

## Save Our Sea: Rancangan Game 2D RPG Berbasis Android dengan Metode *Extreme Programming* menggunakan Godot Script

Kevin Asgaryansyah<sup>1</sup>, Giri Wahyu Wiriasto<sup>1</sup>

<sup>1</sup>Jurusan Teknik Elektro – Universitas Mataram, 83127 – Lombok, Indonesia

### ARTICLE INFO

#### Article history:

Received July 07, 2023  
Revised August 25, 2023  
Accepted August 28, 2023

#### Keywords:

Save our sea game;  
RPG 2D Game;  
Extreme Programming method;  
Godot script;  
NPC Non-player Character;

### ABSTRACT

*This research is a development of a 2D role-playing game or RPG game using the extreme programming method. The game's rules have been designed to feature a character, a fish, which will be played by the player, embarking on a mission to combat an evil corporation that is polluting the ocean by disposing of marine waste. The gameplay includes platforming challenges, environmental puzzles, and battles. The technical aspects of the game have been explained, detailing the methods and techniques used in its development. This includes implementing player movement, physics, sprite animations, and interactions with various objects, including algorithms for waste collection and encounters with enemies. The main objectives of the game are to overcome platforming obstacles, solve environmental puzzles, and defeat NPC enemy characters. The progress of the game is measured by collecting various types of waste as an indicator of environmental cleanup. The game has been successfully developed and can be played on a simulation scale using mobile and desktop platforms.*

Penelitian ini merupakan pengembangan dari game role-playing atau game RPG 2D dengan menggunakan metode extreme programming. Aturan permainan telah dirancang untuk menampilkan karakter, seekor ikan, yang akan dimainkan oleh pemain, memulai misi untuk memerangi perusahaan jahat yang mencemari lautan dengan membuang limbah laut. Gameplay nya mencakup tantangan platforming, teka-teki lingkungan, dan pertempuran. Aspek teknis permainan telah dijelaskan, merinci metode dan teknik yang digunakan dalam pengembangannya. Ini termasuk penerapan pergerakan pemain, fisik, animasi, dan interaksi dengan berbagai objek, termasuk algoritma pengumpulan sampah dan pertemuan dengan musuh. Tujuan utama permainan ini adalah untuk mengatasi rintangan platforming, memecahkan teka-teki lingkungan, dan mengalahkan karakter musuh NPC. Kemajuan permainan diukur dengan mengumpulkan berbagai jenis sampah sebagai indikator kebersihan lingkungan. Game tersebut telah berhasil dikembangkan dan dapat dimainkan dalam skala simulasi menggunakan platform seluler dan desktop.

### Corresponding Author:

Giri Wahyu Wiriasto, Jurusan Teknik Elektro Universitas Mataram, Jalan Majapahit 63 Kota Mataram, 83127 – Lombok, Indonesia  
Email: giriwahyuwiriasto@unram.ac.id

### 1. PENDAHULUAN

Game merupakan permainan digital yang dimainkan melalui media visualisasi dan interaksi yang ditampilkan pada *hardware* seperti desktop dan *smartphone*. Game yang dimainkan oleh player dapat memiliki banyak kategori atau tipe [1]. Game *Role Playing Game* (RPG) merupakan salah satu dari tipe yang ada. Game RPG ini merupakan game dengan visual 2 dimensi (2D) dan perilaku dari karakter pada game berupa *platformer*, dimana merupakan tipe yang menggabungkan elemen peran dalam lingkungan visualisasi game 2D. Permainan game *platformer*, yang sering juga disebut sebagai permainan *jump and run*, merupakan salah satu variasi dari permainan aksi di mana pemain diharuskan mengendalikan karakter dari satu lokasi ke lokasi lain di dalam suatu arena. Permainan game *platformer* memiliki ciri khas dalam desainnya, yang melibatkan penggunaan medan dengan permukaan yang tidak merata dan berbagai *platform* dengan variasi ketinggian. Inti dari permainan ini adalah kemampuan karakter untuk melakukan lompatan atau pemanjatan guna mencapai titik lokasi tertentu serta meraih tujuan tertentu. Di dalam permainan ini, elemen-elemen

manuver akrobatik juga memainkan peran penting. Berbagai perilaku atau kemampuan seperti pemanjatan, ayunan dengan tali, lompatan antara dinding, melompat tinggi di udara, meluncur, dan bahkan memantul, semuanya menjadi faktor penting yang dimiliki oleh berbagai karakter dalam sub-tipe game ini. Selalu ada hal yang menarik dari setiap cerita suatu game yang dimainkan oleh pengguna, baik dari sisi skenario, karakter, dan sisi lainnya yang belum pernah terpikirkan sebelumnya [3].

Pada game jenis RPG ini dimasukkan aspek penceritaan yang kompleks serta karakter yang membuat seseorang merasa seperti menjadi tokoh yang diperankannya dalam game tersebut [4]. Tipe ini memberikan kebebasan dalam eksplorasi lingkungan yang variatif, menantang pemain dengan *puzzle* dan pertarungan yang menarik, serta memungkinkan peningkatan karakter dan penyesuaian yang lebih dalam. Seiring berkembangnya game sekarang telah muncul sistem operasi berbasis android diharapkan bisa meningkatkan minat para programmer untuk membuat aplikasi dengan *platform* independen. Tentunya hal ini akan memberi *opportunity* bagi para pengembang dan pengguna game memungkinkan dapat terlibat dalam membuat game berbasis android [2]. Tersedianya android sebagai salah satu sistem operasi yang banyak digunakan memberikan dampak perubahan bagi para pemain game [7][8][10] yakni dengan memanfaatkan ketersediaan pilihan aplikasi game. Agar pembuatan game dapat tercapai pada penelitian ini, telah digunakan IDE (*Integrated Development Environment*) yang dapat digunakan [9].

