

BAB II

LANDASAN TEORI

2.1 Teori-Teori Terkait Bahasan Penelitian (Relevan Theories)

Agar penelitian yang dilakukan dapat terlaksana, maka perlu adanya acuan teori yang berkaitan dengan penelitian yang akan dilakukan. Adapun teori-teori yang berkaitan dengan penelitian ini adalah sebagai berikut:

2.1.1 Game

Game atau permainan adalah bagian mutlak dari kehidupan anak dan permainan merupakan bagian integral dari proses pembentukan kepribadian anak. Game dikenal sebagai sarana hiburan yang baik untuk mengembangkan daya kreatifitas pemainnya.

Game application merupakan salah satu implementasi dari ilmu dibidang komputer. Inti dari sebuah aplikasi permainan adalah mengembangkan kemampuan otak untuk mengatur strategi, kecepatan, dan ketepatan dalam mencapai tujuan akhir (Krisdiawan & Ramdhany Tri, 2018).

2.1.2 Game Edukasi

Menurut Novaliendry (2013), game edukasi adalah permainan yang telah dirancang khusus untuk mengajarkan siswa suatu pembelajaran tertentu, pengembangan konsep dan pemahaman serta membimbing mereka dalam melatih kemampuan mereka.

Sedangkan menurut Widiastuti dan Setiawan (2013), game edukasi adalah game digital yang dirancang untuk pnceayaan pendidikan (mendukung pengajaran dan pembelajaran), menggunakan teknologi multimedia interatif (Yudi Wahyudin et al., 2023). Contoh game edukasi bisa kita lihat pada gambar 2.2 :



Gambar 2.2 Contoh Game Edukasi Mencocokkan Nama Hewan

2.1.3 Huruf Hijaiyah

Huruf hijaiyah merupakan kunci dasar mampu membaca Al-Qur'an dan Hadis. Huruf itu, bagi seorang muslim, menjadi kebutuhan dasar dalam memahami kedua pedoman pokok kehidupannya. Hijaiyah atau juga sering dikenal sebagai huruf Arab itu berjumlah 28 huruf. Huruf itu kemudian merupakan bagian dari bahasa Arab yang menjadi bahasa pokok dalam Al-Qur'an dan Hadis.

Huruf hijaiyah disusun atas dua bentuk yaitu mufrad (tunggal) dan muzdawij (berangkai) yang ditulis dan dibaca dari kanan ke kiri. Bentuk huruf hijaiyah berbeda-beda. Beberapa huruf hijaiyah berbentuk sama yang membedakan adalah titiknya. Huruf hijaiyah bertitik satu, dua, atau tiga. Tempat titik juga bisa berbeda, ada yang di atas, di dalam, dan di bawah. Oleh karena itu yang dimaksud dengan huruf hijaiyah adalah huruf-huruf ejaan bahasa Arab sebagai bahasa asli Al-Qur'an. Dengan kata lain, hijaiyah adalah huruf yang digunakan dalam bahasa Arab untuk membaca Al-Qur'an (Huruf et al., 2017). Huruf hijaiyah digambarkan pada tabel 2.1 :

Tabel 2.1 Huruf Hijaiyah

Arab	Latin
ا	Alif
ب	Ba
ت	Ta
ث	Tsa
ج	Jim
ح	Ha
خ	Kho
د	Dal
ذ	Dzal
ر	Ra
ز	Za
س	Sin
ش	Syin
ص	Shod
ض	Dho
ط	Tho
ظ	Dho
ع	‘Ain
غ	Ghoin
ف	Fa’
ق	Qof
ك	Kaf
ل	Lam
م	Mim
ن	Nun

و	Wau
هـ	Ha
ي	Ya

2.1.4 Android

Android adalah sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka (Aziz, Pribadi, & Nurcahyani, 2020).

2.1.5 Algoritma

Algoritma berasal dari nama seorang pengarang berkebangsaan Arab yang bernama Abu Ja'far Mohammed ibnu Musa al Khowarizmi (790 – 840) yang terkenal dengan sebutan bapak Aljabar. Algoritma adalah langkah detail yang ditunjukkan untuk komputer guna untuk menyelesaikan suatu masalah.

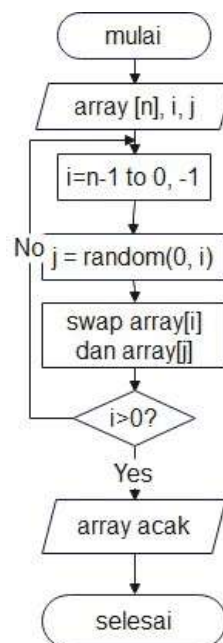
Didalam proses analisis masalah diperlukan tindakan untuk mengidentifikasi informasi yang menjadi keluaran pemecahan masalah untuk mengidentifikasi informasi yang menjadi keluaran pemecahan masalah dan data-data yang menjadi masukan karena itu diperlukan prosedur yang mengolah masukan tersebut menjadi keluaran yang dikehendaki. Karena computer tidak akan berarti apa-apa tanpa adanya langkah detail yang menyusun prosedur (Andriyat Krisdiawan, & Ramdhany Tri, 2018).

2.1.6 Algoritma *Fisher-Yates Shuffle*

Algoritma Fisher-Yates Shuffle bertujuan untuk menghasilkan pengacakan sempurna dengan probabilitas yang seragam untuk setiap elemen dalam daftar. FYS bekerja dengan menukar setiap elemen dalam daftar dengan elemen yang dipilih secara acak dari daftar,

dimulai dari elemen terakhir hingga elemen pertama (Krisdiawan & Sugiharto, 2025).

Fisher-Yates shuffle (diambil dari nama *Ronald Fisher* dan *Frank Yates*) atau juga dikenal dengan nama *Knuth shuffle* (diambil dari nama *Donald Knuth*), adalah sebuah algoritma untuk menghasilkan suatu permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan tersebut. *Fisher-Yates shuffle* jika diimplementasikan dengan benar, maka hasil dari algoritma ini tidak akan berat sebelah, sehingga setiap permutasi memiliki kemungkinan yang sama (Ade-Ibijola, 2012).



