# BAB 2
# LANDASAN TEORI

## 2.1 Tinjauan Penelitian Terdahulu

Penelitian mengenai *game* edukasi memasak telah dilakukan oleh beberapa peneliti sebelumnya. Kartikasari dan Hilmi [4] mengembangkan *game* edukasi memasak *single player* berbasis Android menggunakan Unity 3D dan Blender dengan judul *Cooking Chaos*. Game tersebut menyajikan 11 variasi resep dari menu Burger dan Sate Ayam. Hasil pengujian menunjukkan bahwa game ini berhasil meningkatkan minat belajar memasak pengguna sebesar 80,3%. Namun, penelitian tersebut belum mengimplementasikan algoritma pengacakan untuk mengatur urutan kemunculan resep sehingga urutan resep bersifat tetap di setiap sesi permainan.

Azhari et al. [5] merancang *game* simulasi memasak menggunakan framework *Design, Play, and Experience* (DPE) yang berfokus pada prosedur memasak di industri makanan dan minuman selama pandemi COVID-19. Pengujian terhadap 15 responden menunjukkan tingkat pengalaman simulasi sebesar 88,70% dan tingkat pengalaman belajar sebesar 84,87% dengan kriteria sangat sesuai. Game ini dikembangkan menggunakan Godot Engine, bukan Unity Engine.

Hasnawati dan Asriadi [7] mengembangkan *game* memasak kue tradisional Bugis menggunakan Construct 2 berbasis Android. Game ini bersifat edukatif dan memperkenalkan budaya kuliner lokal kepada pengguna. Penelitian ini menjadi salah satu representasi dominasi Construct 2 dalam genre *game* memasak di Indonesia.

Widjaja [10] mengembangkan *Dishcover Indonesia*, sebuah *game* memasak berbasis Android yang memperkenalkan masakan khas Indonesia. Penelitian ini menjadi referensi dalam perancangan mekanik *game* memasak yang berorientasi pada pengenalan budaya kuliner.

Untuk implementasi algoritma Fisher-Yates *Shuffle*, Rosman et al. [11] mengimplementasikan algoritma ini pada *game* edukasi *adventure* klasifikasi jenis hewan berdasarkan makanannya. Kannabi dan Norhikmah [12] menerapkan Fisher-Yates pada *game* pembelajaran huruf Hijaiyah menggunakan Construct 2 untuk mengacak urutan soal. Harsadi et al. [13] mengimplementasikan Fisher-Yates pada *game* edukasi aksara Jawa menggunakan Godot Engine. Ketiga penelitian tersebut menerapkan Fisher-Yates untuk mengacak soal atau pertanyaan, bukan

untuk mengacak urutan resep dalam *game* simulasi memasak.

Beberapa penelitian lain turut mengembangkan game bertema memasak dan edukasi makanan, seperti game memasak makanan khas Sumatera [14], game simulasi penyajian kuliner tradisional Indonesia [15], dan game edukasi makanan sehat berbasis Construct [16, 17]. Pada tingkat internasional, serious game terbukti efektif meningkatkan pengetahuan gizi pada anak [18, 19]. Tinjauan terhadap genre game memasak secara umum juga telah dilakukan oleh peneliti terdahulu [20].

Untuk memperkuat identifikasi *research gap*, dilakukan tinjauan sistematis terhadap 50 studi ilmiah dalam rentang tahun 2020–2026 yang relevan dengan *game* simulasi memasak, *game* edukasi bertema makanan, serta penerapan algoritma Fisher–Yates *Shuffle*. Matriks lengkap tinjauan tersebut disajikan pada Lampiran 7. Hasil tinjauan menunjukkan bahwa Construct 2 mendominasi pengembangan *game* edukasi bertema makanan (14 dari 50 studi), sementara tidak ditemukan satu pun studi yang mengombinasikan Unity Engine dengan *genre* simulasi memasak. Selain itu, algoritma Fisher–Yates *Shuffle* hanya ditemukan pada *genre* edukasi lain dan belum pernah diterapkan pada *game* bertema memasak.

Perbandingan penelitian terdahulu dengan penelitian ini dapat dilihat pada Tabel 2.1.

