



IMPLEMENTASI METODE *GDLC* PADA GAME *TAXI RUSH* MENGUNAKAN *UNITY ENGINE*

Matthew George Wellson¹⁾, Wahyu Tisno Atmojo²⁾

^{1,2}*Sistem Informasi, Universitas Pradita*

^{1,2}*Scientia Business Park, Jl. Gading Serpong Boulevard No.1, Curug Sangereng, Kec. Klp. Dua, Kabupaten Tangerang, Banten 15810.*

Email: ¹matthew.george@student.pradita.ac.id, ²wahyu.tisno@pradita.ac.id

Abstract

Games are a means of entertainment to please yourself. In ancient times, games existed all over the world, including Indonesia. Some traditional games that we often hear about are hide and seek, Gobak Sodor, rubber rope jumping, kites, and others. However, as time goes by, technological devices are starting to emerge from computers, game consoles, to the smartphone era. The emergence of this technological device allows humans to create digital games, creating an era of digital games whose playing we call video games. Currently, video games have become rampant and are played by people of all ages using smartphones, PCs (Personal Computers), game consoles, etc. To create a digital game, you need a game engine designed to develop video games on various platforms. One of these game engines is the Unity Engine. Unity Engine is a cross-platform game engine developed by Unity Technologies. This game engine supports C#, a programming language created and developed by Microsoft Corporation in 2000. Unity Engine can be used to create 2D (two-dimensional) and 3D (three-dimensional) games and supports various cross-platform devices such as smartphones, PCs, PlayStation 5, and many other platforms. This game engine is considered easy to use for novice developers, and has been adopted by many in the gaming industry. In this research, researcher want to implement the creation of a 2D game called Taxi Rush using the GDLC (Game Development Life Cycle) method. Taxi Rush is a game created by several other people and myself on the Unity Engine. Taxi Rush has a racing genre and a time-attack sub-genre. The game publication platform used by researchers is itch.io with WebGL game platform. This research was created to implement the GDLC method in making games on the WebGL platform. The reason why this research was carried out was because I saw that the game industry and development in Indonesia was still small. With so few game developers in Indonesia, learning resources to develop good and correct games in Indonesian are very limited. Therefore, I decided to contribute the game development knowledge that I had studied over several years (using Indonesian so that it could be read and accessed by all Indonesian people) in the form of this journal to Indonesia.

Keyword: *Game, 2D Game, Game Development, Unity Engine, GDLC.*

Abstrak

*Game (Permainan) adalah sebuah sarana hiburan untuk menyenangkan diri sendiri. Pada zaman dahulu, game sudah ada di seluruh dunia termasuk Indonesia. Beberapa game tradisional yang sering kita dengar seperti Petak Umpet, Gobak Sodor, Lompat Tali Karet, layangan, dan lain-lain. Namun seiring berkembangnya zaman, perangkat teknologi mulai bermunculan dari Komputer, game konsol, hingga era *smartphone*. Kemunculan perangkat teknologi ini memungkinkan manusia membuat game digital hingga membuat era game digital yang pemainannya kita sebut dengan *video game*. Saat ini *video game* telah merajalela dan dimainkan oleh manusia dengan berbagai usia baik menggunakan *smartphone*, *PC* (Personal Computer), *game konsol*, dan lain-lain. Untuk membuat sebuah game digital, diperlukan sebuah game engine dirancang untuk mengembangkan video game di berbagai platform. Salah satu dari *game engine* ini adalah *Unity Engine*. *Unity Engine* adalah *cross-platform game engine* yang dikembangkan oleh Unity Technologies. *Game engine* ini mendukung C#, bahasa pemrograman yang dibuat dan dikembangkan oleh *Microsoft Corporation* pada tahun 2000. *Unity Engine* dapat digunakan untuk membuat *game 2D* (two-dimensional) dan *3D* (three-dimensional) dan mendukung berbagai perangkat lintas platform seperti *smartphone*, *PC*, *PlayStation 5*, dan banyak platform lainnya. *Game engine* ini dianggap mudah digunakan oleh pengembang pemula, dan telah diadopsi oleh banyak industri game. Dalam penelitian ini, peneliti ingin mengimplementasikan pembuatan *game 2D* bernama *Taxi Rush* dengan menggunakan metode *GDLC* (Game Development Life Cycle). *Taxi Rush* merupakan game yang dibuat oleh peneliti dan beberapa orang lain pada *Unity Engine*. *Taxi Rush* memiliki genre *racing* dan sub-genre *time-attack*. Platform publikasi *game* yang digunakan peneliti adalah *itch.io* dengan platform *game WebGL*. Penelitian ini dibuat untuk mengimplementasikan metode *GDLC* dalam pembuatan *game* pada platform *WebGL*. Alasan mengapa penelitian ini dilakukan dikarenakan peneliti melihat bahwa industri dan pengembangan game di negara Indonesia masih sedikit. Dengan sedikitnya pengembang game di Indonesia, sumber belajar untuk membuat mengembangkan *game* yang baik dan benar dalam Bahasa Indonesia sangat terbatas. Oleh karena itu peneliti memutuskan untuk menghibahkan ilmu pengembangan game yang dipelajari peneliti selama beberapa tahun (Dengan menggunakan Bahasa Indonesia agar dapat dibaca dan diakses oleh seluruh masyarakat*



Indonesia) dalam bentuk jurnal ini kepada Negara Indonesia.

Kata Kunci: *Game, 2D Game, Game Development, Unity Engine, GDLC.*

1. PENDAHULUAN

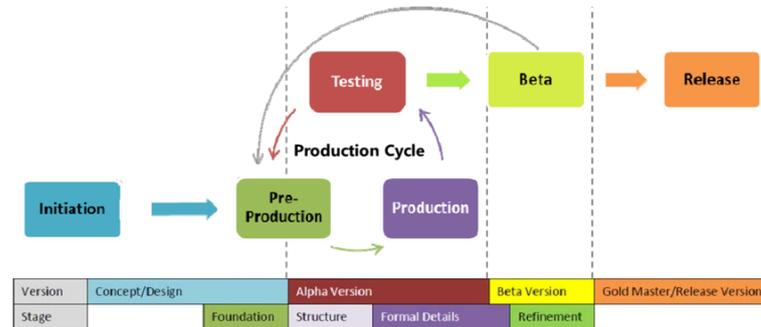
Game (Permainan) adalah sebuah sarana hiburan untuk menyenangkan diri sendiri. Pada zaman dahulu, *game* sudah ada di seluruh dunia termasuk Indonesia. Beberapa *game* tradisional yang sering kita dengar seperti Petak Umpet, Gobak Sodor, Lompat Tali Karet, layangan, dan lain-lain. Namun seiring berkembangnya zaman, perangkat teknologi mulai bermunculan dari Komputer, *game konsol*, hingga era *smartphone*. Kemunculan perangkat teknologi ini memungkinkan manusia membuat *game* digital hingga membuat era *game* digital yang pemainannya kita sebut dengan *video game*. Saat ini *video game* telah merajalela dan dimainkan oleh manusia dengan berbagai usia baik menggunakan *smartphone*, *PC* (*Personal Computer*), *game konsol*, dan lain-lain.

