

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Tinjauan penelitian terdahulu berisi ulasan studi serupa yang pernah dilakukan. Tujuannya adalah untuk menemukan perbedaan atau sebagai pengembangan dari penelitian sebelumnya, sehingga keaslian atau nilai baru dari penelitian ini menjadi jelas.

Penelitian Kurniawan dan Sauda (2021) membahas penggunaan algoritma Fisher-Yates Shuffle untuk mengacak urutan huruf dalam aplikasi pembelajaran huruf hijaiyah bagi pendidikan usia dini. Aplikasi ini dirancang untuk membantu anak-anak mengenal huruf hijaiyah secara interaktif dan menyenangkan, sehingga lebih praktis dan efektif dalam mempersiapkan mereka menghadapi jenjang pendidikan selanjutnya. Penggunaan algoritma ini membuat proses belajar menjadi lebih dinamis dan menarik bagi anak-anak. Kesimpulan dari penelitian ini menunjukkan bahwa implementasi algoritma Fisher Yates Shuffle dalam aplikasi belajar huruf hijaiyah berhasil mengacak pertanyaan kuis secara efektif, sehingga mencegah pengulangan pertanyaan. Aplikasi ini dirancang sebagai media pembelajaran interaktif untuk anak-anak, dengan fokus pada pengenalan huruf hijaiyah melalui perangkat mobile berbasis Android. Penggunaan algoritma Fisher Yates Shuffle dalam aplikasi ini memberikan kontribusi nyata dalam memfasilitasi pendidikan agama bagi anak usia dini.

Penelitian Damanik (2019) membahas penggunaan algoritma Fisher-Yates untuk mengacak himpunan kata dalam game edukasi berbasis Android. Game ini dirancang untuk meningkatkan kemampuan bahasa Inggris anak-anak melalui pengalaman bermain yang interaktif dan kompetitif. Meskipun game ini mendukung interaksi yang baik, tampilan UI/UX masih dianggap kurang memadai, sehingga perlu perbaikan untuk meningkatkan pengalaman pengguna secara keseluruhan. Kesimpulan dari

penelitian ini adalah bahwa penerapan algoritma Fisher Yates dalam game edukasi efektif dalam meningkatkan pengenalan kosakata bahasa Inggris pada anak-anak. Game ini dirancang untuk meningkatkan keterampilan bahasa Inggris pengguna melalui permainan yang interaktif dan menyenangkan, dengan berbagai level yang semakin sulit sesuai dengan perkembangan pemain.

Penelitian Setiawan (2022) membahas penerapan algoritma Fisher-Yates dalam game edukasi berbasis Android yang bertujuan untuk memperkenalkan senjata tradisional Indonesia kepada anak-anak. Dengan kurangnya teknologi edukasi yang tersedia, game edukasi dipandang sebagai solusi yang tepat untuk meningkatkan kemampuan motorik, afeksi, kognitif, dan keseimbangan anak. Algoritma Fisher-Yates digunakan untuk mengacak soal dan jawaban dalam game, sedangkan bahasa pemrograman yang digunakan adalah JavaScript dan HTML. Hasil uji coba menunjukkan bahwa metode ini bekerja dengan baik dan optimal untuk meningkatkan interaksi dalam game edukasi tersebut. Kesimpulan dari penelitian ini menunjukkan bahwa penerapan metode Fisher Yates dalam aplikasi game edukasi tentang alat perang tradisional Indonesia berhasil mengacak pertanyaan dan jawaban dengan efektif. Game ini dirancang untuk meningkatkan pengetahuan anak-anak tentang senjata tradisional Indonesia melalui pengalaman belajar yang menyenangkan dan interaktif, menggunakan teknologi JavaScript dan HTML.

Hasil penelitian menunjukkan bahwa game ini berhasil meningkatkan pengetahuan anak-anak tentang alat perang tradisional, dengan skor rata-rata pengetahuan sebesar 3,58 dan penilaian pengguna yang menyatakan aplikasi ini tergolong "Baik." Metode Fisher Yates yang digunakan dalam game ini memungkinkan pengacakan pertanyaan dan jawaban secara efektif, sehingga meningkatkan dinamika permainan dan mencegah pengulangan yang monoton.