Dilakukan eksplorasi bagaimana memanfaatkan game sebagai media edukasi. Konten edukasi yang diangkat mengenai pencemaran lingkungan pada ekosistem biota laut. Media game sebagai sarana edukasi dan sosialisasi dapat menjadi alternatif. Dengan memainkan game ini player secara tidak langsung diberikan pemahaman tentang dampak dan akibat dari sampah merupakan satu langkah menuju perbaikan ekosistem biota laut. Misi yang terdapat pada game ini juga terkait dengan upaya melestarikan lingkungan untuk masa mendatang. Rotasi sampah akan terus berputar dan kembali terulang yang mengakibatkan kehancuran ekosistem biota laut, maka insiden ini tidak akan pernah terselesaikan [14]. Dalam perancangan game edukasi ini metode yang digunakan yaitu metode pengembangan *Extreme Programming* (XP) dan aplikasi dibangun berbasis android menggunakan Godot Engine [5].

## 2. METODOLOGI

### 2.1 Perancangan Skenario Game

Pada gameplay game *2D RPG platformer* ini dengan skenario yang telah dijelaskan sebelumnya, harapannya player atau pemain akan terlibat dalam permainan yang menarik. Selama permainan berjalan, pemain akan terus mengendalikan karakter mereka, melakukan input melalui tombol-tombol dan gerakan karakter. Setiap input dari pemain akan ditangani untuk merespons dengan tepat dalam permainan. Selanjutnya, game akan memperbarui status permainan secara terus-menerus. Hal ini termasuk memperbarui posisi karakter sesuai dengan input pemain, berinteraksi dengan objek-objek di lingkungan seperti mengumpulkan sampah atau mengalahkan musuh, serta mendeteksi tabrakan antara karakter pemain dengan musuh atau rintangan lainnya. Setiap perubahan dalam status permainan akan mempengaruhi tampilan grafis yang ditampilkan kepada pemain [13].

Dalam pengembangan game ini, ketika pemain bertabrakan dengan musuh, akan dipicu pertarungan *turn-based* yang akan memberikan pengalaman permainan yang berbeda. Namun, saat ini fitur pertarungan *turn-based* belum diterapkan sepenuhnya dalam permainan karena masih dalam tahap pengembangan. Dalam pertarungan *turn-based*, permainan akan beralih ke mode pertarungan di mana pemain dan musuh akan saling bergantian melakukan serangan dan bertahan. Setiap entitas akan memiliki statistik seperti *health point* (HP), serangan, pertahanan, dan kemampuan khusus yang dapat digunakan dalam pertarungan. Pengembang sedang memfokuskan upaya mereka untuk menyempurnakan *gameplay* dasar dan sistem interaksi sebelum memperkenalkan elemen pertarungan *turn-based* yang lebih kompleks.

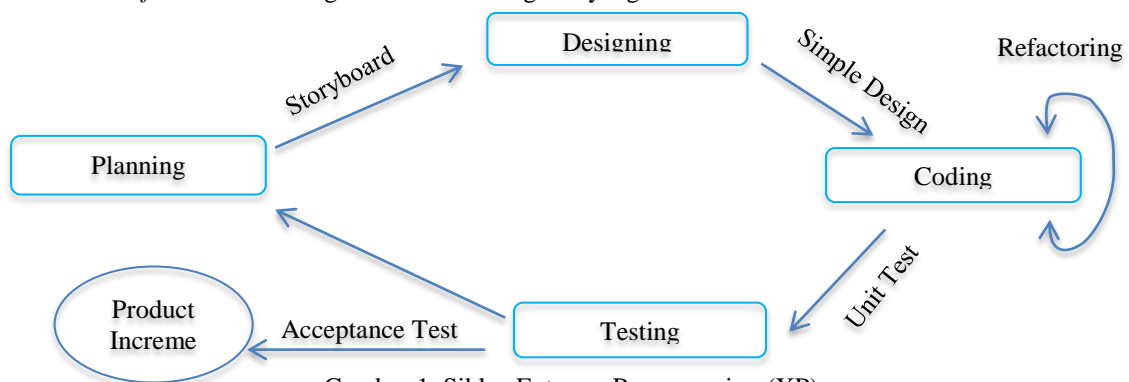
### 2.2. Game Engine

*Godot Engine* (GE) dipilih sebagai alat pengembangan game untuk proyek ini. GE merupakan mesin pengembangan game sumber terbuka yang kuat dan fleksibel. Dengan menggunakan GE, kita dapat mengakses berbagai fitur yang diperlukan untuk mengembangkan game 2D yang berkualitas, termasuk dukungan untuk fisika, animasi, *scripting* atau *godot script* (GS), dan manajemen sumber daya karena game didesain dan diciptakan berdasarkan acuan kehidupan sehari-hari [11] Manajemen berbasis hirarki *node* itu diletakkan dan ditaruh dalam scene yang lain, sehingga GE membantu pengembang bekerja lebih cepat, disamping editor dan interfacenya juga mudah dipahami dengan cepat oleh pengembang [15].