Game terbagi dalam beberapa jenis permainan yang lebih dikenal dengan istilah genre. Genre *game* bisa terdiri dari sebuah genre saja atau bisa merupakan gabungan dari dua atau lebih genre. [1] Genre utama dalam *game* yang diimplementasikan pada penelitian ini adalah *racing* dengan sub-genre *time-attack*. *Game* bergenre *racing* merupakan *game* atau permainan yang sangat banyak diminati oleh para pecinta *game*, *gameplay* pada *game* ini mudah dimainkan sehingga banyak pemain *game* pasti pernah memainkan *game* dengan genre ini. [2] *Game* bergenre *racing* seringkali adalah persaingan diantara pemain dengan menggunakan mobil. Namun, terdapat beberapa *mode* dari *game racing* bersaing berdasarkan waktu. *Time-attack* adalah mode permainan di mana Anda balapan sendirian di lapangan dan bersaing melawan waktu. Hasil waktu Anda akan ditampilkan jika Anda menyelesaikan *time-attack* dalam batas waktu. Permainan akan berakhir ketika waktunya habis atau ketika anda tidak dapat menyelesaikan balapan dalam batas waktu tertentu. *Game typology* pada *game Taxi Rush* adalah *Reach*. *Reach* umumnya mengacu pada gagasan bahwa pemain harus bertindak sedemikian rupa sehingga karakter mencapai lokasi spasial tertentu di dunia permainan.

Dalam pembuatan *game*, diperlukan sebuah *game engine* untuk membuat *game* tersebut. *Game engine* merupakan kumpulan modul kode simulasi yang secara tidak langsung menentukan *game logic* atau level data. *Game engine* mencakup modul untuk menangani *input*, *output* (berupa *3D Rendering*, gambar *2D*, ataupun suara), *generic physics* atau dinamika dunia *game*. [3] *Game engine* yang digunakan untuk membuat *game Taxi Rush* adalah *Unity Engine*. (Disebut *Unity 3D* sebelum mendukung pengembangan permainan *2D* maupun *3D*.) *Unity Engine* adalah sebuah *game engine* yang dikembangkan oleh *Unity Technologies*. Perangkat lunak ini pertama kali diluncurkan pada tahun 2005 dan merupakan salah satu dari banyak *game engine* yang digunakan oleh banyak *game* profesional pengembang di dunia. *Unity Engine* adalah sebuah *game engine* pengembangan dengan kemampuan untuk *rendering* terintegrasi. Dengan menggunakan fitur-fiturnya yang canggih dan kecepatan kerja yang tinggi. *Unity Engine* dapat membuat program interaktif tidak hanya dalam *2D*, tetapi juga dalam *3D*. [4] *Unity Engine* sendiri dibuat menggunakan bahasa pemrograman *C#*. *C#* adalah bahasa pemrograman sederhana yang dikembangkan oleh *Microsoft Corporation*. *C#* merupakan Bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *windows*, web serta aplikasi yang berbasis *web service*. Bahasa *C#* dipengaruhi oleh berbagai macam bahasa lain seperti *Java* dan *Delphi*. Bahasa *C#* bersifat *general-purpose* dan sederhana yang mampu menulisa program aplikasi dengan baik. [5] Pada penelitian kali ini, peneliti akan mengimplementasi pembuatan *game* berjudul *Taxi Rush* yang dapat dimainkan dalam platform *WEBGL* melalui *web browser*. *Taxi Rush* merupakan *game* yang dibuat oleh peneliti dan beberapa orang lain pada *Unity Engine*. *Taxi Rush* memiliki genre *racing* dan sub-genre *time-attack*. Website publikasi *game* yang digunakan peneliti adalah *itch.io* dengan platform *game WebGL*. *WebGL* singkatan dari (*Web-based Graphics Library*) merupakan Platform *Application Programming Interfaces (APIs)* library grafis *3D* yang memungkinkan browser internet untuk membuat adegan *3D* dengan cara sederhana dan efisien. [6] Namun *game* yang dijalankan dengan platform *WebGL* juga dapat menjalankan *game 2D* juga. Penelitian ini dibuat untuk mengimplementasikan metode *GDLC* dalam pembuatan *game* pada platform *WebGL*. Alasan mengapa penelitian ini dilakukan dikarenakan peneliti melihat bahwa industri dan pengembangan *game* di negara Indonesia masih sedikit. Dengan sedikitnya pengembang *game* di Indonesia, sumber belajar untuk membuat mengembangkan *game* yang baik dan benar dalam Bahasa Indonesia sangat terbatas. Oleh karena itu peneliti memutuskan untuk menghibahkan ilmu pengembangan *game* yang dipelajari peneliti selama beberapa tahun (Dengan menggunakan Bahasa Indonesia agar dapat dibaca dan diakses oleh seluruh masyarakat Indonesia) dalam bentuk jurnal ini kepada Negara Indonesia.

2. METODE PENELITIAN

Metodologi penelitian yang digunakan pada penelitian ini adalah *GDLC*. Sebenarnya metode *GDLC* memiliki berbagai macam bentuk yang terkadang beberapa tahapnya ada yang digabungkan dengan yang lain. Namun pada saat ini metode yang kita gunakan adalah *Proposed GDLC* yang terdiri dari 6 tahap: *Initiation*, *Pre-Production*, *Production*, *Testing*, *Beta*, dan *Release*. [7]



Gambar 1. *Proposed GDLC Model*

2.1 Initiation

Tahap *initiation* adalah proses awal dari sebuah pengembangan *game*. *Output* dari tahap *initiation* adalah konsep *game* dan deskripsi permainan yang sangat sederhana. [8] Pada tahap ini produser *game* menentukan *game* seperti apa yang dibuat meliputi: Jenis permainan yang dibuat, *game* skenario, karakter *game*, cerita dalam *game*, target permainan, *game engine*, dan platform *game* apa yang digunakan.

Berikut merupakan deskripsi permainan yang dibuat oleh peneliti dan tim:

Nama Game: *Taxi Rush*

Tema yang diangkat: Edukasi Lalu Lintas

Karakter Game: Berupa mobil taksi online.

Game skenario dan cerita dalam permainan: Anda adalah seorang pengemudi taksi online yang harus mengantar penumpang anda yang sedang berkegas.

Target permainan: Menggunakan *typology game reach*, yang artinya pemain harus sampai tujuan dengan waktu tertentu untuk memenangkan *game* tersebut.

Game Engine: *Unity Engine*

Platform: Dipublikasikan di situs web *itch.io* dengan platform *game WebGL*.

