ujian *online*, contoh kelas yang dapat digambarkan meliputi *User*, *Soal*, *Ujian*, *Jawaban*, dan *HasilUjian*. Setiap kelas memiliki atribut dan metode yang berbeda. Misalnya, kelas *User* memiliki atribut *id_user*, *nama*, dan *password*, sementara kelas *Soal* memiliki atribut *id_soal*, *pertanyaan*, dan *kunci_jawaban*. Hubungan antar kelas dapat berupa *association*, *aggregation*, atau *inheritance* tergantung pada kebutuhan sistem (Wibowo & Santoso, 2022).

Menurut Lestari dan Ahmad (2023), *class diagram* sangat penting untuk menentukan struktur tabel dalam database, terutama ketika menggunakan sistem *relational database* seperti PostgreSQL yang menjadi basis Supabase. Dengan *class diagram*, proses pembuatan tabel dan relasi menjadi lebih efisien serta konsisten dengan model data yang diinginkan.

Tabel 2. 3 Simbol-Simbol Class Diagram

No	Simbol	Deskripsi
1	Kelas 	Kelas pada struktur sistem.

2	Interface 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3	Association 	Relasi antarclass dengan arti umum, asosiasi biasanya juga disertai dengan <i>Multiplicity</i> .
4	Directed Association 	Relasi antarkelas dengan makna kelas yang atau digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6	Dependency 	Relasi antarkelas dengan makna kebergantungan antarkelas.
7	Aggregation 	Relasi antarkelas dengan makna semua bagian (whole-part).

2.9 Profil SMP Negeri 1 Karimun

SMP Negeri 1 Karimun salah satu sekolah menengah pertama yang berada di Kabupaten Karimun. Sekolah menengah pertama ini tepatnya berlokasi di Jl. Pendidikan No. 100, Tanjung Balai Kota Kecamatan Karimun, Kabupaten Karimun - Provinsi Kepulauan Riau. Sekolah ini pertama kali berdiri pada tanggal 1 Agustus 1957 dengan Nomor SK Pendirian 3480/B/III berada dalam naungan Kementerian Pendidikan dan Kebudayaan.

2.9.1 Logo Sekolah

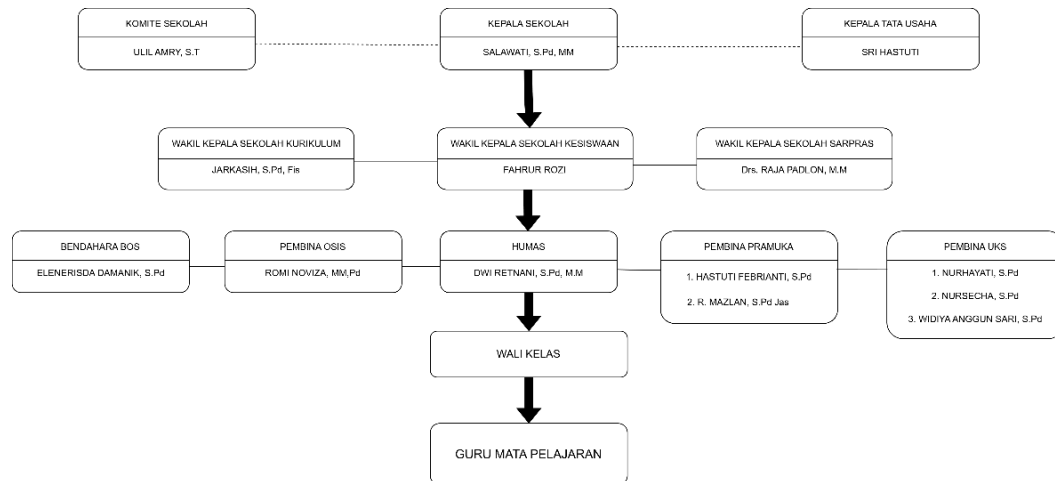
Berikut merupakan lambang sekolah SMP Negeri 1 Karimun



Gambar 2. 1 Logo SMP Negeri 1 Karimun

2.9.2 Struktur Organisasi Sekolah

Berikut merupakan struktur organisasi yang terdapat di SMP Negeri 1 Karimun.