Tabel 2.1. Perbandingan Penelitian Terdahulu

| **Peneliti** | **Topik** | **Engine** | **Platform** | **Algoritma** |
|---|---|---|---|---|
| Kartikasari & Hilmi (2024) | *Game* memasak burger & sate | Unity 3D | Android | – |
| Azhari et al. (2024) | *Game* simulasi memasak | Godot | Android | – |
| Hasnawati & Asriadi (2022) | *Game* memasak kue Bugis | Construct 2 | Android | – |
| Widjaja (2021) | *Game* memasak Indonesia | – | Android | – |
| Rosman et al. (2025) | *Game* edukasi hewan | – | – | Fisher-Yates |
| Kannabi & Norhikmah (2022) | *Game* Hijaiyah | Construct 2 | – | Fisher-Yates |
| Harsadi et al. (2022) | *Game* aksara Jawa | Godot | – | Fisher-Yates |
| **Penelitian ini** | ***Game* simulasi memasak Indonesia** | **Unity 6** | **Android** | **Fisher-Yates** |

## 2.2 Game

*Game* atau permainan video adalah program komputer yang melibatkan interaksi pengguna dengan antarmuka untuk menghasilkan umpan balik visual pada perangkat tampilan. Menurut Prensky [21], *game* memiliki enam elemen struktural, yaitu aturan, tujuan, hasil dan umpan balik, konflik atau tantangan, interaksi, dan representasi atau cerita. *Game* terbagi menjadi berbagai genre, salah satunya adalah *game* simulasi yang merepresentasikan situasi atau aktivitas nyata dalam lingkungan virtual.

## 2.3 Game Simulasi Memasak

*Game* simulasi memasak adalah subgenre dari *game* simulasi di mana pemain melakukan aktivitas memasak secara virtual. Pemain diberikan instruksi resep dan harus mengikuti langkah-langkah yang diberikan untuk menyelesaikan masakan. Genre ini menggabungkan unsur hiburan dan edukasi sehingga pemain dapat mempelajari proses memasak secara interaktif. Kartikasari dan Hilmi [4] menyatakan bahwa *game* simulasi memasak terbukti efektif dalam meningkatkan minat belajar memasak pengguna dengan mekanisme *ingredient assembly* yang interaktif.

## 2.4 Unity Engine

Unity adalah *game engine* lintas platform yang dikembangkan oleh Unity Technologies. Unity mendukung pengembangan *game* 2D dan 3D untuk berbagai platform termasuk PC, Android, iOS, dan konsol. Unity menggunakan bahasa pemrograman C# sebagai bahasa skrip utama dan menyediakan berbagai fitur bawaan seperti *physics engine*, *animator*, sistem UI, dan *Scriptable Objects* yang memudahkan manajemen data *game*. Unity banyak digunakan dalam pengembangan *game* lintas platform karena dukungan terhadap pengembangan 2D dan 3D, ekosistem aset yang luas, serta kemudahan *deployment* ke platform Android.

## 2.5 Algoritma Fisher-Yates *Shuffle*

Algoritma Fisher-Yates *Shuffle* adalah algoritma pengacakan yang digunakan untuk menghasilkan permutasi acak dari suatu himpunan terhingga. Algoritma ini pertama kali diperkenalkan oleh Ronald Fisher dan Frank Yates pada tahun 1938 dalam buku *Statistical Tables for Biological, Agricultural and Medical Research*. Versi yang lebih efisien untuk komputasi kemudian dikembangkan oleh Richard Durstenfeld pada tahun 1964 [11].

Algoritma Fisher-Yates *Shuffle* bekerja dengan cara melakukan iterasi dari elemen terakhir menuju elemen pertama dalam suatu *array*. Pada setiap iterasi ke-$i$, dipilih satu indeks acak $j$ dengan nilai antara 0 hingga $i$, kemudian elemen ke-$i$ dan elemen ke-$j$ ditukar posisinya. Proses ini diulang hingga semua elemen telah diproses sehingga menghasilkan permutasi yang sepenuhnya acak. Kompleksitas waktu algoritma ini adalah $O(n)$, di mana $n$ adalah jumlah elemen dalam *array*, dan

setiap permutasi memiliki probabilitas kemunculan yang sama yaitu $\frac{1}{n!}$, sehingga hasil pengacakan terbukti tidak bias [9].