2.2 Pre-Production

Tahap *Pre-Production* adalah fase yang mana melibatkan penciptaan dan revisi desain *game* dan pembuatan prototipe permainan. Desain *game* bertujuan untuk berfokus mendefinisikan genre permainan, *gameplay*, *game* mekanik, alur cerita, karakter, tantangan, faktor kesenangan, aspek teknis, dan dokumentasi elemennya dalam *GDD (Game Design Document)*. Tahap *Pre-Production* akan berakhir pada saat revisi atau perubahan desain *game* sudah disetujui dan didokumentasi pada *GDD*. [9]

Pada tahap ini peneliti menambahkan hal seperti berikut:

Genre permainan: *Time-attack*

Gameplay: Pada awal *game* akan menunjukkan sebuah *cutscene* berupa cerita. Cerita yang ada pada setiap level berbeda. *Game* mekanik: Mekanisme *game* adalah menggunakan key “*WASD*” pada keyboard *PC* untuk menggerakkan pemain (mobil) kepada titik tujuan yang ditentukan. *Game* terdiri dari 5 level dengan tingkat kesulitan yang meningkat setiap levelnya. Tingkat kesulitan yang berubah karena faktor letak titik tujuan dari titik awal serta jenis dan jumlah musuh.

Karakter: Mobil 2D yang digunakan untuk permainan *Top-Down*. *Asset* mobil tersebut akan dibuat oleh tim.

Tantangan: Pemain akan diberikan waktu 1 menit untuk menyelesaikan level. Dikarenakan semakin tinggi levelnya, semakin sulit juga *gamenya* maka peneliti dan tim memberikan 3 jenis power up yakni:

Immunity: Nyawa mobil yang tidak ada habis-habisnya. Setiap level hanya diberikan 1.

Increase Speed: Menambahkan maksimal kecepatan mobil 10km/jam. Hanya diberikan 1 per level.

Increase time: Menambahkan waktu sebesar 10 detik setiap kali power up ini diterima. Power up ini dipasang sekitar 20

lebih pada setiap level untuk menyesuaikan waktu yang diberikan pada awal game yang jumlahnya sedikit.

2.3 Production

Tahap *Production* bertujuan untuk menyempurnakan tahap *Pre-Production*. Tahap ini memiliki fokus untuk menerjemahkan rancangan yang produser game buat dalam *GDD* menjadi nyata dengan melakukan *asset creation*, *programming*, dan integrasi antara *asset game* dan *source code*. [10]

Pada tahap ini tim peneliti membuat asset game berupa mobil 2D sebagai karakter serta *asset cutscene* pada game yang dibuat.



Gambar 2. Asset berupa mobil dan cutscene yang dibuat pada game

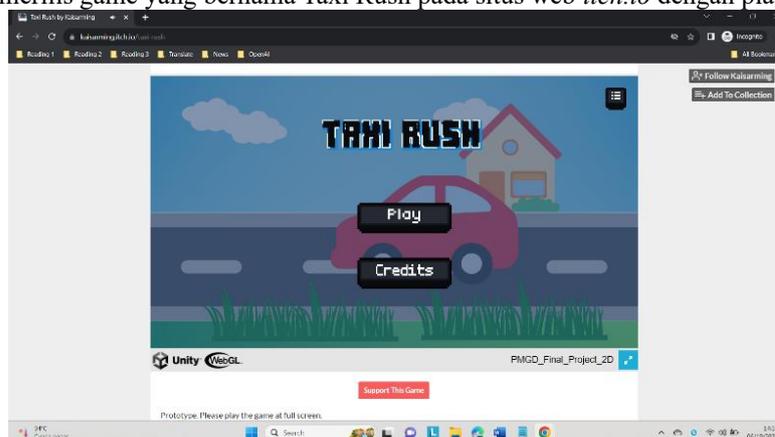
2.4 Testing dan Beta

Tahap *Testing* adalah pengujian terhadap prototipe yang dibangun oleh pengembang game untuk pengujian kegunaan dan fungsi. [11] Pada tahap ini bertujuan untuk mencoba game yang ada dan memperbaiki *bug* pada *game*. (Jika ada.) Sebenarnya kedua tahap ini terpisah antara satu sama lain, perbedaannya adalah *testing* dilakukan oleh diri kita sendiri sedangkan *Beta* adalah *testing* yang dilakukan oleh pihak ketiga. Namun peneliti tidak menggunakan pihak ketiga untuk melakukan *testing* dalam pembuatan game *Taxi Rush*. Oleh karena itu, peneliti menggabungkan keduanya pada penelitian ini.

2.5 Release

Tahap *release* merupakan tahap akhir dalam *GDLC* yang dimana game siap dirilis ke publik. Rilis melibatkan peluncuran produk, dokumentasi proyek, berbagi pengetahuan, *post-mortem*, dan perencanaan untuk pemeliharaan dan perluasan *game*. [12]

Pada tahap ini peneliti merilis game yang bernama *Taxi Rush* pada situs web *itch.io* dengan platform *game WebGL*.



Gambar 3. Gambar game pada website *itch.io*



3. HASIL DAN PEMBAHASAN

Pada bagian ini, peneliti akan membahas fitur, *scene*, dan juga *gameplay* dari permainan dengan lebih terperinci lagi.

3.1 Tampilan *splash screen*



Gambar 4. *Splash screen Unity* Gambar 5. *Splash screen ICEI* Gambar 6. *Splash screen logo publisher*



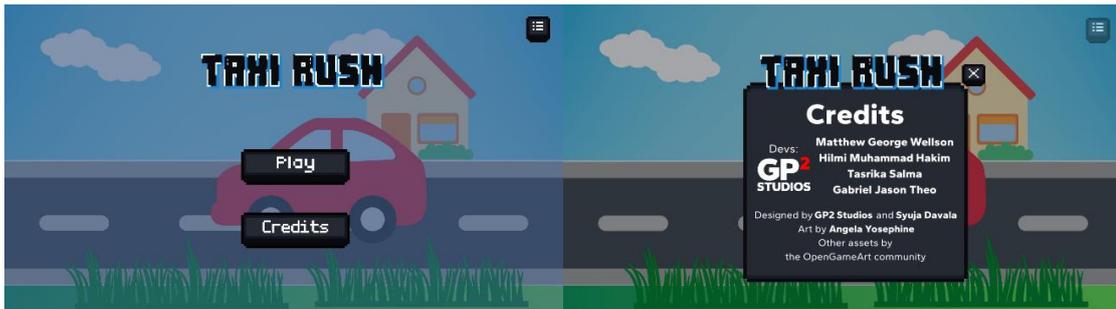
Gambar 7. Durasi *splash screen* pada gambar 4

```
· FadeIn(splashImage1);
· yield return new WaitForSeconds(2.5f);
· FadeOut(splashImage1);
· yield return new WaitForSeconds(2f);
· FadeIn(splashImage2);
· yield return new WaitForSeconds(2.5f);
· FadeOut(splashImage2);
· yield return new WaitForSeconds(2f);
· SceneManager.LoadScene("MainMenu");
```