Penelitian Mariani dan Witanti (2023) membahas penerapan algoritma Fisher-Yates untuk mengacak urutan soal pada tes berbasis web. Randomisasi soal sangat penting untuk mengukur hasil belajar siswa secara adil. Algoritma Fisher-Yates terbukti efektif dalam menghasilkan permutasi acak pada elemen array, termasuk soal tes. Sistem ini dirancang menggunakan Unified Modeling Language (UML) dengan berbagai diagram, dan diuji menggunakan metode black-box, yang menunjukkan bahwa sistem bekerja dengan baik. Hasil pengacakan pada 30 soal tes menghasilkan Permutasian soal yang berbeda dengan akurasi 94.43%, serta distribusi yang lebih merata dan lebih sedikit pengulangan. Kesimpulan dari penelitian ini menunjukkan bahwa penerapan algoritma Fisher-Yates dalam permutasi acak pada soal tes berbasis web berhasil mengacak pertanyaan dengan efektif, mencapai tingkat akurasi sebesar 94,43%. Algoritma ini mampu menghasilkan urutan soal yang berbeda untuk setiap peserta ujian, sehingga mengurangi kemungkinan kecurangan dengan memberikan variasi urutan soal pada setiap sesi ujian.

Dari kajian beberapa laporan yang telah dibahas di atas, dapat disimpulkan bahwa penggunaan algoritma Fisher-Yates Shuffle secara konsisten diterapkan untuk mengacak elemen-elemen, baik dalam aplikasi edukasi maupun sistem pengacakan soal tes. Setiap laporan menunjukkan bahwa algoritma ini memiliki keunggulan dalam menghasilkan urutan acak yang efisien dan efektif, seperti yang terlihat dalam aplikasi pembelajaran huruf hijaiyah, pengenalan kosakata bahasa Inggris, game edukasi, dan sistem acak soal tes.

Penelitian oleh Kurniawan dan Sauda (2021) serta Damanik (2019) menekankan pada interaktivitas dan pengalaman belajar yang dinamis bagi anak-anak, sedangkan laporan oleh Setiawan (2022) dan Mariani serta Witanti (2023) lebih fokus pada penerapan acak soal yang adil dan merata, baik dalam konteks game edukasi maupun sistem tes berbasis web. Kesamaan dari semua penelitian ini terletak pada penggunaan algoritma Fisher-Yates untuk meningkatkan pengalaman pengguna dan memastikan variasi acak yang optimal.

Maka dari itu, penelitian ini akan mengembangkan sistem pengelolaan kompetisi pada aplikasi GOHit dengan mengintegrasikan algoritma Fisher-Yates untuk acak data peserta. Penggunaan algoritma ini diharapkan dapat membantu penyelenggara dalam menentukan peserta secara acak dan lebih adil, seperti yang telah dibuktikan dari penelitian sebelumnya.

2.2 Teori Terkait

2.2.1 *Fisher-Yates*

Fisher-Yates Shuffle (atau Knuth Shuffle) adalah algoritma yang digunakan untuk melakukan permutasi acak terhadap elemen-elemen dalam suatu himpunan terhingga (Rizki & Op, 2021; Tinambunan & Sintaro, 2021). Prinsip kerjanya mirip dengan mengambil undian bernomor dari sebuah wadah secara sekuensial (Aziz et al., 2021; Puspaningrum & Susanto, 2021). Variasi dari algoritma ini, yang dikenal sebagai algoritma Sattolo, secara spesifik dapat menghasilkan siklus acak (Surahman, Aditama, et al., 2021; Siregar & Hambali, 2020).

Proses pengacakan dengan Fisher-Yates melibatkan beberapa langkah. Pertama, siapkan daftar berisi N elemen. Kemudian, secara iteratif, sebuah elemen dipilih secara acak dari daftar elemen yang tersisa, lalu dipindahkan ke urutan hasil yang baru. Proses ini terus berlanjut hingga seluruh elemen dari daftar awal telah dipindahkan, menghasilkan sebuah urutan baru yang teracak secara acak.