Gambar 2.3 Flowchart Algoritma Fisher Yates Shuffle
(Krisdiawan & Sugiharto, 2025)

Berdasarkan gambar 2.3, tahapan untuk menghasilkan permutasi acak dari flowchart adalah sebagai berikut:

1. **Mulai:** Proses dimulai untuk melakukan pengacakan.
2. **Inisialisasi Array, Variabel i dan j:**

- Array [n] adalah kumpulan elemen yang akan diacak.
- Variabel i digunakan sebagai indeks yang dimulai dari elemen terakhir array (indeks n-1) dan akan menurun hingga mencapai 0.
- Variabel j akan digunakan untuk menyimpan indeks acak.

3. Loop dari i = n-1 hingga i = 0:

Proses ini mengacak array dengan mengulang mulai dari indeks terakhir (n-1) hingga indeks pertama (0), mengurangi nilai i setiap iterasi.

4. Pilih j secara Acak dalam Rentang (0, i):

Setiap kali iterasi, j dipilih sebagai angka acak antara 0 dan i. Ini memastikan bahwa setiap elemen dalam array memiliki kemungkinan untuk dipilih pada setiap iterasi.

5. Tukar Elemen pada Indeks i dengan Elemen pada Indeks j:

Nilai pada indeks i dan j diarray ditukar. Proses ini akan mengacak urutan elemen dalam array.

6. Cek Kondisi i > 0:

Jika i masih lebih besar dari 0, proses kembali ke langkah pengacakan berikutnya. Jika i sudah mencapai 0, maka proses pengacakan selesai.

7. Array Acak:

Setelah seluruh elemen array selesai diacak, array kini menjadi urutan acak dari elemen-elemen awalnya.

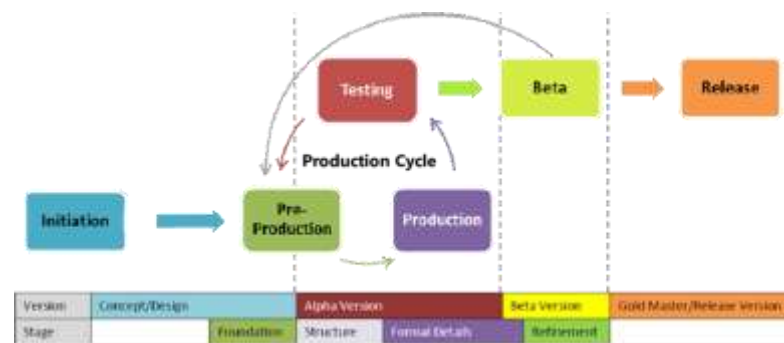
8. Selesai: Proses algoritma Fisher-Yates Shuffle telah selesai, menghasilkan array yang sudah diacak.

2.1.7 Game Development Life Cycle

Game Development Live Cycle (GDLC). *Game Development Live Cycle (GDLC)* adalah suatu metode yang mengelola proses pembuatan game dari awal hingga akhir (Kurniawan et al., 2022). GDLC menerapkan pendekatan iteratif yang terdiri dari 6 fase

pengembangan, dimulai dari fase inisialisasi/pembuatan konsep, preproduction, production, testing, beta dan realease. Dari 6 fase tersebut dapat dikelompokkan menjadi 3 proses utama yaitu :

1. Proses inisialisasi yang terdiri dari konsep dan desain.
2. Proses produksi terdiri dari Pra Produksi, Produksi, dan Pengujian (*Alpha* dan *Beta*).
4. *Release* (Krisdiawan & Darsanto, 2019).



Gambar 2.4 Fase dan Proses GDLC

(Krisdiawan & Darsanto, 2019)

Berdasarkan gambar 2.4, Proses yang dilakukan adalah sebagai berikut :

a. *Initiation.*

Initiation (inisiasi) adalah langkah awal berupa pembuatan konsep kasar dari game (Syarif et al., 2022).

Pada tahap ini peneliti membuat konsep awal permainan yang fokus pada pengenalan huruf hijaiyah beserta huruf latinnya dan jenis game yang akan dibuat sesuai dengan target pengguna.

b. *Pre-production.*

Pra-produksi adalah salah satu fase yang penting dalam siklus produksi. Pra-produksi melibatkan penciptaan dan revisi desain game dan pembuatan prototipe permainan. Desain game berfokus pada mendefinisikan genre permainan, gameplay, game mekanik/konvensional, alur cerita, karakter, tantangan, faktor kesenangan, aspek teknis, dan dokumentasi elemennya

dalam Dokumen Desain Game (GDD) (Krisdiawan & Darsanto, 2019). Pada tahap ini peneliti menentukan gameplay, karakter, alur sistem, storyboard, antarmuka, dan mengidentifikasi huruf hijaiyah yang akan dimasukkan ke dalam permainan serta mempersiapkan aset yang akan digunakan.

c. *Production.*

Produksi merupakan inti dari proses yang difokuskan pada pembuatan aset, pengembangan source code, dan integrasi keduanya. Prototipe yang terkait dengan tahap ini melibatkan perincian dan penyempurnaan mekanika serta aset yang lebih lengkap. Kegiatan produksi terkait mencakup penciptaan dan penyempurnaan detail formal, seperti penyeimbangan permainan yang terkait dengan kriteria kualitas seimbang, penambahan fitur baru, peningkatan kinerja secara keseluruhan, dan perbaikan bug terkait dengan kriteria kualitas fungsional dan internal yang lengkap. Proses penyeimbangan permainan juga mencakup penyesuaian kesulitan permainan untuk mencapai tingkat kesulitan yang sesuai (Leveling) (Syarif et al., 2022). Pada tahap ini peneliti mulai melakukan penyempurnaan perancangan, membuat aset, menerapkan kode program yang mana didalamnya terdapat pengimplementasian algoritma *fisher-yates shuffle*, melakukan integrasi antara aset dan source code dengan tools Unity.

d. *Testing.*

Pengujian dalam konteks ini berarti pengujian internal dan eksternal dilakukan untuk menguji kegunaan permainan. Metode pengujian khusus untuk setiap tahap prototype (Krisdiawan & Darsanto, 2019). Dalam konteks ini, merujuk pada pengujian internal untuk menguji kinerja dan jalannya permainan. Metode pengujian disesuaikan untuk setiap tahap prototipe. Pengujian Rincian Formal dilakukan menggunakan playtest

untuk mengevaluasi fungsionalitas fitur dan tingkat kesulitan permainan (terkait dengan keseimbangan) (Syarif et al., 2022). Pada tahap testing ini peneliti menggunakan Balckbox dan Whitebox testing sebagai pengujian pada game edukasi yang dibangun.