Gambar 8. Coding *splash screen image*

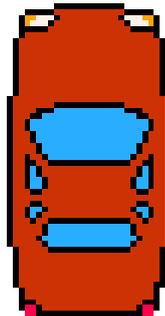
Gambar 4, 5, dan 6 diatas merupakan gambar *splash screen* ketika *game* dijalankan. *Splash screen* tersebut dijalankan mulai dari gambar 4 lalu ke gambar 5, dan terakhir akan memunculkan *splash screen* gambar 6. Gambar 7 menunjukkan *splash screen* “Made With Unity” yang ada pada gambar 4 berdurasi selama 2 detik. *Splash screen* “Made With Unity” merupakan *splash screen* bawaan *Unity Engine* yang akan muncul pertama kali *game* dijalankan jika *game* tersebut dibuat dengan menggunakan *Unity Engine*. Untuk menghilangkan *splash screen* “Made With Unity”, developer harus *upgrade Unity Plan* mereka ke *Unity Pro*, *Unity Enterprise*, atau *Unity Industry*. Sedangkan gambar 8 merupakan *source code* untuk memunculkan dan menghilangkan *splash screen* pada gambar 5 dan 6. Pada saat gambar 5 muncul, aplikasi akan menunggu 2.5 detik sebelum menghilangkan logo pada gambar 5. Kemudian sebelum memunculkan logo pada gambar 6, *game* ini akan menunggu selama 2 detik sebelum memunculkannya. Dan proses yang sama terjadi sebelum akhirnya menuju pada *scene* “MainMenu.”

3.1.1 Main Menu

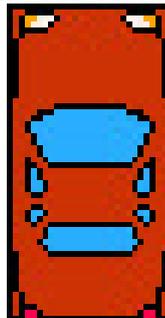


Gambar 9. *Main Menu*

Gambar 10. *Credits Panel*



Gambar 11. Mobil PNG

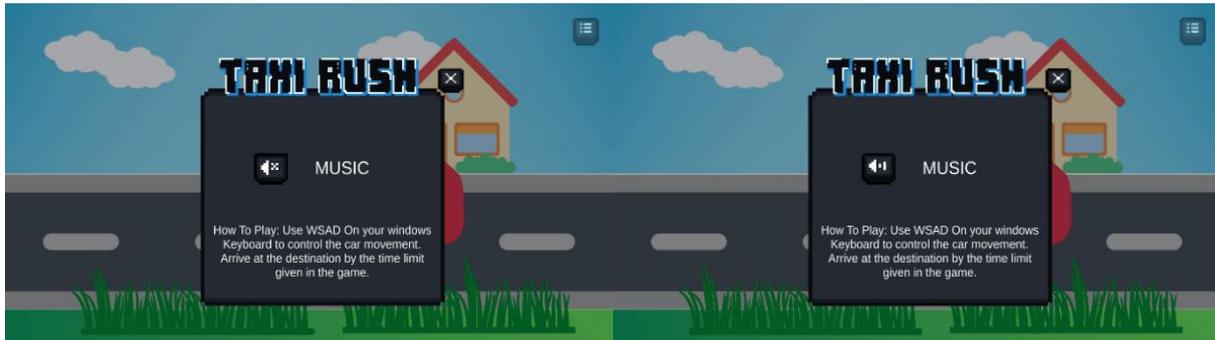


Gambar 12. Gambar mobil JPEG

Pada gambar 9 merupakan tampilan menu utama pada *game Taxi Rush*. Pada bagian atas menampilkan judul *game* yang dibuat menjadi format gambar PNG. Format PNG (*Portable Network Graphics*) merupakan format yang menggunakan metode *kompresi lossless* dan sering digunakan pada *website* dikarenakan ukurannya yang relatif lebih kecil sehingga mempercepat akses *website* itu sendiri. File PNG juga menggunakan kompresi yang tidak menghilangkan data (*lossless compression*). [13] Itulah alasan mengapa banyak *game* yang dibuat hampir selalu menggunakan *sprite file PNG*. Yakni gambar yang memiliki kualitas tinggi dan *transparent background*. PNG yang mempunyai *transparent background* tidak ada apapun pada *background pixel* yang membuat apa yang dibelakang gambar ini terlihat. Dengan mempunyai format transparan ini, gambar PNG dapat digunakan dengan baik karena tidak adanya *background pixel* dibelakangnya. Bayangkan jika yang dibutuhkan dalam sebuah gambar hanyalah sebuah *sprite* karakter 2D, pasti akan jelek jika *background pixel* berwarna merah ada. Karena *background pixel* itu akan menghalangi *sprite* lain yang ada pada *game*. Contoh dari perbedaan gambar PNG dengan gambar dengan format lain (Pada jurnal ini dipakai format JPEG) adalah adanya *background pixel* berwarna merah pada format JPEG. Selanjutnya kita akan membahas tentang *button* yang ada. Pada bagian bawah dari judul mempunyai 2 jenis *button* yang menuju pada *scene Unity* yang berbeda bernama “Play” dan “Credits”. “Credits” merupakan tombol yang ketika di klik akan memunculkan “credits panel” seperti yang ada pada gambar 10. Sedangkan tombol “Play” akan menuju ke “level scene” yang ada pada gambar 15. Pada kanan atas mempunyai tombol “option panel” untuk menuju seperti pada gambar 13. Alasan mengapa *game Taxi Rush* tidak menggunakan “quit button” karena *game WebGL* tidak memerlukan tombol “quit button”. Berbeda dengan *game* yang dijalankan pada *mobile (Android dan iOS)* yang memerlukan tombol “quit” agar *apps* dapat di *terminate* (ditutup). Ditambah *game* yang dibuat dengan platform *WebGL* karena *web browser* sudah memiliki tombol “quit” sendiri pada kanan atas *browser*.

Credits menampilkan daftar pembuat *game Taxi Rush*. *Game Taxi Rush* dibuat oleh 4 orang *game programmer* yakni peneliti sendiri, Hilmi Muhammad Hakim, Tasrika Salma, dan Gabriel Jason Theo. Sedangkan *design game Taxi Rush* dibuat oleh Syuja Davala, dan terakhir *asset game* berupa mobil pemain dan *cutscene* dibuat oleh Angela Yosephine. Untuk *asset-asset* lainnya, peneliti dan tim mengambil *asset* tersebut dari *Unity Asset Store* dan *OpenGameArt community* yang berlisensi CC0.

