

LAMPIRAN

Lampiran 1 : Lembar Permohonan dan Persetujuan Pembimbing 1



PT BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III.INF/TA/2024
Lampiran : ---
Perihal : Pembimbing Utama Skripsi

Kepada : **Yth. Bpk/Ibu Renaldi Primaswara Prasetya S.Kom., M.Kom.**
Dosen Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang


Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Prita Patricia Lakzmi
Nim : 2118126
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

21 Agustus 2024 s/d 21 Pebruari 2025

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih.


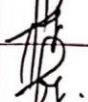

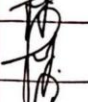


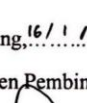

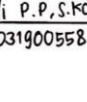

Mengetahui
Program Studi Teknik Informatika S-1
Ketua

Yosep Agus Pranoto, ST., MT.
NIP.P. 1031000432

Form S-4a

Lampiran 2 : Formulir Bimbingan Skripsi 1

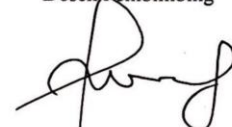
FORMULIR BIMBINGAN SKRIPSI

Nama : Prita Patricia Larzmi
 Nim : 2118126
 Masa Bimbingan : 21 Agustus 2024 - 21 Februari 2024
 Judul Skripsi : Rancang Bangun Game Edukasi Berbasis Android 2D "Nawa Sanga"
 Dengan Penerapan Multimedia Development Life Cycle (MDLC)

No.	Tanggal	Uraian	Paraf Pembimbing
1.	15 Oktober 2024	-Melanjutkan dan menyelesaikan pembuatan aset	
2.	24 Oktober 2024	Melanjutkan Animasi, dan Aset yang lain.	
3.	31 Oktober 2024	ACC Seminar Progress	
4.	21 November 2024	Revisi laporan bagian tujuan	
5.	28 November 2024	Pengujian	
6.	3 Desember 2024	Pengujian dan jurnal	
7.	5 Desember 2024	ACC Semhar	
8.	9 Januari 2025	Revisi pengujian FSM	
9.	13 Januari 2025	Revisi laporan diagram FSM	
10.	16 Januari 2025	ACC	

Malang, 16/1/2024

Dosen Pembimbing



(Renaldi P.P., S.Kom., M.Kom)
 NIP. 1031900558

Lampiran 3 : Lembar Permohonan dan Persetujuan Pembimbing 2



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 04 Oktober 2024

Nomor : ITN-883/III.INF/TA/2024
Lampiran : ---
Perihal : Pembimbing Pendamping Skripsi

Kepada : **Yth. Bpk/Ibu Febriana Santi Wahyuni S.Kom., M.Kom.**
Dosen Pembina Program Studi Teknik Informatika S-1
Institut Teknologi Nasional
Malang

Dengan Hormat,
Sesuai dengan permohonan dan persetujuan dalam proposal skripsi untuk mahasiswa :

Nama : Prita Patricia Lakzmi
Nim : 2118126
Prodi : Teknik Informatika S-1
Fakultas : Teknologi Industri

Maka dengan ini pembimbingan kami serahkan sepenuhnya kepada Saudara/i selama waktu 6 (enam) bulan, terhitung mulai tanggal :

21 Agustus 2024 s/d 21 Pebruari 2025

Sebagai satu syarat untuk menempuh Ujian Akhir Sarjana Teknik, Program Studi Teknik Informatika S-1.
Demikian agar mahlum dan atas perhatian serta bantuannya kami sampaikan terima kasih.

Mengetahui
Program Studi Teknik Informatika S-1
Ketua,

Yosep Agus Pranoto, ST., MT.
NIP.P. 1031000432

Form S-4a

Lampiran 4 : Formulir Bimbingan Skripsi 2

FORMULIR BIMBINGAN SKRIPSI

Nama : Prita Patricia Lakzmi
 Nim : 2118126
 Masa Bimbingan : 21 Agustus 2024 - 21 Februari 2024
 Judul Skripsi : Rancang Bangun Game Edukasi Berbasis Android 2D "Nawa Sarga"
 Dengan Penerapan Multimedia Development Life Cycle (MDLC)

No.	Tanggal	Uraian	Paraf Pembimbing
1.	17/10/24	Langgulkan desain aset, karakter,	<u>hm</u>
2.	24/10/24	Background, map, enemy (2), tile,	<u>hm</u>
3.	31/10/24	ACC Seminar Progress	<u>hm</u>
4.	7/11/24	Revisi Laporan. Lanjutkan program	<u>hm</u>
5.	21/11/24	level 1, tambahkan voice over	<u>hm</u>
6.	3/12/24	revisi level game + Laporan	<u>hm</u>
7.	5/12/24	ACC Semhas.	<u>hm</u>
8.	9/1/25	Revisi Laporan	<u>hm</u>
9.	13/1/25	Revisi Laporan	<u>hm</u>
10.	15/1/25	ACC Uti Komprehensif	<u>hm</u>