### 2.3 Metode Pengembangan Sistem

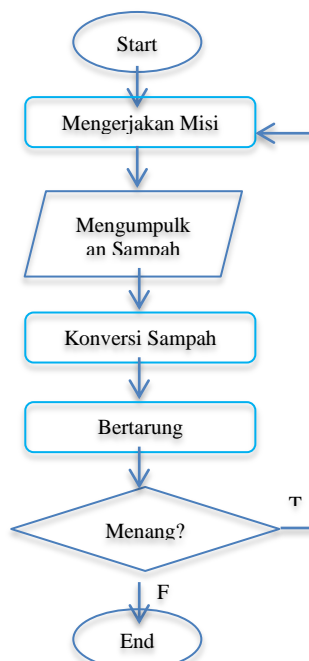
Berdasarkan proses pengembangan yang telah dilakukan, konsep pengembangan perangkat lunak dengan cara Agile adalah metode yang cocok dalam kasus ini. Salah satu metode yang digunakan pada pengembangan ini yaitu *extreme programming (XP)*. XP merupakan metode yang memberikan tahapan pengerjaan dalam waktu yang efektif, serta sesuai dengan fokus yang ingin dicapai oleh *developer* [12]. XP mendorong programmer untuk berani bereksperimen dan menulis ulang kode jika mereka tidak dengan kode atau desain yang sudah ada. Hal ini mempertahankan moral serta integritas para pengembang game dan dapat mendukung lebih lanjut komunikasi dengan anggota proyek lainnya. Metode yang digunakan pada XP memiliki empat tahapan dalam pelaksanaannya, yaitu tahap perencanaan, perancangan, pengkodean, dan pengujian. Terdapat 4 tahapan pada pengembangan metode *Extreme Programming (XP)* [5], yaitu:

1. *Planning*, pada tahap ini sebuah konsep game dipelajari, pengumpulan kebutuhan perancangan sistem, penentuan output dari program, fitur-fitur dari aplikasi dan fungsionalitas sebuah sistem yang akan dibangun. Berikut adalah *flowchart* rancangan awal skenario game yang akan dibuat:



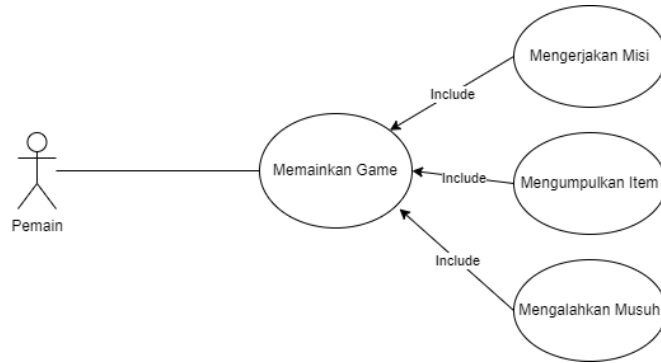
Gambar 1. Siklus Extreme Programming (XP)

Pada gambar 2 diatas merupakan skenario awal dari sebuah game yang akan dirancang. Dimana pemain dapat mengerjakan misi yang diberikan sembari mengumpulkan sampah yang tersebar dilautan. Jika sampah yang dikumpulkan dirasa sudah cukup, maka pemain dapat mengkonversi sampah tersebut menjadi koin yang dapat digunakan untuk membeli item di dalam game. Tidak hanya itu, pemain juga dapat bertarung dengan musuh yang berada di sekitar peta. Apabila pemain menang, maka pemain dapat melanjutkan misi, apabila kalah game akan berakhir.



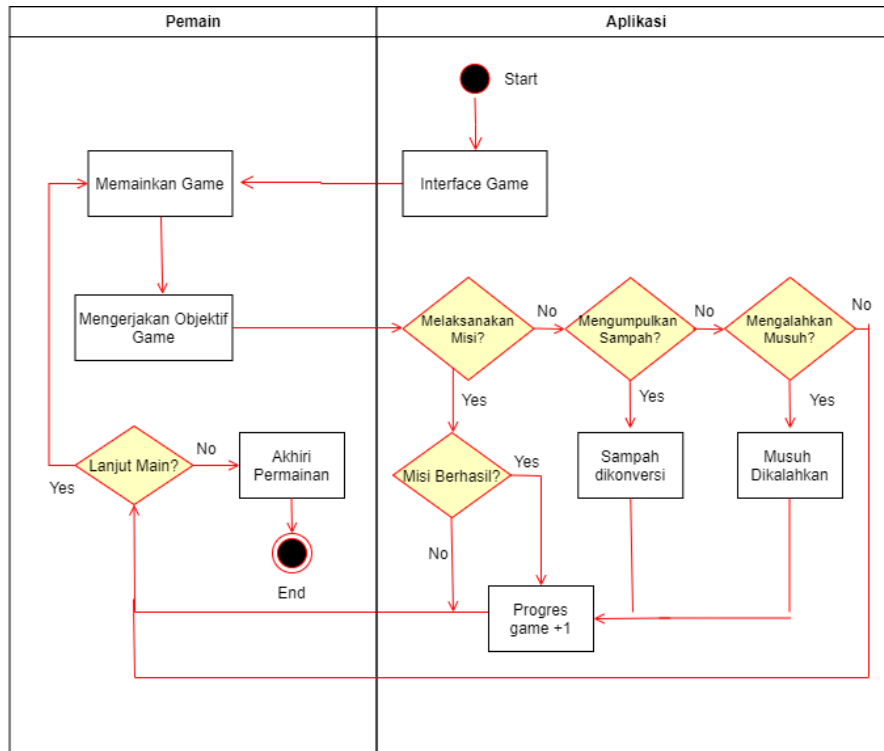
Gambar 2. Flowchart Rancangan Skenario Awal

2. Design Penggunaan model *Unified Modelling Language (UML)* digunakan dalam penelitian ini yaitu pada desain rancangan sistem. Model yang digunakan berupa use case diagram, sequence diagram dan activity diagram. Tahapan desain pada tahap ini digunakan sebagai *prototype* dari program yang akan dibangun sebagai sarana untuk mempermudah proses pengembangan. Berikut adalah *Use Case Diagram* pada rancangan game ini memberikan gambaran hubungan komunikasi antara pengguna dengan sistem itu sendiri, seperti yang dipaparkan pada Gambar 3.