*Pseudocode* algoritma Fisher-Yates *Shuffle* adalah sebagai berikut:

**Algorithm 1** Fisher-Yates *Shuffle*

**Require:** *Array A* dengan $n$ elemen
**Ensure:** *Array A* yang telah diacak
1: **for** $i \leftarrow n-1$ **downto** 1 **do**
2: $j \leftarrow \text{Random}(0, i)$ ▷ Pilih indeks acak antara 0 dan $i$
3: $\text{tukar}(A[i], A[j])$ ▷ Tukar posisi elemen ke-$i$ dan ke-$j$
4: **end for**

Algoritma Fisher-Yates *Shuffle* dipilih karena memiliki kompleksitas waktu $O(n)$, bersifat *unbiased* (setiap permutasi memiliki peluang $\frac{1}{n!}$ yang sama), dan efisien untuk diimplementasikan secara *in-place*. Dibandingkan dengan metode pengacakan lain, Fisher-Yates *Shuffle* merupakan pilihan yang paling optimal sebagaimana dirangkum pada Tabel 2.2.

Tabel 2.2. Perbandingan Algoritma Pengacakan

| **Algoritma** | **Kompleksita** | ***Unbiased*** | **Keterangan** |
|---|---|---|---|
| Fisher-Yates | $O(n)$ | Ya | Optimal, digunakan pada penelitian ini |
| Naive Shuffle | $O(n)$ | Tidak | Bias ($n^n \neq n!$) |
| Random Sort | $O(n \log n)$ | Ya | Lebih lambat |
| Sattolo | $O(n)$ | Parsial | Tidak semua permutasi |

Dalam konteks penelitian ini, algoritma Fisher-Yates *Shuffle* diterapkan pada kelas `LevelManager` untuk mengacak pemilihan dan urutan resep yang muncul pada setiap sesi permainan, sehingga pemain tidak dapat memprediksi urutan resep berikutnya.

## 2.6 Android

Android adalah sistem operasi berbasis Linux yang dikembangkan oleh Google untuk perangkat *mobile*. Android menjadi platform target dalam penelitian ini karena tingkat penetrasinya yang tinggi di Indonesia. Unity Engine mendukung *build* ke platform Android melalui Android SDK dan JDK yang terintegrasi dalam

*pipeline* pengembangan Unity, sehingga *game* yang dikembangkan dapat dijalankan pada perangkat Android tanpa perubahan signifikan pada kode program.

## 2.7 Masakan Tradisional Indonesia

Indonesia memiliki kekayaan kuliner yang beragam dan telah diakui secara internasional. Beberapa masakan tradisional Indonesia yang dipilih dalam penelitian ini antara lain:

### 2.7.1 Rendang

Rendang adalah masakan daging bercita rasa pedas yang berasal dari Suku Minangkabau, Sumatera Barat. Masakan ini dimasak menggunakan santan kelapa dan campuran bumbu rempah-rempah seperti cabai, lengkuas, serai, bawang merah, bawang putih, dan daun kunyit. Proses memasak rendang membutuhkan waktu 3–4 jam hingga kuah santan mengering dan bumbu meresap sempurna ke dalam daging [6]. Rendang dinobatkan oleh CNN Travel sebagai makanan terenak di dunia peringkat pertama pada tahun 2011 dan 2017.

### 2.7.2 Sate Ayam

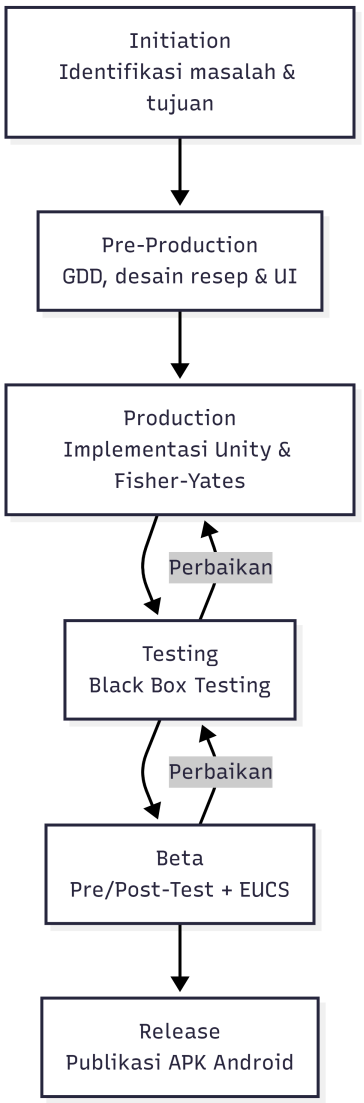
Sate ayam adalah masakan yang terdiri dari potongan daging ayam yang ditusuk dengan tusukan bambu, dibumbui dengan marinasi rempah, kemudian dibakar di atas arang. Sate ayam biasanya disajikan dengan bumbu kacang tanah dan lontong. Sate ayam Madura merupakan variasi sate paling terkenal di Indonesia [6].