e. *Beta.*

Pada fase Beta, permainan diuji oleh pihak ketiga atau eksternal melalui pengujian beta. Terdapat dua jenis pengujian beta: tertutup (hanya diundang) dan terbuka (siapa pun dapat mendaftar). Hasilnya melibatkan laporan bug dan masukan pengguna. Sesi Beta ditutup karena berakhirnya periode atau jumlah tester yang mencapai batas tertentu (Syarif et al., 2022). Pada tahap ini peneliti melakukan tester tertutup, dimana yang menjadi tester dari game edukasi yang dibangun yaitu Guru TPA Al-Widayah dan murid TPA Al-Widayah. pada tahap ini juga peneliti melakukan UAT (*User Acceptence Test*) berupa kuesioner.

f. *Release.*

Fase ini merupakan fase dimana pengembangan game telah mencapai tahap puncaknya. Proses rilis melibatkan peluncuran produk, penyusunan dokumentasi proyek dan berbagi informasi (Syarif et al., 2022). Pada tahap ini game edukasi sudah siap dirilis dan game edukasi siap digunakan oleh para murid TPA Al-Widayah Pondok Ranggon.

2.1.8 Tools Perancangan

2.1.8.1 *Game Design Document (GDD)*

Game Design Document (GDD) adalah dokumen yang berisi informasi secara detail dan terperinci mengenai pengembangan sebuah game, gamification atau game-based learning. Dokumen tersebut bertujuan untuk memberikan kemudahan antar bagian

dalam perancangan game, mulai dari *Game Artist*, *Sound I*, *Game Designer*, *Software Engineer* dan *Game Tester* (Arifudin, Pramesti, & Heryanti, 2022). Berikut ini merupakan isi dari *Game Design Document* :

1. *Storyboard*

Storyboard adalah area berseri dari sebuah gambar sketsa yang digunakan sebagai alat perencanaan untuk menunjukkan secara visual bagaimana aksi dari sebuah cerita berlangsung. *Storyboard* merupakan naskah yang dituangkan dalam bentuk gambar atau sketsa yang berguna untuk lebih memudahkan cameraman dalam pengambilan gambar (Nurhasanah & Destyany, 2011).

Storyboard secara harfiah berarti dasar cerita, storyboard adalah penjelasan bagaimana cara seseorang akan membuat suatu proyek. Jika diumpamakan sebagai pembuatan film, maka bisa dibilang bahwa storyboard adalah skenario film tersebut (Nurhasanah & Destyany, 2011). Contoh storyboard dapat kita lihat pada gambar 2.1:

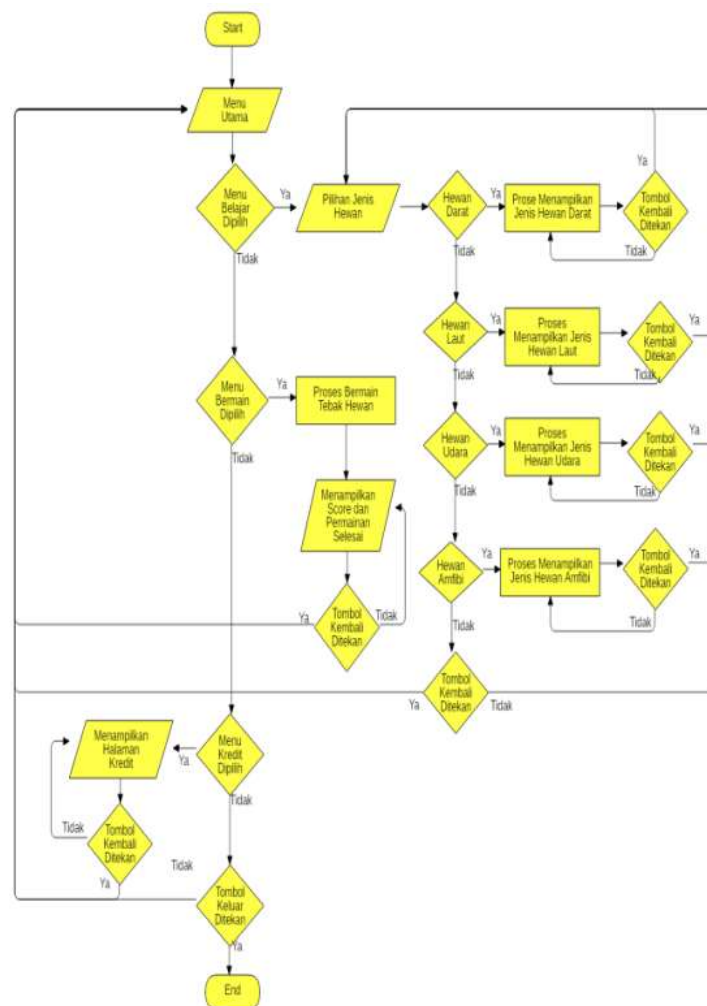
Frame	Gambar	Keterangan
1		Tampilan <i>Matching Game</i> . Kotak-kotak yang berisi gambar masih tertutup.

Gambar 2.5 Contoh Storyboard Pada Game

(Nurhasanah & Destyany, 2011)

2. Game Layout Chart

Game layout chart merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program. Biasanya untuk mempermudah penyelesaian masalah yang khususnya perlu dipelajari dan dievaluasi lebih lanjut (Kurnia Arrifqie, 2022).



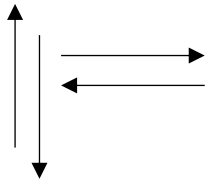
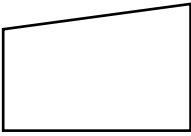
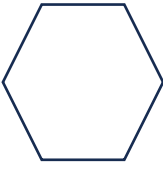
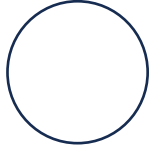
Gambar 2.6 Contoh Game Layout Chart

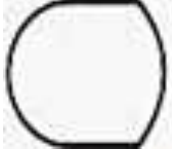
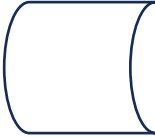
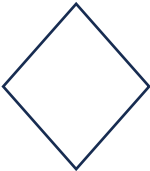
(Kurnia Arrifqie, 2022)

2.1.8.2 Flowchart

Flowchart atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah (Septiansyah, Hasanah, Nita Permatasari, & Yuliawati). Seorang analis sistem menggunakan flowchart sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer (Rosaly, Prasetyo, & Kom, 2019).