Gambar 3. Use Case Diagram

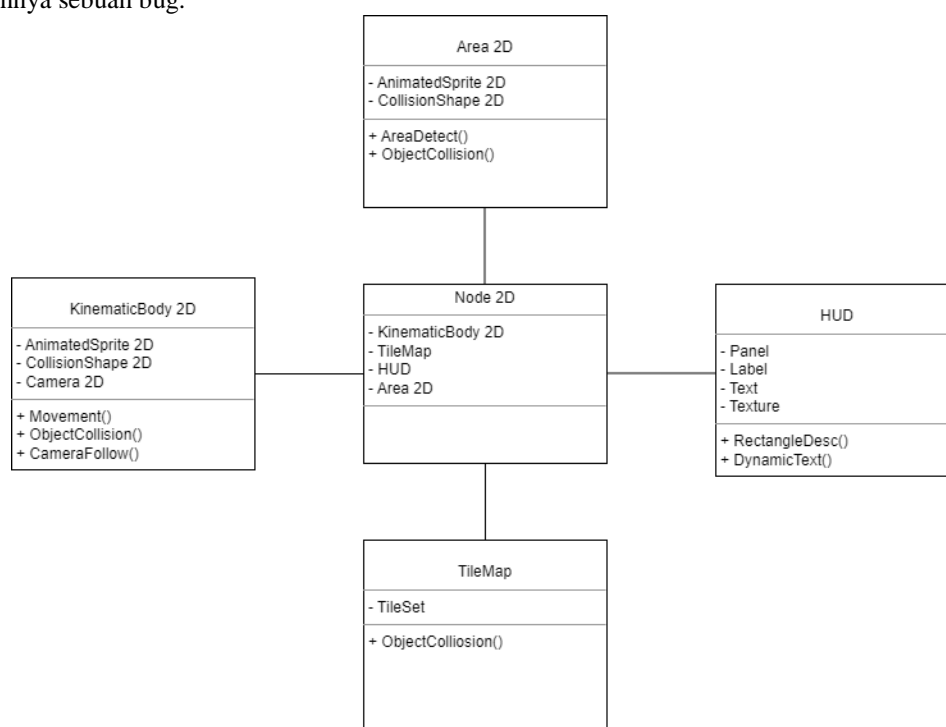
Pada gambar 4 digunakan *diagram activity* untuk menggambarkan alur aktivitas permainan game. Aktivitas player bermain tergambarkan dibagian kolom bagian kiri dan aplikasi game yang dimainkan pada kolom lainnya.



Gambar 4. Activity Diagram

Pada Gambar 5 merupakan *class diagram* dari aplikasi game yang dikembangkan. Pemrograman dengan *script* godot mengikuti rancangan *class diagram* ini dalam menggerakkan setiap karakter pada asset atau objek game.

3. Coding Sistem yang dijadikan sebagai objek penelitian dibangun pada tahap ini. Pada tahap coding disisipkan juga tahap refactoring, yaitu tahapan penyesuaian sistem software dasar pengembangan aplikasi untuk memperbaiki struktur sistem yang sedang dibangun tanpa mengubah eksternal kode sistem itu sendiri. Hal ini dilakukan untuk pengembangan perangkat lunak agar lebih cepat dan meminimalisir kemungkinan ditemukannya sebuah bug.



**Gambar 5.** Class Diagram

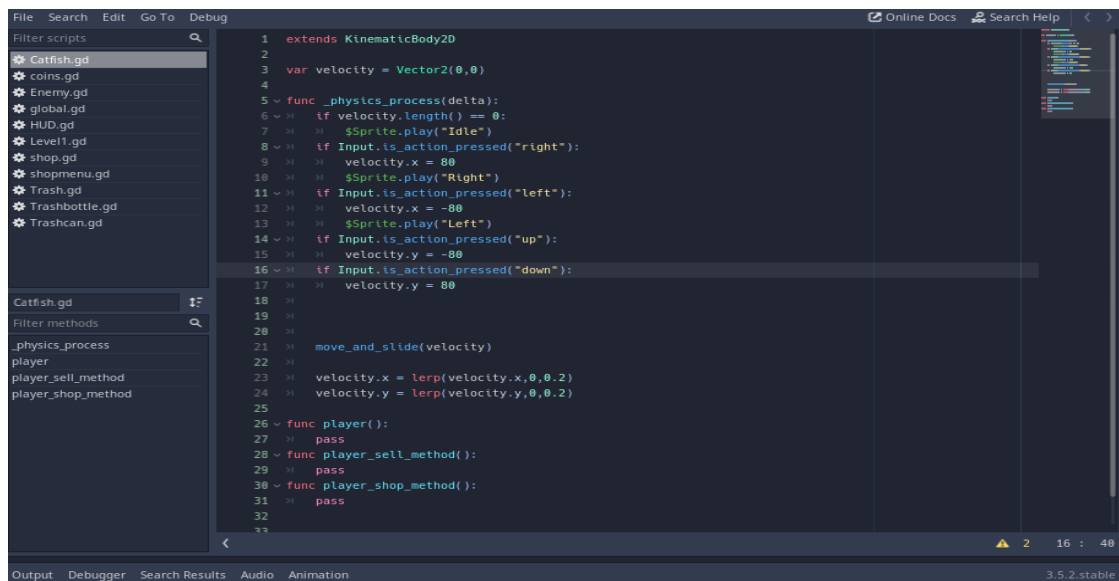
*Class diagram* pada gambar 5 menunjukkan hierarki kelas utama '*Node 2D*'. *Class* lainnya seperti '*KinematicBody 2D*' sebagai kelas untuk pengaturan asset objek. Kemudian *class* '*area 2D*' dan '*TileMap*' sebagai kelas untuk peta atau map area.