### 2.7.3 Nasi Goreng

Nasi goreng adalah masakan nasi yang digoreng dalam wajan dengan kecap manis, bawang putih, dan bumbu lainnya. Nasi goreng merupakan makanan nasional Indonesia yang sangat populer dan menempati peringkat ke-2 dalam daftar *World's 50 Best Foods* versi CNN Travel [6]. Variasi nasi goreng mencakup nasi goreng biasa dan nasi goreng spesial yang dilengkapi telur mata sapi.

## 2.8 *Game Development Life Cycle* (GDLC)

*Game Development Life Cycle* (GDLC) adalah metodologi pengembangan *game* yang terdiri dari serangkaian tahapan sistematis untuk merancang, mengembangkan, dan merilis sebuah *game*. GDLC dipilih karena sesuai dengan karakteristik pengembangan *game* yang bersifat iteratif dan membutuhkan pengujian berkelanjutan di setiap tahapannya [22].




Gambar 2.1. Diagram Prosedur Penelitian (GDLC)

GDLC terdiri dari enam tahapan utama sebagai berikut:

1. ***Initiation***: Tahap awal yang meliputi identifikasi masalah, penentuan

tujuan, dan kajian literatur untuk merumuskan konsep *game* yang akan dikembangkan.

2. ***Pre-Production***: Tahap perancangan *game* secara menyeluruh yang menghasilkan *Game Design Document* (GDD), termasuk perancangan *gameplay*, alur permainan, daftar fitur, dan desain antarmuka.

3. ***Production***: Tahap implementasi atau pembuatan *game* berdasarkan GDD yang telah dirancang, meliputi pemrograman, pembuatan aset, dan integrasi seluruh komponen *game*.

4. ***Testing***: Tahap pengujian internal untuk memastikan semua fitur dan mekanik *game* berjalan sesuai dengan yang dirancang. Metode yang umum digunakan adalah *black box testing*.

5. ***Beta***: Tahap pengujian kepada pengguna nyata (*beta tester*) untuk mengevaluasi kelayakan dan pengalaman bermain. Instrumen yang digunakan dapat berupa kuesioner kepuasan pengguna.

6. ***Release***: Tahap akhir di mana *game* yang telah lulus pengujian didistribusikan kepada pengguna dalam format yang sesuai dengan platform target.

Dibandingkan dengan metode pengembangan perangkat lunak lain seperti *Waterfall* atau *Agile*, GDLC lebih cocok untuk pengembangan *game* karena secara khusus dirancang untuk mengakomodasi sifat iteratif dan kreatif dari proses pembuatan *game*, termasuk aspek desain visual, mekanik permainan, dan pengalaman pengguna [22].

## 2.9 *End-User Computing Satisfaction* (EUCS)

*End-User Computing Satisfaction* (EUCS) adalah model evaluasi yang dikembangkan oleh Doll dan Torkzadeh untuk mengukur tingkat kepuasan pengguna akhir terhadap suatu sistem aplikasi [23]. Model ini telah digunakan secara luas dalam berbagai penelitian untuk mengevaluasi kepuasan pengguna terhadap sistem berbasis komputer, termasuk aplikasi *game* [24].

EUCS terdiri dari lima dimensi utama sebagai berikut [23]:

1. ***Content***: Mengukur seberapa relevan dan berguna konten atau informasi yang disediakan oleh aplikasi bagi pengguna.

2. ***Accuracy***: Mengukur seberapa akurat dan benar informasi atau hasil yang diberikan oleh aplikasi.

3. ***Format***: Mengukur kualitas tampilan dan tata letak antarmuka aplikasi, termasuk desain visual dan estetika.

4. ***Ease of Use***: Mengukur kemudahan penggunaan aplikasi secara keseluruhan, termasuk navigasi dan interaksi pengguna.