Tabel 2.1 Proses Visual Pengacakan Fisher-Yates

| Range | Roll | Scratch | Result |
|-------|----------|----------------------------------|-----------------------------|
| | | 1 2 3 4 5 6 7 8 9 10 | |
| 1-10 | 6 | 1 2 3 4 5 10 7 8 9 | 6 |
| 1-9 | 3 | 1 2 9 4 5 10 7 8 | 3 6 |
| 1-8 | 7 | 1 2 9 4 5 10 8 | 7 3 6 |
| 1-7 | 5 | 1 2 9 4 8 10 | 5 7 3 6 |
| 1-6 | 6 | 1 2 9 4 8 | 10 5 7 3 6 |
| 1-5 | 2 | 1 8 9 4 | 2 10 5 7 3 6 |
| 1-4 | 1 | 4 8 9 | 1 2 10 5 7 3 6 |
| 1-3 | 2 | 4 9 | 8 1 2 10 5 7 3 6 |
| 1-2 | 2 | 4 | 9 8 1 2 10 5 7 3 6 |
| | | | 4 9 8 1 2 10 5 7 3 6 |

Untuk setiap indeks i dari $n - 1$ ke 1:

Ambil bilangan acak j antara 0 dan i

Tukar elemen $A[i]$ dan $A[j]$

Gambar 2.1 Rumus Fisher-Yates

2.2.2 CodeIgniter

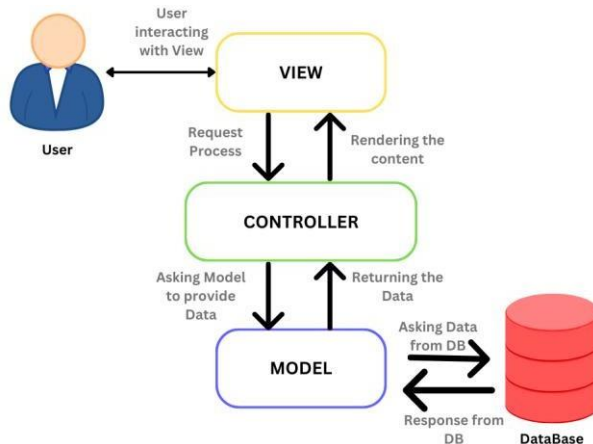
CodeIgniter adalah sebuah kerangka kerja (framework) PHP yang memanfaatkan konsep MVC (Model, View, Controller) untuk mempermudah pengembangan situs dengan menyediakan sumber daya yang lengkap (Sulistiani & Hendra Saputra, 2020). Pada intinya, CodeIgniter menggunakan pola MVC untuk memisahkan logika aplikasi dari bagian tampilan. Pendekatan ini membuat halaman web hanya berisi sedikit skrip PHP karena tampilan (View) sudah terpisah.

Pemisahan ini secara praktis membuat halaman web lebih rapi karena skrip PHP tidak banyak tercampur di dalam kode tampilan.

Struktur MVC dalam CodeIgniter terdiri dari tiga komponen utama:

- **Model:** Bertugas mengelola struktur data aplikasi. Komponen ini biasanya berisi fungsi-fungsi untuk berinteraksi dengan database, seperti mengambil, menyimpan, dan memperbarui data.
- **View:** Merupakan segala sesuatu yang ditampilkan kepada pengguna. Ini bisa berupa halaman web utuh, bagian-bagian halaman (seperti header dan footer), atau bahkan format lain seperti halaman RSS.
- **Controller:** Berperan sebagai perantara yang menghubungkan Model, View, dan sumber daya lain yang diperlukan. Controller memproses permintaan dari pengguna dan menghasilkan halaman web sebagai respons.