CC0 adalah sebuah CC yang bertujuan untuk melepaskan dan mengesampingkan semua hak cipta dan hak terkait dalam suatu karya di seluruh dunia. CC0 sangat populer dalam bidang *open data* dan bermanfaat bagi penerbit karya yang ingin karya mereka memiliki izin terbuka [14]. Singkatnya, semua hak kekayaan intelektual yang memiliki lisensi CC0 dapat disalin, dimodifikasi, didistribusikan, dan memakainya secara komersial tanpa meminta izin dari pemilik hak kekayaan intelektual itu.

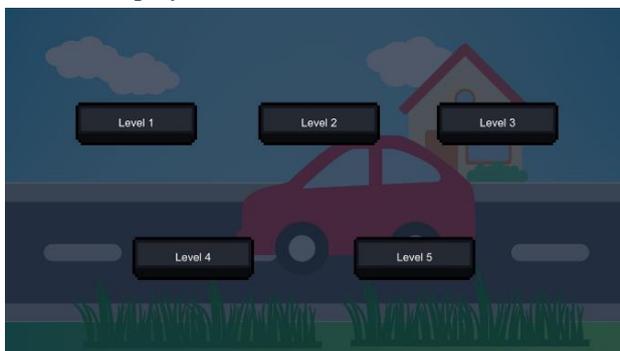


Gambar 13. Option Panel

Gambar 14. Music mute Option Panel

Pada gambar 13 ditampilkan sebuah *panel* yang berisikan tombol musik yang dapat menyalahkan atau mematikan *background music* yang ada pada *game*. Ketika tombol tersebut di klik, maka *icon* dari tombol tersebut akan berubah seperti yang ada pada gambar 14 dan *background music* dalam *game* akan dimatikan. Kemudian ada deskripsi untuk cara bermain *game Taxi Rush* yang bertuliskan, “Use WSAD On your Windows Keyboard to control the car movement. Arrive at the destination by the time limit given in the game.” Instruksi tersebut jika diartikan ke dalam bahasa Indonesia mempunyai arti, “Pakai key pada Windows yang bertuliskan WSAD untuk menggerakkan mobil. Pemain harus sampai pada tujuan dengan waktu yang diberikan.”

3.1.2 Gameplay



Gambar 15. Level Scene



Gambar 16. Cutscene

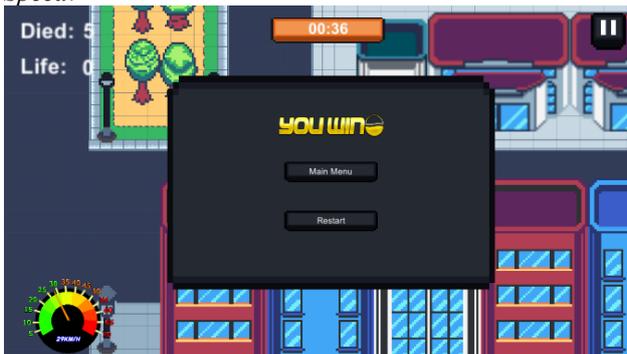
Pada gambar 15, terdapat *level scene* yang terdapat 5 tombol di layar *game*. Kelima tombol tersebut akan membawa pemain kepada *level game* yang berbeda-beda. Dimulai dari *level 1* yang kesulitannya paling mudah sampai dengan *level 5* yang kesulitannya paling tinggi.

Pada saat permulaan yang ada pada setiap level, terdapat *cutscene* yang berisikan sebuah cerita di dalam *game*. Cerita tersebut bertujuan untuk memberi tau pada pemain tujuan dari *customer taxi online*. Pemain dapat menekan tombol *Next* untuk menjalankan *text* pada *cutscene* yang ada. Pada saat *cutscene* selesai dijalankan, pemain dapat memulai *game*. Waktu yang bertuliskan “01:00” akan mulai berjalan hingga “00:00” Ikutilah penunjuk jalan agar sampai pada destinasi dengan waktu yang diberikan. Pada kanan atas terdapat *pause button* yang berguna ketika di klik, maka *game* tidak dapat digerakan dan akan memunculkan sebuah *panel* yang ada pada gambar 16. Pada bagian kiri atas terdapat “Died” dan “Life.” “Died” merupakan hitungan berapa kali pemain mati saat main *game* sedangkan “Life” merupakan jumlah nyawa yang diberikan pada pemain. Ketika “Life” mencapai 0, maka pemain akan meledak (mati) dan angka yang ada pada kanan “Died” akan bertambah satu.

**Gambar 17.** *Pause Panel***Gambar 18.** *You win panel level 1 sampai level 4*

Pada gambar 17 terdapat sebuah “*pause panel*” yang akan muncul jika *pause button* di klik pada saat *game Taxi Rush* dimainkan. Tombol ini dibuat agar pemain dapat memberhentikan permainan jika ingin beristirahat sementara sebelum meneruskan permainan kembali. Untuk memulai permainan kembali, tekan tombol “*continue*”. Untuk kembali ke *Main Menu*, tekan tombol “*Main Menu*.” Dan untuk memulai kembali level yang ada dalam game, tekan tombol “*Restart*.”

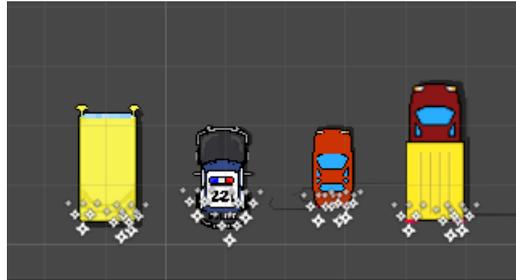
Gambar 18 merupakan gambar yang menunjukkan “*You Win*” panel yang akan muncul ketika pemain memenangkan permainan. Pemain dapat memenangkan permainan jika mereka sampai kepada tujuan. Pada “*You Win*” panel terdiri dari 3 tombol yakni: “*Main Menu*”, “*Restart*”, dan “*Next Level*”. Tombol “*Main Menu*” bertujuan untuk kembali pada menu utama. Tombol “*Restart*” bertujuan untuk memulai kembali permainan dengan level yang sama. Sedangkan tombol “*Next Level*” digunakan ketika pemain ingin menuju level berikutnya. Pada bagian kiri bawah terdapat *speedometer*. *Speedometer* digunakan untuk melihat seberapa kencang laju mobil yang ada pada *game*. Batas laju mobil pada *game* adalah 29km/h, sedangkan yang ada pada gambar 18 adalah *speedometer* yang telah menerima *power up* “*Increase Speed*.”

**Gambar 19.** *You Win Panel Level 5***Gambar 20.** *You Lose Panel*

Gambar 19 merupakan gambar “*You Win panel*” yang akan muncul ketika pemain menyelesaikan *game* pada level 5. Perbedaannya dengan “*You Win panel*” yang ada pada level-level sebelumnya adalah tombol “*Next Level*” dihilangkan dikarenakan level 5 adalah level terakhir yang ada pada *game Taxi Rush*.