5. ***Timeliness***: Mengukur ketepatan waktu sistem dalam memberikan respons dan informasi kepada pengguna.

Instrumen EUCS menggunakan skala Likert untuk mengukur tingkat kepuasan responden. Hasil rata-rata skor kemudian diinterpretasikan berdasarkan interval skala Likert untuk menentukan kategori kepuasan pengguna [24].

## 2.10 *Black Box Testing*

*Black box testing* adalah metode pengujian perangkat lunak yang menguji fungsionalitas sistem tanpa mengetahui struktur kode internal. Pengujian dilakukan berdasarkan spesifikasi kebutuhan (*requirement specification*) dengan memberikan *input* dan mengamati *output* yang dihasilkan. Dalam konteks pengembangan *game*, *black box testing* digunakan untuk memastikan setiap fitur *game* berfungsi sesuai dengan desain yang telah ditetapkan [4].

## 2.11 *Pre-Test*, *Post-Test*, dan *N-Gain*

*Pre-test* dan *post-test* adalah instrumen evaluasi yang digunakan untuk mengukur perubahan pengetahuan atau keterampilan responden sebelum dan sesudah menerima suatu perlakuan (*treatment*). Peningkatan pengetahuan diukur menggunakan rumus *N-Gain* (*Normalized Gain*) yang dikemukakan oleh Hake (1999), dengan formula sebagai berikut:

$$\langle g \rangle = \frac{S_{post} - S_{pre}}{S_{max} - S_{pre}} \tag{2.1}$$

di mana $\langle g \rangle$ adalah nilai *N-Gain*, $S_{post}$ adalah nilai rata-rata *post-test*, $S_{pre}$ adalah nilai rata-rata *pre-test*, dan $S_{max}$ adalah nilai maksimum yang dapat dicapai. Interpretasi nilai *N-Gain*: $\langle g \rangle > 0{,}70$ (Tinggi), $0{,}30 \leq \langle g \rangle \leq 0{,}70$ (Sedang), dan $\langle g \rangle < 0{,}30$ (Rendah). Media edukasi dikatakan efektif apabila $\langle g \rangle \geq 0{,}30$ [25].

## 2.12 Skala *Likert*

Skala *Likert* dikembangkan oleh Rensis Likert pada tahun 1932 untuk mengukur tingkat kesepakatan responden terhadap suatu pernyataan. Skala ini umumnya menyediakan lima hingga sepuluh poin dengan tingkat jawaban yang berbeda, sehingga pendapat atau sikap responden dapat dikuantifikasi [26].

Penelitian ini menggunakan skala *Likert* enam poin (1–6). Penggunaan skala genap tanpa titik tengah netral dipilih agar responden menentukan sikap secara tegas, sehingga terhindar dari kecenderungan memilih jawaban tengah (*central tendency bias*).

Pengumpulan hasil data dilakukan dengan menghitung rata-rata dari total jawaban responden, kemudian diinterpretasikan berdasarkan interval skala yang telah ditentukan [27]. Penentuan interval dimulai dengan mencari nilai *range* (Persamaan 2.2), kemudian membaginya dengan jumlah poin untuk memperoleh lebar interval (Persamaan 2.3), dan menetapkan nilai interval tiap poin dengan menambahkan batas bawah dengan lebar interval (Persamaan 2.4).

$$Range = Skor_{maksimal} - Skor_{minimal} \tag{2.2}$$

$$Lebar\ Interval = \frac{Range}{Jumlah\ Poin} \tag{2.3}$$

$$Nilai\ Interval = Batas\ Bawah + Lebar\ Interval \tag{2.4}$$

Berdasarkan perhitungan tersebut, interval interpretasi skala *Likert* enam poin yang digunakan dalam penelitian ini disajikan pada Tabel 2.3.

Tabel 2.3. Interval Skala *Likert* Enam Poin dan Interpretasinya

| Poin | Interval | Interpretasi |
|---|---|---|
| 1 | 1,00 – 1,83 | Sangat Tidak Puas |
| 2 | 1,84 – 2,66 | Tidak Puas |
| 3 | 2,67 – 3,49 | Agak Tidak Puas |
| 4 | 3,50 – 4,32 | Agak Puas |
| 5 | 4,33 – 5,15 | Puas |
| 6 | 5,16 – 6,00 | Sangat Puas |