4. Testing Pada tahap akhir ini dilakukan dengan menggunakan beta testing atau pengujian kuesioner untuk mendapatkan *feedback*. Dalam tahap ini sistem yang telah dibangun diuji coba kepada beberapa responden untuk menentukan apakah kebutuhan awal sudah terpenuhi atau belum.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Kontrol Karakter dan Animasi

'*KinematicBody2D*' adalah *class* berupa salah satu tipe node dalam *GS* yang digunakan untuk mengontrol pergerakan objek dalam dunia game 2D. Sebagai contoh, dalam game *platformer*, kita dapat menggunakan '*KinematicBody2D*' untuk mengendalikan karakter pemain. Metode '*\_physics\_process(delta)*' adalah salah satu metode yang paling penting dalam pengembangan game ini. Metode ini bertanggung jawab untuk mengontrol karakter berdasarkan input pengguna dan memutar animasi yang sesuai pada '*sprite*' karakter. Dalam metode ini, penggunaan '*velocity*' memungkinkan karakter untuk bergerak ke kanan, kiri, atas, dan bawah dengan kecepatan yang ditentukan.

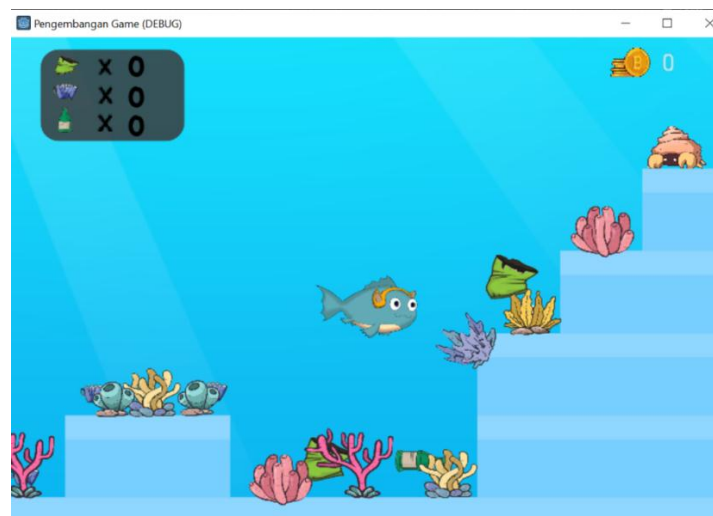


Gambar 6. Script Editor Godot

```

extends KinematicBody2D
var velocity = Vector2(0,0)
func _physics_process(delta):
if velocity.length() == 0:
    $Sprite.play("Idle")
if Input.is_action_pressed("right"):
    velocity.x = 80
    $Sprite.play("Right")
if Input.is_action_pressed("left"):
    velocity.x = -80
    $Sprite.play("Left")
if Input.is_action_pressed("up"):
    velocity.y = -80
if Input.is_action_pressed("down"):
    velocity.y = 80
move_and_slide(velocity)
velocity.x = lerp(velocity.x,0,0.2)
velocity.y = lerp(velocity.y,0,0.2)

```



Gambar 7. Tampilan Game Ketika Dijalankan

Pemutaran animasi sprite menggunakan metode '\$Sprite.play()' dengan parameter animasi yang sesuai, seperti 'Idle', 'Right', dan 'Left'. Selain itu, metode ini juga menggunakan 'move\_and\_slide(velocity)' untuk memindahkan karakter berdasarkan vektor kecepatan. Penggunaan 'lerp' digunakan untuk mengurangi kecepatan karakter secara perlahan hingga mencapai kecepatan bernilai '0' ketika tidak ada input dari pengguna.

#### a. Pengumpulan Sampah

'Area2D' adalah class sebuah node dalam GS yang digunakan untuk mendeteksi dan mengatur interaksi antara objek dalam lingkungan game 2D. Metode 'add\_Trash()', 'add\_Trashcan()', dan 'add\_Trashglass()' digunakan untuk menghitung jumlah sampah yang dikumpulkan oleh karakter.

```
extends Area2D

signal plastic_collected
func _on_Trash_body_entered(body):
    $AnimationPlayer.play("bounce")
    emit_signal("plastic_collected")
func _on_AnimationPlayer_animation_finished(anim_name):
    queue_free()
```

Setiap kali metode ini dipanggil, variabel yang sesuai seperti metode 'Trash', 'Trashcan', dan 'Trashglass' tadi akan bertambah '1'. Selain itu, metode ini juga mencetak jumlah sampah yang terkumpul ke konsol menggunakan perintah 'print()'. Metode ini berguna dalam menggambarkan kemajuan karakter dalam mengumpulkan sampah dan dapat digunakan dalam mekanik *gameplay* yang melibatkan sistem inventaris atau peningkatan karakter.

#### b. Pembaruan Jumlah Sampah

```
func _ready():
    $Pcount.text = String(plastic)
    $Ccount.text = String(can)
    $Bcount.text = String(bottle)
    $Coinstext.text = String(coins)
```



Gambar 8. HUD (Head Up Display)

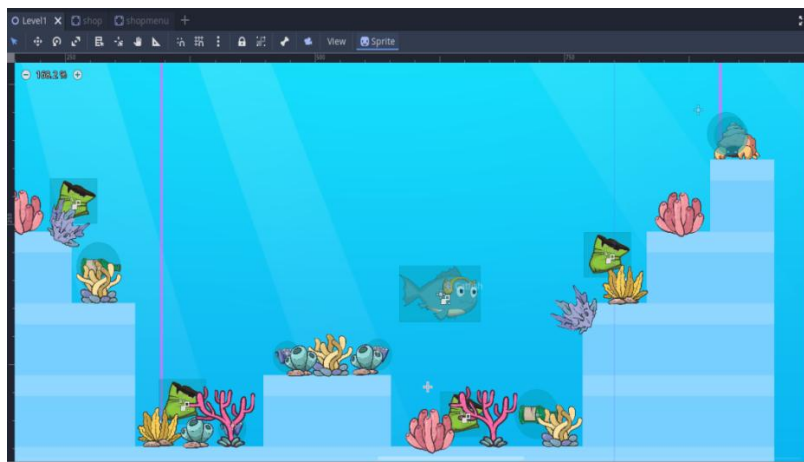
Metode '\_ready()' merupakan metode yang dipanggil saat objek siap digunakan. Dalam metode ini, label-label 'Pcount', 'Ccount', dan 'Bcount' yang mewakili jumlah sampah plastik, kaleng, dan kaca ditetapkan dengan nilai awal variabel 'plastic', 'can', dan 'bottle'. Metode ini memastikan bahwa tampilan jumlah sampah pada antarmuka pengguna diperbarui secara otomatis ketika terjadi perubahan dalam pengumpulan sampah.