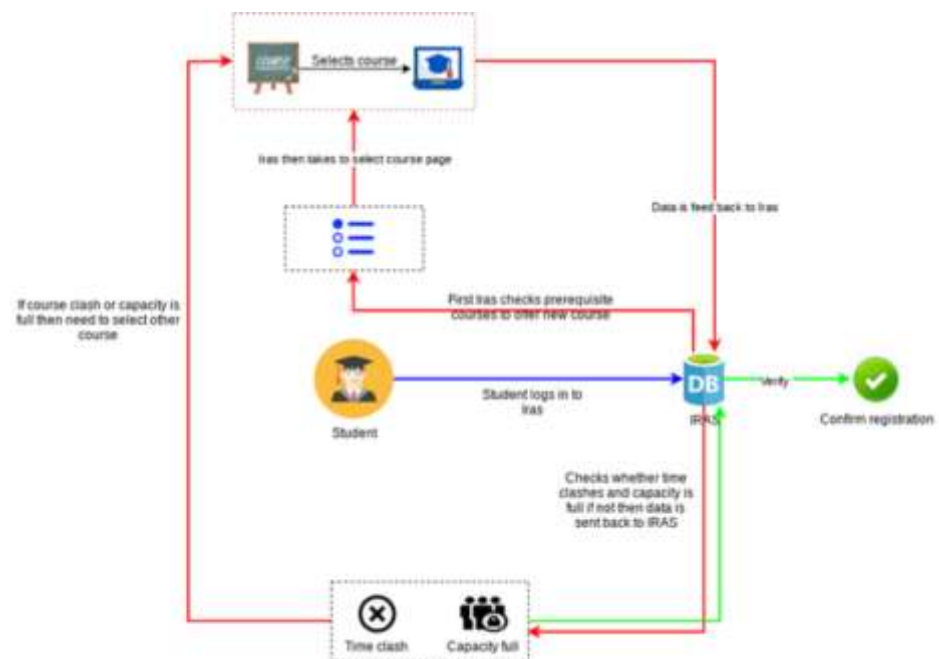
Tabel 2.2 Simbol Flowchart

	Flow Direction Symbol Simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan didalam storage
	Connector Symbol Yaitu simbol untuk keluar masuk atau penyambungan proses dalam lembar/halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure

	Connector Symbol Yaitu simbol untuk keluar masuk atau penyambungan proses pada lembar/halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses Input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas

2.1.8.3 Rich Picture

Rich Picture adalah gambaran informal yang menggambarkan situasi yang dipahami atau dimengerti oleh pelukis. Sedangkan menurut Andreas Riel (2010, p17) Rich picture adalah sebuah metodologi untuk merepresentasikan ide, permasalahan atau suatu konsep. Rich picture menyediakan gambaran umum dari sebuah topik dan juga menunjukkan hubungan dan saling ketergantungan diantara elemen – elemennya, mengidentifikasi aktivitas utama dan aktor dalam aktivitas utama tersebut dengan jelas. Dari pengertian diatas dapat kami simpulkan bahwa Rich Picture adalah suatu gambaran mengenai situasi yang dipahami atau dimengerti dalam penggambaran sistem secara menyeluruh (Suharni, Susilowati, & Pakusadewa, 2023).



Gambar 2.7 Contoh Rich Picture

(Suharni et al., 2023)

2.1.8.4 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah sebuah Bahasa pemodelan perangkat lunak yang telah distandardisasi sebagai

media penulisan cetak biru (*blueprints*) perangkat lunak (*Pressman*). UML bisa saja digunakan untuk visualisasi, spesifikasi, konstruksi dan dokumentasi beberapa bagian-bagian dari system yang ada dalam perangkat lunak (Sumiati, Abdillah, & Cahyo, 2021).

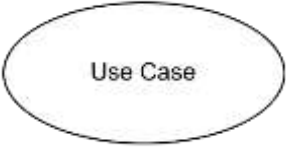
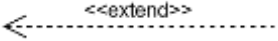
Menurut Munawar, *Unified Modeling Language* (UML) adalah salah satu alat bantu yang sangat handal di dunia pengembangan system berorientasi obyek. Hal ini disebabkan karena UML menyediakan Bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2018).

1. Use Case Diagram

Use case diagram merupakan gambaran dari fungsionalitas yang diharapkan dari sebuah sistem, dan mempresentasikan sebuah interaksi antara aktor dan sistem. Di Dalam Use Case terdapat aktor yang merupakan sebuah gambaran entitas dari manusia atau sebuah sistem yang melakukan pekerjaan di sistem (Ramdany et al.). Tabel 2.3, merupakan tabel simbol Use Case Diagram yang menggambarkan simbol dan kegunaan dari Use Case Diagram :

Tabel 2.3 Simbol Use Case Diagram

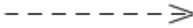
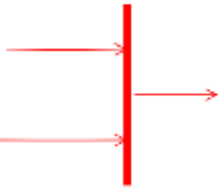
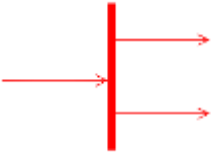
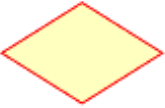
Gambar	Nama	Fungsi/Deskripsi
	Actor	Digunakan untuk menjelaskan sesuatu atau seseorang yang sedang berinteraksi dengan sistem.

	Use Case	Menggambarkan suatu perilaku dari sistem tanpa mengungkapkan struktur internal dari sistem tersebut.
	Association	Menggambarkan relasi antara actor dengan use case.
	Extend	Relasi use case tambahan ke sebuah use case dan use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan itu.
	Include	Use case yang ditambahkan akan selalu dipanggil saat use case tambahan dijalankan.
	Generalization	Menggambarkan relasi lanjut antar use case atau menggambarkan struktur pewarisan antar actor.

2. Activity Diagram

Diagram aktivitas merupakan pemodelan dari keseluruhan proses yang terjadi pada system. Tabel 2.4, merupakan simbol Activity Diagram yang menggambarkan simbol dan kegunaan dari Activity Diagram (Ramdany et al.).