CodeIgniter menerapkan MVC secara fleksibel karena penggunaan Model bersifat opsional. Jika tidak diperlukan, pengembang bisa membangun aplikasi hanya dengan Controller dan View untuk menghindari kompleksitas yang tidak perlu. Framework ini juga mengizinkan pengembang untuk mengintegrasikan skrip yang sudah ada atau membuat pustaka sendiri, sehingga memungkinkan alur kerja yang sangat bisa disesuaikan (CodeIgniter, 2025).



Gambar 2.2 diagram arsitektur mvc

2.2.3 Flowchart

Flowchart merupakan gambaran visual berbentuk grafik yang menyajikan langkah-langkah serta urutan dari sebuah prosedur program. Diagram ini membantu dalam proses analisis, perancangan, dan pengkodean dengan cara memecah masalah menjadi bagian-bagian operasional yang lebih kecil, sehingga biasanya mempermudah penyelesaian masalah pada tahap evaluasi. Dengan kata lain, flowchart adalah diagram dengan simbol-simbol grafis yang menyatakan alur proses dan menampilkan urutan langkah dari suatu prosedur program dengan fungsi spesifik. Fungsi utamanya adalah untuk memberikan gambaran proses agar mudah dipahami berdasarkan urutan langkahnya. Hal ini pada akhirnya menyederhanakan rangkaian proses sehingga pengguna lebih mudah memahami informasi yang disampaikan (Budiman, Saori, Anwar, Fitriani, & Pangestu, 2021).

| | | | |
|--|---|--|--|
| | Flow Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line. | | Input/output Simbol yang menyatakan proses input atau output tanpa tergantung peralatan. |
| | On-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama. | | Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer. |
| | Off-Page Reference Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda. | | Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak. |
| | Terminator Simbol yang menyatakan awal atau akhir suatu program. | | Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur. |
| | Process Simbol yang menyatakan suatu proses yang dilakukan komputer. | | Display Simbol yang menyatakan peralatan output yang digunakan. |
| | Decision Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak. | | Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal. |



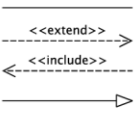
Gambar 2.3 Simbol dan Notasi Flowchart

2.2.4 Use Case Diagram

Menurut Musthofa & Adiguna (2022), Use Case Diagram adalah sebuah model visual yang menggambarkan perilaku (behavior) dari sebuah sistem informasi yang akan dikembangkan. Tujuan utamanya adalah untuk mengidentifikasi semua fungsi yang akan ada di dalam sistem tersebut dan menentukan siapa saja (aktor) yang berhak untuk mengakses dan menggunakan setiap fungsi.

Secara sederhana, diagram ini merupakan representasi grafis yang menunjukkan hubungan dan interaksi antara satu atau lebih aktor dengan sistem. Dengan demikian, use case diagram menjadi alat yang efektif untuk memahami seluruh fungsionalitas yang dimiliki oleh sebuah sistem.

Tujuan dari dibuatnya use case diagram yaitu untuk mempermudah client atau pihak yang memiliki kepentingan terhadap sistem tersebut untuk memahami dan berkomunikasi terkait rancangan dari alur sistem yang akan dibuat, selain itu juga untuk memudahkan pemahaman akan requirement atau kebutuhan sistem. Berikut merupakan simbol yang ada pada use case diagram :


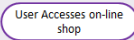
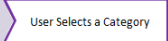
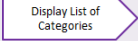
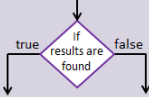


| Symbol | Reference Name |
|---|----------------|
|  | Actor |
|  | Use case |
|  | Relationship |

Gambar 2.4 Simbol Use Case Diagram

2.2.5 Activity Diagram

Menurut Musthofa & Adiguna (2022), Activity Diagram adalah sebuah diagram yang menggambarkan alur kerja (workflow) atau aktivitas dari sebuah sistem informasi yang sedang dikembangkan. Selain itu, diagram ini juga bisa dimanfaatkan untuk memodelkan proses bisnis atau alur menu yang ada pada sistem tersebut.