“*You lose*” panel akan muncul ketika waktu yang diberikan habis. Waktu yang diberikan pada pemain pada saat permainan mulai adalah 1 menit. Namun pemain dapat mendapatkan waktu tambahan setiap kali pemain mendapatkan *power up* “*Increase Time*” pada *game* sebesar sepuluh detik per *power up*. Ketika batas waktu habis, maka akan muncul “*You Lose*” panel seperti yang ada pada gambar 20. Pada “*You Lose panel*” terdapat tulisan “*You Lose*” yang dibuat menggunakan gambar PNG. Kemudian terdapat dua tombol bertuliskan “*Main Menu*” yang dapat digunakan untuk kembali ke menu utama dan juga tombol “*Restart*” yang dapat digunakan untuk pemain mengulangi level yang sama pada *game*.

3.1.3 *Sprite Game*

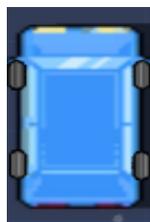


Gambar 21. Mobil Komputer/NPC

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EnemyPatrolling : MonoBehaviour
6 {
7     public Transform[] waypoints;
8     public float speed = 5f;
9     public float rotationSpeed = 2f;
10    public float waypointRadius = 1f;
11
12    private int currentWaypoint = 0;
13
14    private void Update()
15    {
16        // Calculate the distance to the current waypoint
17        float distanceToWaypoint = Vector2.Distance(transform.position, waypoints[currentWaypoint].position);
18
19        // If the car has reached the current waypoint, move to the next one
20        if (distanceToWaypoint < waypointRadius)
21        {
22            currentWaypoint++;
23            if (currentWaypoint >= waypoints.Length)
24            {
25                currentWaypoint = 0;
26            }
27        }
28
29        // Calculate the direction to the current waypoint
30        Vector2 direction = waypoints[currentWaypoint].position - transform.position;
31
32        // Rotate the car to face the direction of the waypoint
33        float angle = Mathf.Atan2(direction.y, direction.x) * Mathf.Rad2Deg;
34        Quaternion targetRotation = Quaternion.Euler(new Vector3(0, 0, angle - 90));
35        transform.rotation = Quaternion.Lerp(transform.rotation, targetRotation, rotationSpeed * Time.deltaTime);
36
37        // Move the car towards the current waypoint
38        transform.position = Vector2.MoveTowards(transform.position, waypoints[currentWaypoint].position, speed * Time.deltaTime);
39    }
40 }
41
```

Gambar 22. Script Mobil Komputer/NPC

Pada gambar 21, terdapat keempat mobil yang digerakkan menggunakan *Artificial Intelligence*. *Artificial Intelligence* (*AI*) adalah kecerdasan buatan yang diimplementasikan pada sistem komputer yang memungkinkan pemain untuk bersaing melawan komputer dalam permainan yang mirip dengan pemain lain. Kecerdasan buatan, sering dikenal sebagai *machine learning*, adalah tiruan dari kecerdasan manusia yang telah ada dilatih untuk berpikir seperti manusia. [15] Gambar 22 merupakan *script AI* untuk menggerakkan mobil. Pada kode baris 8 terdapat *speed* untuk memberi kecepatan pada mobil. Dalam *private void update* yang berjalan setiap *frame*, mobil *AI* akan mengalkulasi jarak mobil ke *waypoint*. *Waypoint* sendiri berbentuk dalam *gameObject* pada *Unity* yang digunakan sebagai tempat tujuan. Ketika jarak sudah di kalkulasi, maka mobil akan mendarangi *waypoint* tujuan.



Gambar 23. Player (Mobil)



Pada gambar 23 terdapat mobil/*player*. Mobil ini digerakkan oleh pemain untuk menyelesaikan tugas/misi yang diberikan pada setiap level yang ada pada *game Taxi Rush*.



Gambar 24. *Directional Arrow*

Pada gambar 24 terdapat *directional arrow*. *Directional arrow* digunakan sebagai penunjuk jalan untuk mencapai lokasi tujuan.



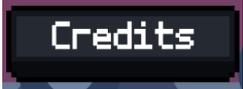
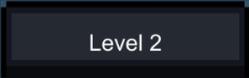
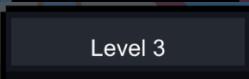
Gambar 25. *Car Explosion*

Pada gambar 25 terdapat *car explosion*. *Car explosion* akan muncul di tempat *player* setelah nyawa *player* sama dengan 0.

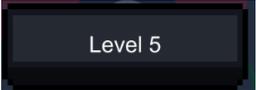
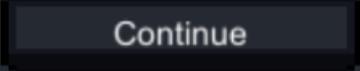
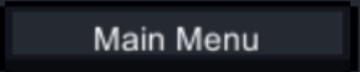
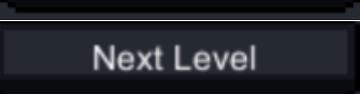
3.2 Implementasi

Dibawah ini merupakan hasil pengujian *button* yang ada pada *game Taxi Rush*:

Tabel 1. Pengujian *Button*

No.	Tombol	Hasil	Keterangan
1		Pindah ke <i>scene</i> " <i>GameLevel</i> "	Berhasil
2		Menampilkan panel <i>credits</i>	Berhasil
3		Menutup sebuah panel	Berhasil
4		Menampilkan panel untuk mengatur musik pada menu dan peraturan permainan	Berhasil
5		Pindah ke <i>scene</i> " <i>Level1</i> "	Berhasil
6		Pindah ke <i>scene</i> " <i>Level2</i> "	Berhasil
7		Pindah ke <i>scene</i> " <i>Level3</i> "	Berhasil
8		Pindah ke <i>scene</i> " <i>Level4</i> "	Berhasil



9		Pindah ke <i>scene</i> "Level5"	Berhasil
10		Menampilkan dialog percakapan selanjutnya	Berhasil
11		Memberhentikan <i>game</i> sementara	Berhasil
12		Melanjutkan <i>game</i>	Berhasil
13		Pindah ke <i>scene</i> "MainMenu"	Berhasil
14		Restart level pada <i>game</i>	Berhasil
15		Menuju level <i>game</i> berikutnya	Berhasil

Berdasarkan hasil pengujian dari setiap *button* yang ada, setiap *button* berfungsi dengan baik tanpa adanya kendala yang menghalangi *button* tersebut berfungsi.

Selanjutnya merupakan pengujian terhadap *key* pada *keyboard PC* untuk melakukan tes apakah semua fungsi berfungsi dengan baik atau tidak:

Tabel 2. Pengujian input *player*

Input	Process	Output	Hasil Pengujian
Tombol <i>key</i> "W"	<pre>private void FixedUpdate() { turningAmount = -Input.GetAxis("Horizontal"); speed = 0f; if (Input.GetAxis("Accelerator") > 0) { speed = Input.GetAxis("Accelerator") * forward; } else if (Input.GetAxis("Reverse") < 0) { speed = Input.GetAxis("Reverse") * reverse; } direction = Mathf.Sign(Vector2.Dot(rb.velocity, rb.GetRelativeVector(Vector2.up))); rb.rotation += turningAmount * turningForce * rb.velocity.magnitude * direction; rb.AddRelativeForce(Vector2.up * speed); rb.AddRelativeForce(-Vector2.right * rb.velocity.magnitude * turningAmount / 2); } </pre>	<i>Player</i> bergerak keatas	Berhasil
Tombol <i>key</i> "A"	<pre>private void FixedUpdate() { turningAmount = -Input.GetAxis("Horizontal"); speed = 0f; if (Input.GetAxis("Accelerator") > 0) { speed = Input.GetAxis("Accelerator") * forward; } else if (Input.GetAxis("Reverse") < 0) { speed = Input.GetAxis("Reverse") * reverse; } direction = Mathf.Sign(Vector2.Dot(rb.velocity, rb.GetRelativeVector(Vector2.up))); rb.rotation += turningAmount * turningForce * rb.velocity.magnitude * direction; rb.AddRelativeForce(Vector2.up * speed); rb.AddRelativeForce(-Vector2.right * rb.velocity.magnitude * turningAmount / 2); } </pre>	<i>Player</i> bergerak ke kiri	Berhasil



Tombol key "S"	<pre>private void FixedUpdate() { turningAmount = -Input.GetAxis("Horizontal"); speed = 0f; if (Input.GetAxis("Accelerator") > 0) { speed = Input.GetAxis("Accelerator") * forward; } else if (Input.GetAxis("Reverse") < 0) { speed = Input.GetAxis("Reverse") * reverse; } direction = Mathf.Sign(Vector2.Dot(rb.velocity, rb.GetRelativeVector(Vector2.up))); rb.rotation += turningAmount * turningForce * rb.velocity.magnitude * direction; rb.AddRelativeForce(Vector2.up * speed); rb.AddRelativeForce(-Vector2.right * rb.velocity.magnitude * turningAmount / 2); }</pre>	<p>Player bergerak ke bawah</p> <p>Berhasil</p>
Tombol key "D"	<pre>private void FixedUpdate() { turningAmount = -Input.GetAxis("Horizontal"); speed = 0f; if (Input.GetAxis("Accelerator") > 0) { speed = Input.GetAxis("Accelerator") * forward; } else if (Input.GetAxis("Reverse") < 0) { speed = Input.GetAxis("Reverse") * reverse; } direction = Mathf.Sign(Vector2.Dot(rb.velocity, rb.GetRelativeVector(Vector2.up))); rb.rotation += turningAmount * turningForce * rb.velocity.magnitude * direction; rb.AddRelativeForce(Vector2.up * speed); rb.AddRelativeForce(-Vector2.right * rb.velocity.magnitude * turningAmount / 2); }</pre>	<p>Player bergerak ke kanan</p> <p>Berhasil</p>

Berdasarkan table diatas, dapat disimpulkan bahwa semua tombol untuk menggerakkan *player* dapat digunakan semuanya. Selanjutnya adalah hasil dari fungsi *power up* yang player oleh peneliti dan tim:

Tabel 3. Pengujian Power Up Player/Mobil

Power Up	Process	Output	Hasi Pengujian
		<p>Life dari <i>player</i> menjadi tidak terbatas.</p>	<p>Berhasil</p>

Immunity



Increase Speed



Increase Time