#### c. Kollision Objek Sampah

Berikut ini algoritma dari proses pengumpulan sampah agar dapat diberikan nilai. Beberapa metode, seperti '\_on\_Trash\_plastic\_collected()', '\_on\_Trashcan\_can\_collected()', dan '\_on\_Trashbottle\_bottle\_collected()', adalah metode penanganan *kollision* objek sampah yang dikumpulkan oleh karakter. Metode ini dipanggil saat karakter bertabrakan dengan objek sampah yang sesuai, dan mereka menghasilkan sinyal 'plastic\_collected', 'can\_collected', dan 'bottle\_collected'.

```
func _on_Trash_plastic_collected():  
plastic = plastic + 1  
_ready()  
func _on_Trash5_plastic_collected():  
plastic = plastic + 1  
_ready()  
  
func _on_Trash2_plastic_collected():  
plastic = plastic + 1  
_ready()  
  
func _on_Trash3_plastic_collected():  
plastic = plastic + 1  
_ready()  
  
func _on_Trash4_plastic_collected():  
plastic = plastic + 1  
_ready()  
  
func _on_Trashcan_can_collected():  
can = can + 1  
_ready()  
  
func _on_Trashcan2_can_collected():  
can = can + 1  
_ready()  
  
func _on_Trashcan3_can_collected():  
can = can + 1  
_ready()  
func _on_Trashbottle_bottle_collected():  
bottle = bottle + 1  
_ready()  
  
func _on_Trashbottle2_bottle_collected():  
bottle = bottle + 1  
_ready()  
func _on_coins_coins_collected():  
coins = coins + 1  
ready()
```

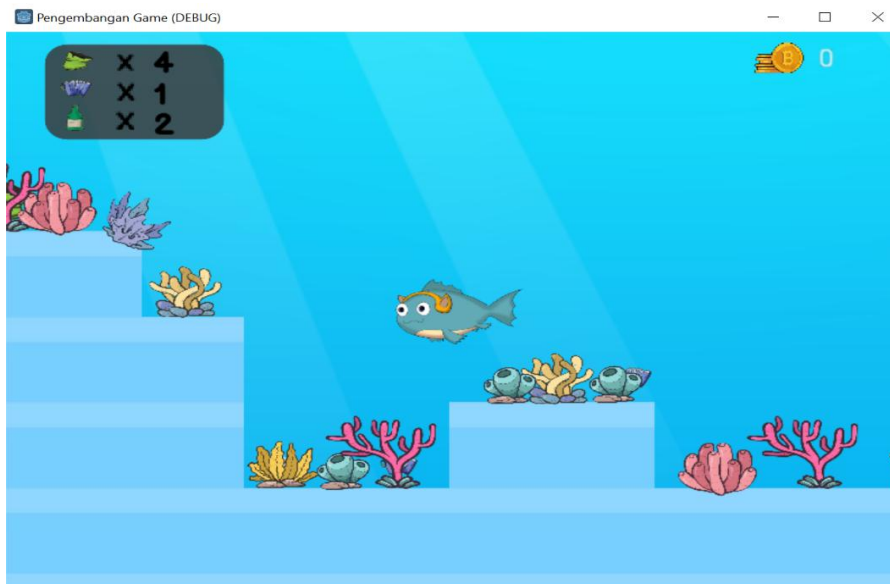
Setiap metode ini bertanggung jawab untuk menambah jumlah sampah yang sesuai (seperti *'plastic'*, *'can'*, dan *'bottle'*) dan memperbarui tampilan jumlah sampah pada antarmuka pengguna melalui pemanggilan metode *'\_ready()'*.



**Gambar 9.** Ketika Pemain Memasuki Area Tertentu



Area bayangan berwarna hijau yang bersampingan dengan asset objek merupakan area *collision* atau area tumbukan, dimana suatu objek akan dapat bertumbukan apabila telah memasuki area *collision* dari masing masing objek. Dalam hal ini apabila karakter bertumbukan dengan sampah maka sampah akan secara otomatis terkumpulkan atau ter-*collected* dan sampah yang dikumpulkan akan mempengaruhi nilai *scoreboard*.



Gambar 10. Tampilan Gameplay

Pada gambar 10 dapat dilihat bahwa gambar tersebut merupakan tampilan *gameplay*. Tahap *gameplay* ini merupakan proses roaming atau hanya berkeliling untuk mengumpulkan sampah. Pada antarmuka *gameplay* terdapat 2 buah panel yang menampilkan berapa jumlah sampah yang telah dikumpulkan berdasarkan jenisnya dan panel kedua menampilkan berapa jumlah koin yang telah didapatkan.