Tabel 2.4 *Simbol Activity Diagram*

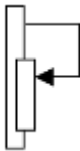
Gambar	Nama	Fungsi/Deskripsi
	Activity	Menyatakan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	Control Flow	Menunjukkan urutan eksekusi.
	Object Flow	Menunjukkan aliran objek dari sebuah action atau activity ke action.
	Start Point	Menyatakan bahwa sebuah objek dibentuk atau diawali.
	End Point	Menyatakan bahwa sebuah objek diakhiri.
	Join/penggabungan	Menyatakan untuk menggabungkan kembali activity atau action yang parallel.
	Fork	Menyatakan untuk memecah kembali activity atau action yang parallel.
	Decisions	Menunjukkan penggambaran suatu keputusan/tindakan

		yang harus diambil pada kondisi tertentu.
--	--	---

3. Sequence Diagram

Diagram sequence menjelaskan interaksi pada objek berdasarkan urutan waktu. Diagram sequence berfokus pada perilaku dalam system, mengilustrasikan bagaimana objek dan pesan dikirim antar proyek. Tabel 2.5, merupakan simbol Sequence Diagram yang menggambarkan simbol dan kegunaan dari Sequence Diagram (Septiansyah et al.).

Tabel 2.5 Simbol Squence Diagram

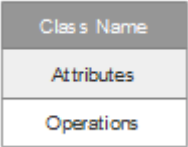
Gambar	Nama	Fungsi/Deskripsi
	Entity Class	Gambaran sistem sebagai landasan dalam menyusun basis data.
	Boundary Class	Menangani komunikasi antar lingkungan sistem.
	Control Class	Bertanggung jawab terhadap kelas-kelas terhadap objek yang berisi logika.
	Recursive	Pesan untuk dirinya.

	Activation	Mewakili proses durasi aktivasi sebuah operasi.
	Life Line	Komponen yang digambarkan garis putus terhubung dengan objek.

4. Class Diagram

Diagram Kelas adalah representasi visual yang digunakan untuk menampilkan kelas-kelas atau paket-paket pada sistem yang akan digunakan. Tabel 2.6, merupakan table simbol Class Diagram yang menggambarkan simbol dan kegunaan dari Class Diagram (Suharni, Susilowati, & Pakusadewa, 2023).

Tabel 2.6 Simbol Class Diagram

Gambar	Nama	Fungsi/Deskripsi
	Class	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
	Association	Relasi antar kelas dengan makna umum dan biasanya disertai multiplicity.
	Directed Association	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain.
	Aggregation	Mengindikasikan keseluruhan bagian relationship disebut sebagai relasi.

	Composition	Menggambarkan relasi generalisasi
	Dependency	Menunjukkan operasi pada suatu class yang menggunakan class yang lain.

2.1.9 Bahasa Pemrograman

Pada penelitian ini Bahasa pemrograman yang digunakan adalah :

2.1.9.1 *Personal Home Page* (PHP)

PHP adalah bahasa yang dirancang secara khusus untuk penggunaan pada Web (Lutfi, 2017). PHP adalah tool untuk pembuatan halaman web dinamis. PHP merupakan script yang terintegrasi dengan HTML dan berada pada server (server side HTML embedded scripting). PHP adalah script yang digunakan untuk membuat halaman website yang dinamis (Sitanggang, Dachi, & Manurung, 2022).

2.1.9.2 C-sharp (C#)

C#, dikatakan sebagai C Sharp, merupakan bahasa pemrograman berorientasi objek yang dikembangkan oleh Microsoft pada awal 2000-an, dipimpin oleh Anders Hejlsberg. Sebuah bagian dari kerangka .Net dan dimaksudkan untuk menjadi bahasa pemrograman tujuan umum sederhana yang dapat digunakan untuk mengembangkan berbagai jenis aplikasi, termasuk aplikasi konsol, windows, web, dan seluler.

Seperti semua bahasa pemrograman modern, kode C# menyerupai bahasa Inggris yang tidak dapat dipahami oleh komputer. Oleh karena itu, kode C# harus diubah menjadi bahasa mesin menggunakan apa yang dikenal sebagai *compiler*.

Jadi, C# adalah bahasa pemrograman berorientasi objek (OOP). Pemrograman berorientasi objek adalah pendekatan pemrograman yang memecah masalah pemrograman menjadi objek yang berinteraksi satu sama lain (Raharjo, 2022).

2.1.10 Tools Perangkat Lunak

2.1.10.1 Unity 2D

Unity adalah game engine buatan Unity Technologies Inc. Unity adalah sebuah tool yang terintegrasi untuk membuat game, arsitektur bangunan dan simulasi. Unity bisa untuk games PC dan games Online. Untuk games Online diperlukan sebuah plugin, yaitu Unity Web Player, sama halnya dengan Flash Player pada Browser. Unity tidak dirancang untuk proses desain atau modelling, dikarenakan Unity bukan tool untuk mendesain. Fitur scripting yang disediakan, mendukung tiga bahasa pemrograman yaitu; JavaScript, C#, dan Boo (Stefano Mongi, M Lumenta, & Sambul, 2018).

2.1.10.2 Visual Studio Code

Visual Studio Code adalah salah satu Integrated Development Environment (IDE) yang sangat efektif dan populer di kalangan pengembang. IDE ini memiliki fitur-fitur yang sangat lengkap dan memudahkan pengembang dalam membuat aplikasi mobile. Dengan mempelajari Visual Studio Code, pemula dapat menggunakan IDE ini untuk mengembangkan aplikasi mobile dengan lebih mudah dan cepat (Gunawan, 2024).

2.1.10.3 MySQL

MySQL merupakan database engine atau server database yang mendukung bahasa database SQL sebagai bahasa interaktif dalam mengelola data. MySQL adalah sebuah perangkat lunak

sistem manajemen basis data SQL atau DBMS yang multithread, multi-user (Sitanggang et al., 2022).

2.1.10.4 Rational Rose

Rational Rose adalah software yang memiliki perangkat pemodelan secara visual untuk membangun suatu solusi rekayasa piranti lunak dan pemodelan bisnis. Rational Rose memakai UML sebagai bahasa pemodelannya ditambah beberapa fitur lain yang membuatnya menjadi software pemodelan visual yang terkemuka (Pujadi, 2011).

Beberapa fitur diantaranya adalah Rational Rose memiliki *Rational Unified Process* (RUP) dan juga memiliki kemampuan membuat solusi client/server yang dapat diterapkan dan didistribusikan dalam lingkungan sebuah Perusahaan (Pujadi, 2011).

2.1.10.5 Visual Paradigm

Visual Paradigm adalah sebuah software model dengan system visualisasi memungkinkan model yang telah dibuat dapat digunakan sebagai representasi proyek-proyek lain dilengkapi dengan beberapa fitur yang ada didalamnya sampai pada menganalisa sebuah proyek yang akan dikerjakan (Bastian Malik & Supriatna, 2019).