Fungsi utama dari Activity Diagram adalah untuk memperlihatkan urutan aktivitas dalam sebuah proses secara jelas, sehingga membantu klien atau pengguna dalam memahami cara kerja sistem secara keseluruhan. Diagram ini biasanya digunakan untuk mendefinisikan atau mengelompokkan alur dan tampilan sistem, serta dibuat berdasarkan pada satu atau beberapa Use Case yang telah ditentukan. Activity Diagram sendiri menggunakan serangkaian simbol komponen untuk visualisasinya. Berikut merupakan komponen simbol yang ada pada Activity diagram :

| Activity Diagram / Shape: | Meaning: |
|---|-------------------------|
|  | Starting Point |
|  | Activity |
|  | Input |
|  | Output |
|  | Decision |
|  | Simultaneous Activities |
|  | End Point |

Gambar 2.5 Simbol *Activity Diagram*

2.2.6 Pemodelan Data (*Data Modeling*)

Menurut Fikry (2019), pemodelan data merupakan kerangka konseptual yang digunakan untuk mendefinisikan struktur sebuah basis data dan menjelaskan berbagai tingkatan abstraksi data.

Dengan kata lain, pemodelan data adalah proses untuk menganalisis kebutuhan data guna mendukung proses bisnis yang ada, kemudian merancang sebuah diagram deskriptif yang memetakan hubungan antar berbagai jenis informasi tersebut. Hasil akhir dari proses pemodelan ini adalah sebuah rancangan basis data yang terstruktur.

Jenis model data yang digunakan dalam pemodelan data yaitu sebagai berikut :

1. CDM (Conceptual Data Model)

Menurut Shandy Ardianto (2012), Conceptual Data Model (CDM) adalah sebuah model yang memandang dunia nyata sebagai kumpulan objek dasar yang disebut entitas beserta hubungan (relationship) yang terjadi di antara entitas-entitas tersebut. Model ini umumnya divisualisasikan dalam bentuk Entity Relationship Diagram (ERD). Manfaat utama dari penggunaan CDM dalam perancangan database adalah kemampuannya untuk menyajikan gambaran lengkap dan detail mengenai struktur logis basis data—mencakup arti, relasi, dan batasan data—tanpa bergantung pada perangkat lunak (software) atau model struktur data spesifik yang akan digunakan.

2. PDM (Physical Data Model)

Menurut Shandy Ardianto (2012), Physical Data Model (PDM) adalah model yang merepresentasikan desain fisik sebuah database menggunakan struktur tabel. Model ini menggambarkan data dan hubungan antar data secara konkret, di mana setiap tabel memiliki kolom-kolom dengan nama unik. PDM merupakan implementasi fisik dari sebuah rancangan database yang dibuat dengan mempertimbangkan sistem manajemen basis data (DBMS) spesifik yang akan digunakan. Model ini dapat dihasilkan dari Conceptual Data Model (CDM) yang valid dan fungsinya setara dengan skema relasi. Tujuan utamanya adalah untuk memodelkan struktur penyimpanan data secara detail dan benar sesuai dengan yang akan diterapkan pada basis data sesungguhnya.

2.2.7 WORLD RANKING SYSTEM

Beberapa kompetisi berskala nasional maupun internasional telah memiliki panduan resmi mengenai sistem pemeringkatan pemain untuk mendukung penyusunan braket yang adil dan terstruktur. Hal ini bertujuan untuk mencegah pertemuan pemain unggulan di babak awal dan memastikan distribusi lawan yang seimbang. Menurut sistem ranking dari BWF [2018], pemain mendapatkan poin berdasarkan hasil turnamen yang mereka ikuti dalam 52 minggu terakhir. Poin yang diperoleh bergantung pada level turnamen dan sejauh mana pemain melaju dalam kompetisi.

Selain itu, dalam proses penentuan lawan, pemain dengan peringkat lebih tinggi tidak akan bertemu di babak awal untuk memastikan distribusi pertandingan yang lebih seimbang. Badminton World Federation, BWF Statutes, Section 5.3.3.1: World Ranking System, In Force: 01/01/2018. [Online].