```

IncreaseSpeed.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class IncreaseSpeed : MonoBehaviour
6 {
7     public GameObject powerUp;
8     private MainCar mainCar;
9     private SpeedoMeter speedoMeter;
10
11     private void OnTriggerEnter2D(Collider2D collision)
12     {
13         if (collision.gameObject.tag == "Player")
14         {
15             powerUp.SetActive(false);
16             mainCar = GameObject.FindObjectOfType<MainCar>();
17             speedoMeter = GameObject.FindObjectOfType<SpeedoMeter>();
18             mainCar.forward += 10;
19             speedoMeter.maxSpeedAngle = -100f;
20         }
21     }
22 }
    
```

```

IncreaseTime.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class IncreaseTime : MonoBehaviour
6 {
7     public float addedTime = 10f;
8     private GameTimeLimit gameTimeLimit;
9
10    private void OnTriggerEnter2D(Collider2D collision)
11    {
12        if (collision.CompareTag("Player"))
13        {
14            gameTimeLimit = GameObject.FindObjectOfType<GameTimeLimit>();
15            // tambahkan waktu pada timer
16            gameTimeLimit.timeRemaining += addedTime;
17            gameTimeLimit.UpdateTimerText();
18
19            // hapus game object power-up
20            Destroy(gameObject);
21        }
22    }
23 }
    