Gambar 2. 2 Struktur Organisasi SMP Negeri 1 Karimun

2.9.3 Karakteristik Murid

SMP Negeri 1 Karimun memiliki keunggulan di bidang akademik maupun ekstrakurikuler. Untuk bidang akademik SMP N 1 Karimun bidang OSN masuk tingkat provinsi, sedangkan untuk bidang ekstrakurikuler FLS2N masuk tingkat provinsi untuk setiap tahunnya. Berdasarkan data rapor Pendidikan kemampuan literasi murid berpredikat Baik (100% murid sudah mencapai kompetensi minimum). Berdasarkan data agama yang di anutnya sebagai berikut: agama Islam 98,55 %, Protestan 0,72 %, Katolik 0,36 % dan Budha 0,36 %. Berdasarkan data gaya belajar sebagai berikut: Visual 42,80 %, Audiotory 42,80 % dan Kinestetik 19,78%.

SMP Negeri 1 Karimun terdiri atas tiga tingkatan kelas, yaitu kelas VII, VIII, dan IX. Setiap tingkatan terdiri dari delapan rombongan belajar (kelas) dengan jumlah peserta didik yang bervariasi pada tiap jenjang. Pada tahun ajaran berjalan, jumlah siswa kelas VII sebanyak 278 orang, kelas VIII sebanyak 258 orang, dan kelas IX sebanyak 278 orang. Dengan total keseluruhan 814 siswa.

2.9.4 Karakteristik Pendidik

Tenaga pendidik di SMP Negeri 1 Karimun terdiri atas 46 orang guru yang memiliki latar belakang dan status kepegawaian beragam. Berdasarkan jenis kelamin, terdapat 8 orang guru laki-laki dan 38 orang guru perempuan. Dari segi status kepegawaian, sebanyak 23 orang guru berstatus Pegawai Negeri Sipil (PNS), 22 orang berstatus Pegawai Pemerintah dengan Perjanjian Kerja (PPPK), serta 1 orang guru berstatus honorer.

2.9.5 Kurikulum Sekolah

Kurikulum yang dilaksanakan di SMP Negeri 1 Karimun ini menggunakan kurikulum “pendekatan *deep learning*”. Murid dapat mencapai prestasi akademik yang lebih tinggi dan mengembangkan kemampuan sosial serta emosional yang baik. Kurikulum Sekolah ini diberlakukan pada Tahun Ajaran 2025/2026 yang mencerminkan merdeka belajar dan pengimplementasian profil pelajar Pancasila. Kurikulum ini memuat karakteristik satuan pendidikan, profil pembelajar, struktur kurikulum dan rancangan pembelajaran.

2.10 Penelitian Terkait

Penelitian mengenai pengacakan soal ujian online telah banyak dilakukan dalam beberapa tahun terakhir, terutama dengan meningkatnya penggunaan *Computer Based Test (CBT)* di lembaga pendidikan. Salah satu algoritma yang sering digunakan untuk mengatasi permasalahan kecurangan akibat kesamaan urutan soal adalah algoritma *Fisher-Yates Shuffling*. Algoritma ini dikenal efisien dalam melakukan pengacakan elemen data secara merata dengan kompleksitas

waktu $O(n)$, sehingga cocok digunakan dalam sistem ujian online yang membutuhkan kecepatan dan keadilan dalam penyajian soal. Beberapa penelitian terdahulu telah membuktikan efektivitas algoritma ini dalam konteks pembelajaran digital dan sistem ujian berbasis Android maupun web.

Berikut merupakan beberapa penelitian yang terkait dengan topik yang penulis angkat:

1. Penelitian yang dilakukan oleh Subagja dan Aziz (2022) berjudul “Penerapan Algoritma *Fisher-Yates* Pada *Computer Based Test* di Panda Mandarin *Course*” merupakan salah satu studi yang secara langsung menerapkan algoritma ini dalam sistem ujian berbasis komputer. Hasil penelitian mereka menunjukkan bahwa penggunaan *Fisher-Yates* dapat mengacak urutan soal dengan distribusi yang adil, sehingga setiap peserta mendapatkan variasi soal yang berbeda tanpa mengubah tingkat kesulitannya. Penerapan algoritma ini terbukti mampu meningkatkan keadilan dan mengurangi kemungkinan kecurangan dalam pelaksanaan ujian berbasis komputer.
2. Pratama dan Wibowo (2021) melakukan penelitian berjudul “Implementasi Algoritma *Fisher-Yates Shuffle* pada Sistem Ujian *Online* Berbasis Web” yang bertujuan untuk menguji efisiensi pengacakan data menggunakan *Fisher-Yates* pada sistem ujian berbasis web. Hasil penelitian menunjukkan bahwa algoritma ini mampu menghasilkan urutan acak yang unik dengan waktu eksekusi kurang dari 0,002 detik untuk setiap set soal. Hal ini menunjukkan bahwa *Fisher-Yates* tidak hanya memberikan hasil acak yang merata, tetapi juga unggul dalam hal kecepatan dan efisiensi komputasi

dibandingkan metode pengacakan sederhana berbasis fungsi *random()* bawaan bahasa pemrograman.

3. Penelitian lain dilakukan oleh Rahmadani dan Lestari (2023) dalam jurnal Jurnal Sistem dan Teknologi Informasi dengan judul “Penggunaan *Fisher-Yates Algorithm* untuk Pengacakan Soal pada Aplikasi Ujian Berbasis Android.” Mereka mengimplementasikan algoritma ini pada sistem ujian berbasis Android dengan integrasi ke dalam *backend* Firebase. Hasil eksperimen menunjukkan bahwa *Fisher-Yates* mampu menjaga konsistensi pengacakan antar pengguna serta meningkatkan kecepatan pemuatan soal pada perangkat *mobile*. Dengan demikian, algoritma ini dapat diandalkan dalam sistem ujian *mobile* yang menuntut performa tinggi dan sinkronisasi data *real-time*.
4. Yuliana dan Setiawan (2020) dalam penelitiannya “Analisis Kinerja *Fisher-Yates Shuffle* dalam Pengacakan Data pada Sistem Evaluasi *Online*” melakukan analisis perbandingan antara *Fisher-Yates* dengan metode swap acak berganda. Hasilnya menunjukkan bahwa *Fisher-Yates* memiliki performa yang lebih efisien secara waktu dan memori. Kompleksitas $O(n)$ yang dimilikinya membuat algoritma ini lebih stabil dan dapat diterapkan dalam sistem ujian yang memiliki ribuan soal tanpa penurunan kinerja yang signifikan. Penelitian ini juga menegaskan bahwa pengacakan *Fisher-Yates* menghasilkan distribusi urutan yang lebih acak secara statistik.
5. Putra dan Hidayat (2024) dalam jurnal Jurnal Informatika dan Sistem Terdistribusi meneliti “Optimalisasi Pengacakan Soal Ujian Menggunakan *Fisher-Yates Shuffle Algorithm* pada Supabase *Cloud Database*.” Studi ini

menggunakan pendekatan *cloud computing* dengan memanfaatkan Supabase sebagai *backend service*. Implementasi algoritma *Fisher-Yates* dilakukan di sisi server untuk memastikan konsistensi pengacakan antar pengguna yang mengakses soal secara bersamaan. Hasil penelitian menunjukkan bahwa integrasi antara algoritma ini dan Supabase menghasilkan efisiensi tinggi, dengan waktu pemrosesan di bawah 10 milidetik untuk 1.000 soal, serta sinkronisasi data real-time yang mendukung transparansi hasil ujian.

6. Fauzan dan Nugraha (2023) dalam penelitian berjudul “*Comparative Study: Fisher-Yates Shuffle vs Randomized Block Algorithm for CBT Systems*” membandingkan dua metode pengacakan dalam sistem CBT. Mereka menemukan bahwa Fisher-Yates lebih unggul dalam menghasilkan variasi urutan soal yang benar-benar acak dibandingkan metode *Randomized Block*, yang cenderung mengelompokkan soal berdasarkan kategori tertentu. Fisher-Yates juga terbukti lebih efisien dari sisi kecepatan dan konsumsi memori, sehingga lebih cocok diterapkan pada aplikasi ujian skala besar.