Diagram dapat disusun sedemikian rupa sehingga dapat dipustakakan menjadi proyek per proyek yang saling berkaitan. Hal ini dapat juga membantu memisahkan terhadap pekerjaan proyek sampai level terkecil (Bastian Malik & Supriatna, 2019).

2.1.10.6 Inkscape

Inkscape merupakan software atau perangkat lunak editor gambar vektor yang bersifat bebas terbuka dibawah lisensi GNU GPL. Tujuan utama dari Inkscape adalah untuk membuat perangkat grafik mutakhir yang memenuhi standar XML, SVG, dan CSS (Fitriani et al.).

Inkscape juga tersedia untuk sistem operasi Windows, Macintosh, dan Linux. Program dan kode sumber (source code) Software Inkscape dapat digunakan untuk membuat suatu gambar vektor untuk berbagai kebutuhan, seperti membuat gambar ilustrasi pada web, ikon untuk smartphone, gambar kartun atau animasi, membuat garis sederhana, kaligrafi, logo, brosur, dan masih banyak lagi (Fitriani et al.).

2.1.10.7 Figma

Figma adalah salah satu design tool yang biasanya digunakan untuk membuat tampilan aplikasi mobile, desktop, website dan lain-lain. Figma bisa digunakan di sistem operasi windows, linux ataupun mac dengan terhubung ke internet. Umumnya Figma banyak digunakan oleh seseorang yang bekerja dibidang UI/UX, web design dan bidang lainnya yang sejenis. Selain mempunyai kelengkapan fitur layaknya Adobe XD (Agus Muhyidin, Sulhan, & Sevtiana, 2020).

Figma memiliki keunggulan yaitu untuk pekerjaan yang sama dapat dikerjakan oleh lebih dari satu orang secara bersama-sama walaupun ditempat yang berbeda. Hal tersebut bisa dikatakan kerja kelompok dan karena kemampuan aplikasi figma tersebutlah yang membuat aplikasi ini menjadi pilihan banyak UI/UX designer untuk membuat prototype website atau aplikasi dengan waktu yang cepat dan efektif (Agus Muhyidin et al., 2020).

2.1.10.8 XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket.

XAMPP merupakan salah satu paket instalasi Apache, PHP, dan MySQL instant yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut. Dengan menginstal XAMPP maka

tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual (Sitanggang et al., 2022).

2.1.10.9 IBM SPSS

Platform software IBM SPSS merupakan salah satu aplikasi yang dapat digunakan sebagai tools data mining yang memiliki fitur-fitur advanced analisis statistik, machine learning algorithms, dan text analysis, ekstensibilitas sumber terbuka, dan dapat terintegrasi dengan beberapa sumber data. Aplikasi SPSS menawarkan pengoperasian yang mudah bagi pengguna, fleksibilitas, dan skalabilitas SPSS memungkinkan diakses oleh pengguna dari semua tingkat keahlian pengoperasian sistem komputerisasi yang dimiliki pengguna (Ramadhani et al., 2023).

2.1.11 Pengujian Perangkat Lunak

2.1.11.1 Blackbox

BlackBox testing adalah metode untuk menguji fungsionalitas sistem aplikasi. Ini menunjukkan kesalahan pada fungsi dan menu aplikasi yang hilang. Untuk mendapatkan hasil yang akurat, pengujian menggunakan data acak. Jika data input benar, sistem informasi dapat menerima atau menyimpannya dalam database, tetapi jika data input salah, sistem informasi akan menolaknya atau tidak dapat menyimpannya dalam database (Nur Ichsanudin et al., 2022).

2.1.11.2 Whitebox

Pengujian whitebox merupakan pengujian yang dikembangkan berdasarkan kode program. Seorang pengujian pengujian whitebox harus memiliki pengetahuan tentang kode dan pengetahuan menulis kasus uji dengan parameter yang sesuai. Hal ini terutama mempengaruhi aliran kontrol program dan aliran data (Nugraha, 2022).

2.1.11.3 Pengujian UAT

User Acceptance Testing (UAT) adalah pengujian interaksi antara end-user dan sistem secara langsung yang berfungsi untuk memverifikasi bahwa fitur telah berjalan sesuai dengan kebutuhan user tersebut. Pengujian UAT termasuk fase terakhir dalam proses pengujian pada sistem, yang dimana sistem telah selesai melalui tahap pengembangan. UAT menjadi salah satu rangkaian pengujian final dari perangkat lunak dan dilakukan sebelum dikembangkan dan diluncurkan.

Pengujian ini bertujuan untuk mengetahui kesesuaian sistem terhadap kebutuhan user. Pengujian ini menggunakan UAT jenis *Contract Acceptance Testing*. Selain itu, pengujian ini juga berfokus untuk mengukur kenyamanan user dalam menggunakan sistem serta dapat menyelesaikan permasalahan yang dihadapi (Chamida, Susanto, & Latubessy, 2021).

2.2 Penelitian Sebelumnya (*Previous Work*)

Penelitian ini dilakukan tidak terlepas dari hasil penelitian-penelitian terdahulu yang pernah dilakukan sebagai bahan perbandingan dan kajian. Sejauh ini, penelitian mengenai penerapan algoritma Fisher-Yates Shuffle (FYS) dan pengacakan data sudah banyak dilakukan. Hal ini berguna untuk menambah referensi dalam memperkaya bahan kajian. Beberapa penelitian yang relevan dapat dilihat pada Tabel 2.7.