```

Batas kecepatan player bertambah 10km/jam

Berhasil

Batas waktu pada level tersebut bertambah 10 detik.

Berhasil

4. KESIMPULAN

Taxi Rush adalah sebuah *game* yang dibuat menggunakan *Unity Engine*. *Unity Engine* merupakan salah satu *game engine* populer yang digunakan oleh *solo game developer*, *studio game* dan perusahaan besar. *Taxi Rush* dibuat menggunakan *core 2D* yang artinya *game* ini adalah *game 2 dimensi*. *Taxi Rush* memiliki genre *racing* dan sub-genre *time-attack*. *Game* bergenre *racing* seringkali adalah persaingan diantara pemain dengan menggunakan mobil. Namun, terdapat beberapa *mode* dari *game racing* bersaing berdasarkan waktu. *Time-attack* adalah *mode* permainan di mana Anda balapan sendirian di lapangan dan bersaing melawan waktu. Hasil waktu Anda akan ditampilkan jika Anda menyelesaikan *time-attack* dalam batas waktu. Permainan akan berakhir ketika waktunya habis atau ketika anda tidak dapat menyelesaikan *time-attack* dalam batas waktu tersebut. *Game typology* pada *game Taxi Rush* adalah *Reach*. *Reach* umumnya mengacu pada gagasan bahwa pemain harus bertindak sedemikian rupa sehingga karakter mencapai lokasi spasial tertentu di dunia permainan. *Game Taxi Rush* dapat dimainkan pada platform situs web bernama *itch.io* dengan platform *WebGL*. *Game Taxi Rush* dibuat dengan menggunakan metode *GDLC* atau lebih dikenal sebagai *Game Development Life Cycle*. *GDLC* sendiri terdiri dari 6 tahap: *Initiation*, *Pre-Production*, *Production*, *Testing*, *Beta*, dan *Release*. Setiap tahap merupakan tahap-tahap yang penting dalam *GDLC*. Dalam tahap *Initiation* dapat dibilang tahap untuk mendesain awal *game*. Pada tahap ini produser *game* menentukan *game* seperti apa yang dibuat yang dimulai dengan jenis permainan yang dibuat, *game* skenario, karakter *game*, cerita dalam *game*, target permainan, *game engine*, dan platform *game* apa yang digunakan secara singkat. Tahap kedua merupakan tahap *Pre-Production* yang mana pada tahap ini merevisi dan memperbaiki apa yang produser *game* buat pada tahap pertama dan menuliskannya dalam *GDD*. Tahap ketiga yakni *Production*, produser *game* sudah mengerjakan *game* tersebut dalam *game engine* yang mengintegrasikan *asset game* dan juga *source code/script*. Sedangkan tahap keempat (*Testing*) dan kelima (*Beta*) merupakan tahap pengecekan *game* dan memperbaiki *bug* dalam *game*. Kedua tahap digabungkan dikarenakan peneliti tidak menggunakan pihak orang ketiga dalam tahap kelima. Terakhir, pada tahap *Release*, peneliti mempublikasi *game Taxi Rush* kedalam platform *itch.io* dengan platform *WebGL*. Dengan menggunakan platform *WebGL*, *user* dapat memainkan *game* tanpa perlu mendownload *game* terlebih dahulu sehingga dapat langsung dimainkan pada *browser web* masing-masing.



Kesimpulan yang ada pada penelitian ini adalah bahwa implementasi metode *GDLC* dapat diterapkan dalam pembuatan *Game Taxi Rush*. Semua *button* yang ada pada *game* berfungsi dengan baik dan *game* berhasil berjalan dengan baik pada platform *WebGL* dan telah dipublikasi pada situs web *itch.io*.

UCAPAN TERIMAKASIH

Semua *game* yang dihasilkan dari Hibah Penelitian dan Pengabdian kepada Masyarakat Indonesia dari Matthew George Wellson, Muhammad Hilmi Hakim, Tasrika Salma, Gabriel Jason Theo, Syuja Davala, dan Angela Yosephine. Oleh karena itu, saya, Matthew George Wellson, selaku penulis jurnal ini berterima kasih pada pihak-pihak terkait untuk izin dalam mengangkat *game Taxi Rush* sebagai topik jurnal ini.

DAFTAR PUSTAKA

- [1] R. S. Putra and D. Y. Utami, "Pemanfaatan Virtual Reality Pada Perancangan Game Fruit Slash Berbasis Android Menggunakan Unity 3D," *Jurnal Teknik Komputer*, no. 2, pp. 25–30, 2018, doi: 10.31294/jtk.v4i2.3500.
- [2] A. Pujiyanto, "Implementation of Genetic Algorithms in The Application of Car Racing Games," *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIMD)*, vol. 4, no. 1, pp. 29–34, 2021, doi: 10.24014/ijaidm.v4i1.10835.
- [3] M. P. Tjahyadi⁽¹⁾, A. Sinsuw⁽²⁾, V. Tulenan⁽³⁾, and S. Sentinuwo⁽⁴⁾, "Prototipe Game Musik Bambu Menggunakan Engine Unity 3D," *E-journal Teknik Informatika*, vol. 4, no. 2, 2014.
- [4] M. F. Sanjaya, H. Pratiwi, and P. Adytia, "Application of the Finite State Machine Method in the Desktop-Based 'Heroes Of Dawn' RPG Turn-Based Game," *TEPIAN*, vol. 2, no. 2, pp. 69–73, Jun. 2021, doi: 10.51967/tepiian.v2i2.348.
- [5] N. Khairani, J. N. Fadila, and F. Nugroho, "PERANCANGAN GAME 2 DIMENSI PETUALANGAN ANAK MENYELAMATKAN ORANGTUA SEBAGAI MEDIA EDUKATIF BAGI ANAK DENGAN METODE WATERFALL," *Jurnal Teknologi Informasi*, vol. 5, no. 1, 2021.
- [6] N. Fahmi and Ts. Eiva Fatdha, "SATIN-Sains dan Teknologi Informasi journal homepage : <http://jurnal.stmik-amik-riau.ac.id> Aplikasi Perpustakaan Tiga Dimensi (3D) Menggunakan Teknologi WebGL," 2016. [Online]. Available: <http://jurnal.stmik-amik-riau.ac.id>
- [7] R. Ramadan and Y. Widyani, "Game development life cycle guidelines," in *2013 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2013*, IEEE Computer Society, 2013, pp. 95–100. doi: 10.1109/ICACSIS.2013.6761558.
- [8] J. Einstein, U. Citra, B. Vera, R. Bulu, B. Roswita, and L. Nahak, "Pengembangan Media Pembelajaran Game Edukasi Bilangan Pangkat dan Akar menggunakan Genially," *Jurnal Jendela Pendidikan*, vol. 02, 2022.
- [9] R. Andriyat Krisdiawan, "PENERAPAN MODEL PENGEMBANGAN GAMEGDLC (GAME DEVELOPMENT LIFE CYCLE)DALAM MEMBANGUN GAME PLATFORM BERBASIS MOBILE," vol. 2, no. 1, 2019.
- [10] R. Gunawan, T. H. Prastyawan, and Y. Wahyudin, "RANCANG BANGUN GAME EDUKASI PERHITUNGAN DASAR MATEMATIKA SEKOLAH DASAR KELAS 3, 4 DAN 5 MENUNGGUNAKAN CONSTRUCT 2", doi: 10.35969/interkom.v16i1.
- [11] A. Chusyairi, "Game Labirin Let's Clear Up The World Menggunakan Metode Game Development Life Cycle," *Informatics for Educators and Professionals*, vol. 4, no. 2, pp. 183–192, 2020.
- [12] R. Yanwastika Ariyana, E. Susanti, M. Rizqy Ath-Thaariq, and R. Apriadi, "INSOLOGI: Jurnal Sains dan Teknologi Penerapan Metode Game Development Life Cycle (GDLC) pada Pengembangan Game Motif Batik Khas Yogyakarta," *Media Cetak*, vol. 1, no. 6, pp. 796–807, 2022, doi: 10.55123/insologi.v1i6.1129.
- [13] A. Satyapratama and M. Yunus, "ANALISIS PERBANDINGAN ALGORITMA LZW DAN HUFFMAN PADA KOMPRESI FILE GAMBAR BMP DAN PNG." *SSRN Electronic Journal*, Mar. 2016, doi: 10.2139/ssrn.2746044.
- [14] T. Margoni and D. Peters, "Creative Commons Licenses: Empowering Open Access," *SSRN Electronic Journal*, Mar. 2016, doi: 10.2139/ssrn.2746044.
- [15] S. A. Yakan, "Analysis of Development of Artificial Intelligence in the Game Industry," *International Journal of Cyber and IT Service Management*, vol. 2, no. 2, pp. 111–116, May 2022, doi: 10.34306/ijcitsm.v2i2.100.