#### d. Konversi Sampah

```
extends StaticBody2D

var item = 1
var item1price = 1
var item2price = 1
var item3price = 1
var coins = 1
var item1owned = false
var item2owned = false
var item3owned = false
var price

func _ready():
    $icon.play("plastic")
    item = 1

func _physics_process(delta):
    if self.visible == true:
        if item == 1:
            $icon.play("plastic")
        if item == 2:
            $icon.play("can")
        if item == 3:
            $icon.play("bottle")

func _on_buybutton_pressed():
```

```

if item == 1:
    price = item1price
    if coins >= price:
        if item1owned == false:
            buy()
elif item == 2:
    price = item2price

func _on_buttonright2_pressed():
    swap_item_forward()

func _on_buttonleft_pressed():
    swap_item_back()

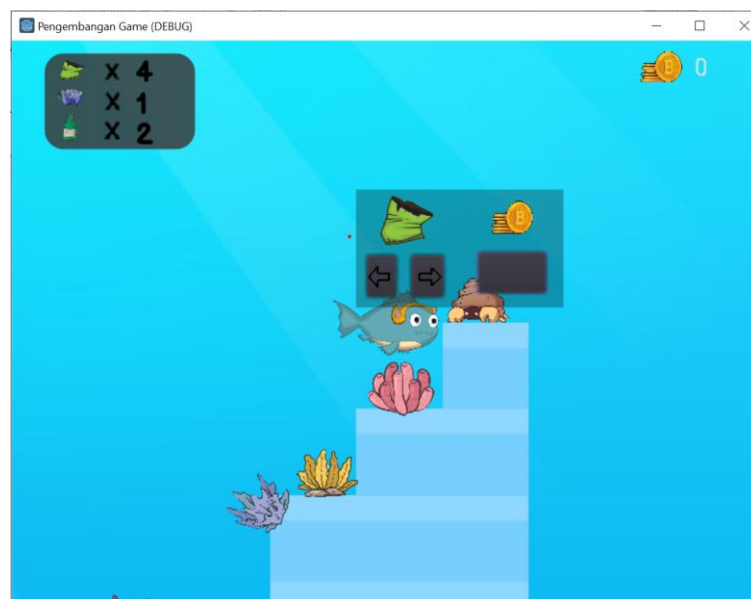
func swap_item_back():
    if item == 1:
        item = 2
    elif item == 2:
        item = 3
    elif item == 3:
        item = 1

func swap_item_forward():
    if item == 1:
        item = 2
    elif item == 2:
        item = 3
    elif item == 3:
        item = 1

func buy():
    coins -= price
    if item == 1:
        item1owned = true
    if item == 2:
        item2owned = true

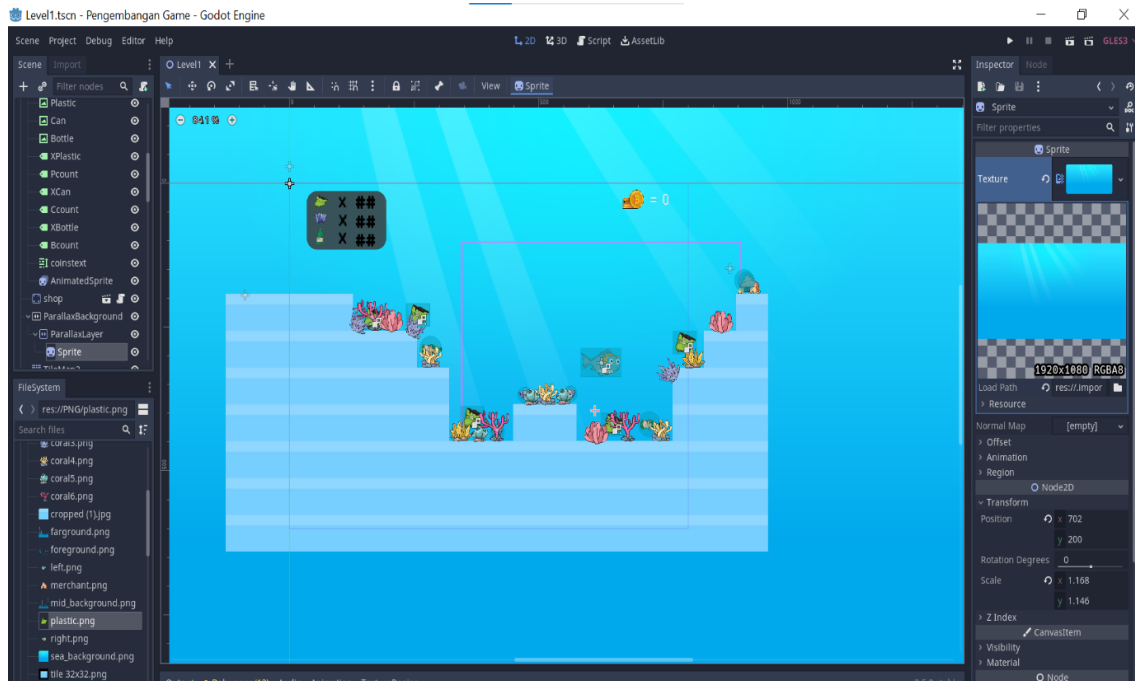
```

Pada gambar 11 dapat dilihat bahwa gambar tersebut merupakan tampilan fitur konversi dalam permainan. Pada fitur konversi ini, kita dapat melakukan penukaran sampah menjadi koin, dimana jumlah koin yang didapatkan akan berbeda tergantung jenis sampahnya.



**Gambar 11.** Menu Konversi

Menu konversi dapat terbuka ketika pemain mengarahkan karakter mendekati suatu objek tempat konversi, ketika masuk dalam radius tertentu, maka menu konversi akan muncul, menu tersebut akan menghilang apabila player menjauh dari area objek tersebut.



Gambar 12. Antarmuka Godot IDE

Pada gambar 12 merupakan interface dari *GE* yang digunakan untuk membangun game secara keseluruhan menggunakan node dan scene yang saling terhubung satu sama lain yang dapat dimanipulasi dengan menggunakan fitur *inspector*, tidak hanya itu, kita juga dapat melakukan *import asset* dengan cara *drag and place*.