Tabel 2.7 *Tabel Perbandingan Penelitian Sebelumnya*

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
1.	Fransiskus Panca Juniawan et al., "Performance Comparison of LCM and Fisher-Yates Shuffle", 2019	Perbandingan kinerja algoritma shuffle	Menganalisis kinerja algoritma Fisher-Yates dan LCM dalam pengacakan data	Uji Chi-Square, Runs Test, analisis kompleksitas waktu dan memori	Dataset dengan variasi 10, 100, 1000, 10000, dan 25000 data.	Fisher-Yates lebih cepat 11.768% dibandingkan LCM, dengan efisiensi yang meningkat seiring bertambahnya data.
2.	Rio Andriyat Krisdiawan & Tito	Membandingkan kinerja tiga	Mengevaluasi dan	Uji Kualitas Pengacakan (Chi-	Dataset 1000 kata bahasa Inggris,	Algoritma LCM memiliki waktu

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
	Sugiharto, "Comparison of Shuffle Algorithms", 2025	algoritma shuffle	membandingkan kinerja algoritma shuffle dalam kualitas pengacakan, efisiensi waktu, dan penggunaan memori	Square Test, Runs Test), Uji Kompleksitas Waktu, Uji Kompleksitas M	ukuran dataset 100, 500, 1000 elemen	eksekusi paling cepat dan penggunaan memori stabil. Fisher-Yates Shuffle efisien, sedangkan Naive Shuffle kurang efisien.
3.	Yati Nurhayati, "Implementasi Algoritma Fisher Yates Shuffle pada Game Pengenalan Buah", 2019	Kurangnya pengetahuan tentang buah-buahan Indonesia	Meningkatkan pengetahuan masyarakat tentang buah-buahan Indonesia	Game berbasis Android menggunakan algoritma Fisher-Yates Shuffle	Tidak disebutkan jumlah data/sample	Game berhasil menarik minat dan meningkatkan pengetahuan tentang buah-buahan Indonesia

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
			melalui game			di kalangan anak-anak.
4.	Sri Ipnuwati, "Aplikasi Game Edukasi Pengenalan Huruf Hijaiah Untuk Anak Usia Dini", 2020	Metode konvensional dalam pengenalan huruf hijaiyah	Menciptakan aplikasi game edukasi untuk mengenalkan huruf hijaiyah kepada anak-anak	Metode Prototype, pengujian menggunakan Black Box Testing	Pengujian di TPQ Al-Muhajirin, Lampung	Aplikasi mampu menarik minat belajar anak-anak dan meningkatkan pemahaman huruf hijaiyah.
5.	Uswatun Hasanah, "Implementasi Game Edukasi Berbasis Android Sebagai Media Pembelajaran", 2023	Rendahnya minat belajar anak dalam mengenal huruf hijaiyah	Mengembangkan game edukasi untuk meningkatkan pemahaman huruf hijaiyah	Game Development Life Cycle (GDLC), pengujian menggunakan ISO 9126	18 siswa di R.A Al-Basyari, Lampung Tengah	Skor pengujian aspek usability mencapai 96,8%, menunjukkan game efektif sebagai media pembelajaran

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
6.	Darius Andana Haris, "Hijaiyah Letters Educational Game 'EFRA'", 2023	Keterbatasan metode pengajaran huruf hijaiyah di masa pandemi	Memberikan alternatif media pembelajaran huruf hijaiyah untuk anak- anak	Pengembangan menggunakan Unity, pengujian Black Box, alpha dan beta testing	Pengujian di Darul Hawasyi School	Game EFRA berhasil menarik minat belajar dan memberikan pengalaman interaktif dalam pengenalan huruf hijaiyah.
7.	Diena Rauda Ramdania, "Fisher- Yates and Fuzzy Sugeno in Game for Children with Special Needs", 2020	Rendahnya pemahaman bahasa Inggris pada anak dengan kebutuhan khusus	Menciptakan game edukasi bahasa Inggris menggunakan algoritma Fisher-Yates dan Fuzzy Sugeno	Quasi-experimental method, pre-test dan post-test	3 siswa dengan kebutuhan khusus di Bandung	Rata-rata skor meningkat dari 80,41 menjadi 88,3 setelah bermain game.

No	Peneliti (Judul, Penulis, Tahun Penelitian)	Masalah Penelitian	Tujuan Penelitian	Metodelogi Penelitian/Pemecahan masalah	Jumlah data/sample data/penelitian	Hasil Penelitian
8.	Imroatun, "Pembelajaran Huruf Hijaiyah bagi Anak Usia Dini", 2017	Kesulitan anak dalam mengenali huruf hijaiyah	Meningkatkan pemahaman anak usia dini terhadap huruf hijaiyah	Pembelajaran berbasis lingkungan, dengan pendekatan yang sesuai untuk anak usia dini	Siswa RA dan yang sederajat usia 4-6 tahun	Pembelajaran huruf hijaiyah efektif jika dilakukan secara terencana dan berkesinambungan

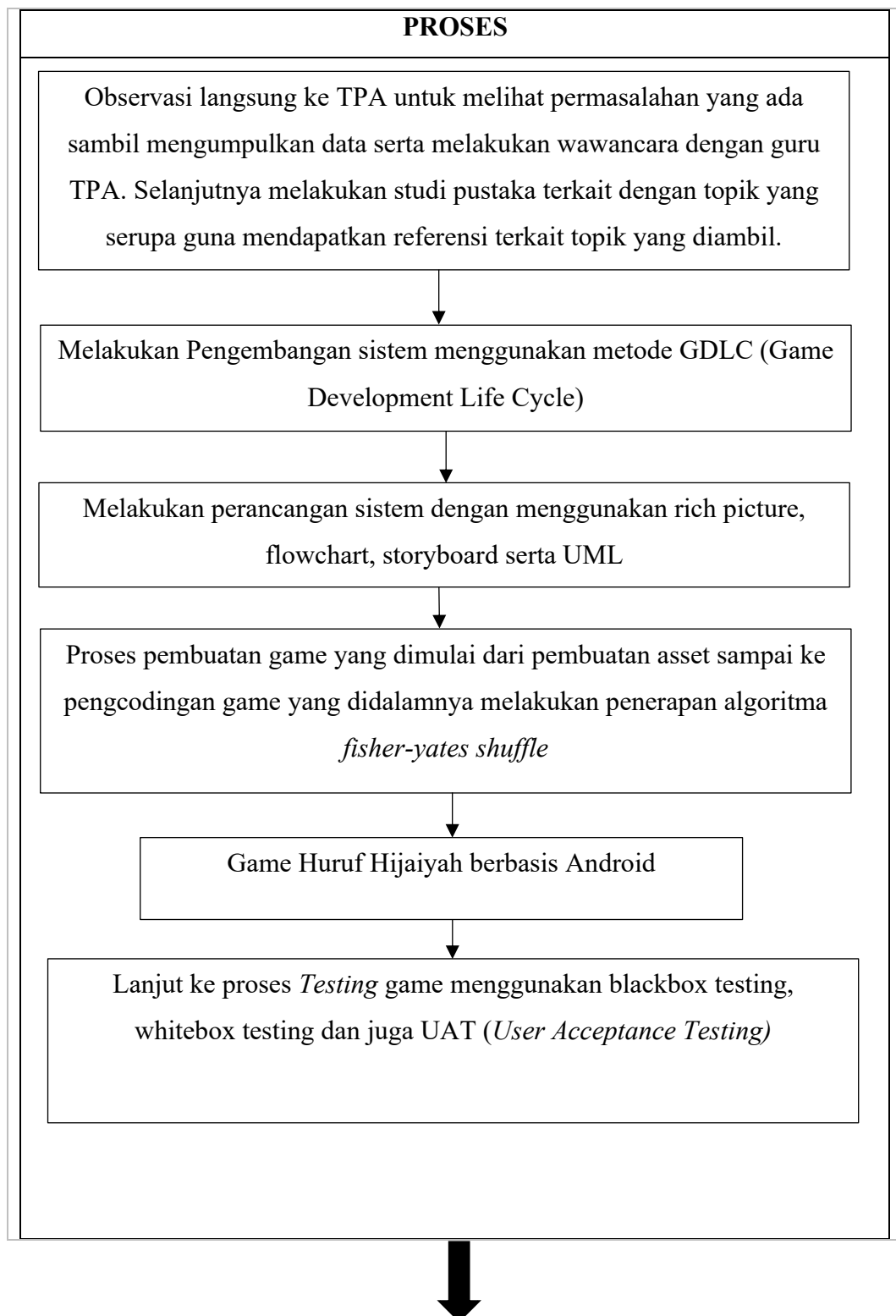
2.3 Kerangka Teoritis (*Theoretical Framework*)