#### 4. KESIMPULAN

Permainan game *save our sea* sebagai *simulasi edukasi dalam menjaga lingkungan laut* telah berhasil dikembangkan. Skenario dan *gameplay* memberikan gambaran tentang potensi dan tujuan akhir dari permainan dimana game menghadirkan perjalanan petualangan dengan misi yang disediakan secara visual. Pemain akan menghadapi rintangan *platforming*, teka-teki lingkungan, serta musuh *NPC* player yang harus dikalahkan. Dalam upaya untuk menyelamatkan lingkungan, pemain akan mengumpulkan sampah-sampah yang ada di sepanjang perjalanan. Terdapat fitur pertarungan *turn-based* yang juga akan diimplementasikan dalam permainan agar menambah interaksi dengan player. Parameter perolehan perhitungan atau *scoring* pada player telah dapat bekerja mengikuti bobot objek yang diberikan pada setiap objek *asset* dalam game.

#### DAFTAR PUSTAKA

- [1] A. F. Naharu, E. M. A. Jonomaro dan M. A. Akbar, "Penerapan Hierarchical Finite State Machine untuk Pengambilan Keputusan Non-Player Character (Studi Kasus: Gim Hack and Slash)", Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X Vol. 5, No. 3, hlm. 1136-1141, Maret 2021.
- [2] C. A. R. Tobi, R. Andrea, and A. Yusika, "Membangun Side Scrolling Game Flying Enggang Berbasis Android Dengan Godot Engine," J. Inform. Wicida, vol. 8, no. 1, pp. 1–8, 2019.
- [3] M. Fikriansyah, G. W. Wiriasto, A.S. Rachman, "Rancang Bangun Prilaku Buatan pada Non-player Character dalam Game Pemadam Kebakaran Menggunakan Finite State Machine dan Godot Script, Jurnal Dielektrika, v. 10, n. 1, p. 1- , ISSN 2579-650X 13, Feb. 2023.
- [4] M. Fikriansyah, G. W. Wiriasto and A.S. Rachman, "Non-player Character in Fire Fighter Games Using Geneti Algorithm", Jurnal Mnemonic, 6(1), pp. 11-19, 2023.

- [5] W. Haryono., dan A. Syafii, “Penerapan Extreme Programming pada Pengembangan Game Edukasi Asmaul Husna , Sifat Allah dan Nama Nabi Menggunakan Aplikasi Construct 2”, Journal of Artificial Intelligence and Innovative Applications Vol. 3, No. 1, 2022.
- [6] M. F. Rahadian, A. Suyatno, and S. Maharani, “Penerapan Metode Finite State Machine Pada Game ‘The Relationship,’” Inform. Mulawarman J. Ilm. Ilmu Komput., vol. 11, no. 1, p. 14, 2016.
- [7] A.B. Pohan , “Pengembangan Idle Game “Havok Runner” Berbasis Android Menggunakan Metode Agile Game Development” J. Media Inform. Budidarma, vol. 6, no. 3, pp. 91–98, 2023.
- [8] R. Hidayatullah and A. Hadiansa, “Perancangan Permainan Ular Tangga Multiplayer Berbasis Android,” J. Media Inform. Budidarma, vol. 2, no. 3, pp. 91–98, 2018.
- [9] I. I. Shagianto, G.W. Wiriastoro, D. F. Budiman, and Misbahuddin “Aplikasi Game Berbasis Android 2D dengan Logika Fuzzy pada NPC (Non - Playable Character)”, 2023.
- [10] W.N. Cholifah, Y. Yulianingsih, and S. M. Sagita. “Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap,” STRING (Satuan Tulisan Ris. dan Inov. Teknol., vol. 3, no. 2, p. 206, CAN - J. Teknol. Inf. dan Komun., vol. 11, no. 2, JITTER- Jurnal Ilmiah Teknologi dan Komputer Vol. 2, No. 3, 2018.
- [11] W. Safitra, A. Faisol dan S. A. Wibowo, “Finite State Machine Method to Non-Player Character (NPC) Action Strategy Game 'Ouroboros’”, Vol. 4 No. 2, JATI Vol. 4 No. 2, 2020
- [12] A. Trisnadoli, “Implementasi Extreme Programming (XP) Agile Software Development pada Pengembangan Sistem Informasi KELUARGAKU” Vol. 6, No. 2, Juni 2021.
- [13] K. Asgaryansyah, “Save Our Sea: Rancangan Game 2D RPG Berbasis Android Menggunakan Godot Engine “, <https://github.com/KevinAsgaryansyah/KevinAsgaryansyah>
- [14] O. Adityawan, dan U. U. T. Saarahdiba, “Perancangan Game Edukasi Penyelamatan Biota Laut “, Vol. 4, no. 1, 2017.
- [15] A. S. Arnomo, “Perancangan Game Platformer Pemburu Koin Menggunakan Godot Engine”, Jurnal COMASIE - Vol. 06 No. 04, 2022.

## BIOGRAPHY OF AUTHORS



Kevin Asgaryansyah is an indie game developer with a strong passion for game programming. He is currently in the process of completing his Bachelor's degree in Electrical Engineering at the University of Mataram.

[kevinfaizkf@gmail.com](mailto:kevinfaizkf@gmail.com)



Giri Wahyu Wiriasto completed Bachelor degree program in Electrical Engineering Department at Institut Teknologi Nasional of Malang, 2006. Completing a Master's Degree Program in Electrical Engineering and Multimedia Smart Computer Network System at Institut Teknologi Sepuluh Nopember of Surabaya, 2010. Since 2010, he has been serving as a lecturer at the Electrical and Computer Engineering Department of the University of Mataram. He has a strong interest in reading and specializes in teaching various courses, including Serious Game, Deep Learning, and Digital Technology Transform.

[giriwahyuwiriasto@unram.ac.id](mailto:giriwahyuwiriasto@unram.ac.id)