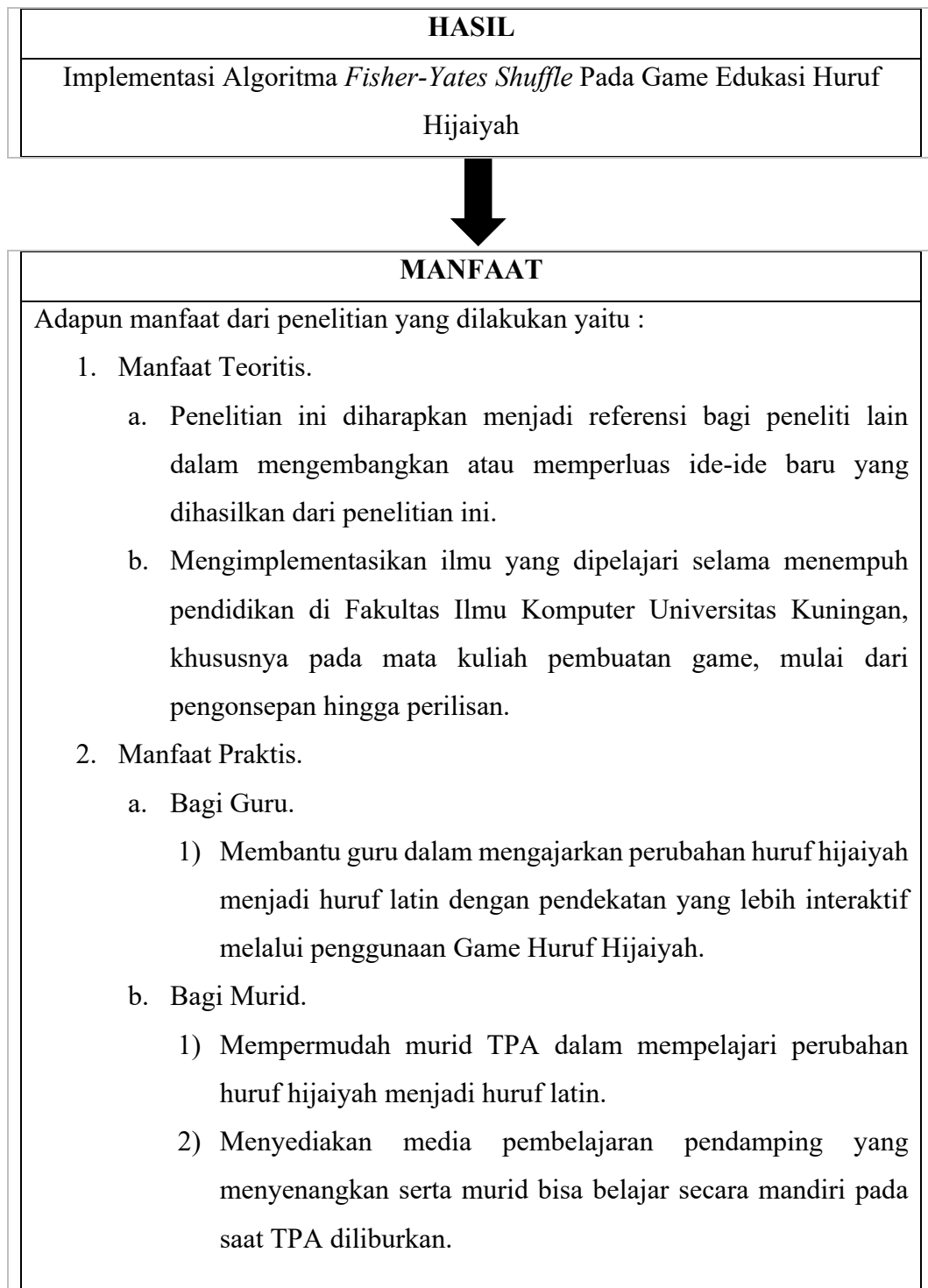
MASALAH
1. Banyak murid di TPA Al-Widayah yang mengalami kesulitan dalam mengubah huruf Hijaiyah menjadi huruf Latin dan sebaliknya, dengan 80% murid tidak mencapai KKM pada pretest. 2. Murid cenderung mengandalkan hafalan pola jawaban tanpa pemahaman mendalam, yang terlihat dari kesalahan berulang pada soal yang sama.



TUJUAN
1. Membangun Game Huruf Hijaiyah yang memfasilitasi media pembelajaran pendamping perubahan huruf hijaiyah menjadi huruf latin di TPA Al-Widayah Pondok Ranggon. 2. Mengimplementasikan algoritma Fisher-Yates Shuffle untuk mengacak soal huruf hijaiyah dan susunan huruf latin pada jawaban soal dalam permainan sehingga murid tidak menghadapi pola soal yang sama setiap kali bermain.







Gambar 2.8 Kerangka Teoritis