Malang, 15.1.2024

Dosen Pembimbing

(Febriana S. W. S. Kom, M. Kom.)
 NIP. 1031000425

Lampiran 5 : Berita Acara



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

Nama : Prita Patricia Lakzmi
Nim : 2118126
Jurusan : Teknik Informatika S-1
Judul : Rancang Bangun Game Edukasi Berbasis Android 2d "Nawa Sanga" dengan Penerapan Multimedia Development Life Cycle (MDLC)

Dipertahankan Dihadapan Majelis Penguji Skripsi Jenjang Strata Satu(S-1)
Pada

Hari : Rabu
Tanggal : 22 Januari 2025
Nilai : 84 (A)

Panitia Ujian Skripsi :
Ketua Majelis Penguji

Yosep Agus Pranoto ST., MT
NIP.P 1031000432

Anggota Penguji :

Dosen Penguji I

Dosen Penguji II

Dr. A Fahrudi Setiawan, S.Kom., MT
NIP.P 1031500497

Joseph Dedy Irawan, ST., MT
NIP.197404162005011002

Lampiran 6 : Form Perbaikan



PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Prita Patricia Lakzmi
NIM : 2118126
JURUSAN : Teknik Informatika S-1
JUDUL : Rancang Bangun Game Edukasi Berbasis Android 2d
"Nawa Sanga" dengan Penerapan Multimedia
Development Life Cycle (MDLC)

No.	Penguji	Tanggal	Uraian	Paraf
1.	Penguji I	22 Januari 2025	1. Kalau membuat produk yang zero cror. 2. Kalau membuat edukasi, yang membuat sudah pernah kesana. 3. Tingkatkan hasil pekerjaan 100% Toto, titi, tutuq, tuntas, tataq.	
2.	Penguji II	22 Januari 2025	1. Pertanyaan dari pengujian. 2. Tujuan harus spesifik. 3. Kesimpulan harus menjawab tujuan. 4. Daftar isi. 5. Daftar pustaka. 6. Semua gambar diberi penjelasan. 7. Jelaskan secara detail dari setiap karakter.	

Dosen Penguji I

Dr. A Fahrudi Setiawan, S.Kom., MT
NIP.P 1031500497

Dosen Pembimbing I

Renaldi Primaswara P, S.Kom., M.Kom
NIP.P 1031900558

Dosen Penguji II

Joseph Dedy Irawan, ST., MT
NIP.197404162005011002

Dosen Pembimbing II

Febriana Santi W, S.Kom., M.Kom
NIP.P 1031000425

Lampiran 7 : Hasil persentase cek plagiasi Turnitin



INSTITUT TEKNOLOGI NASIONAL MALANG
PERPUSTAKAAN PUSAT
Jln. Bendungan Sigura-gura No 2 Malang 65145
Telp. (0341) 551431 Pcs. 163-146-147 Fax. (0341) 553015 Website : library.itn.ac.id

FORM UJI PLAGIASI UNTUK MAHASISWA

Yang bertandatangan di bawah ini, Mahasiswa Institut Teknologi Nasional Malang:

Nama : Prita Patricia Lazmi
NIM : 208126
Fakultas / Jurusan : Fakultas Teknologi Industri / Teknik Informatika 1-1
Email : pritaakzmi@gmail.com
No. Tlp : 081230660937
Judul/ Jml artikel : Rancang Bangun Game Edukasi Berbasis Android 2D "Nawa Sanga"
dengan Penerapan Multimedia Development Life Cycle (MDLC).

Karya ilmiah yang bersangkutan di atas melalui proses cek plagiasi menggunakan aplikasi trunitin dengan hasil kemiripan (Similarity) Sebesar...13..... %

Demikian surat keterangan ini dibuat agar dapat dipergunakan sebagaimana mestinya.

Malang, 07-02-2021

Mahasiswa

Prita Patricia L

Pelaksana

Prita Patricia L

Lampiran 8 : Hasil persentase cek plagiasi Turnitin

RANCANG BANGUN GAME EDUKASI BERBASIS ANDROID 2D
"NAWA SANGA" DENGAN PENERAPAN MULTIMEDIA
DEVELOPMENT LIFE CYCLE (MDLC)

ORIGINALITY REPORT



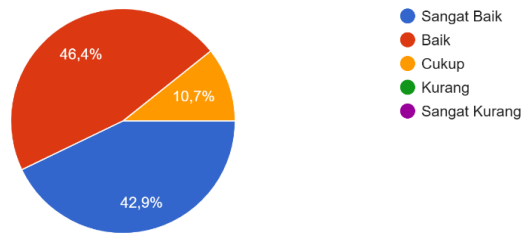
Lampiran 9 : Dokumentasi device dan pengujian pengguna



Lampiran 10: Hasil pertanyaan yang diajukan kepada pengguna

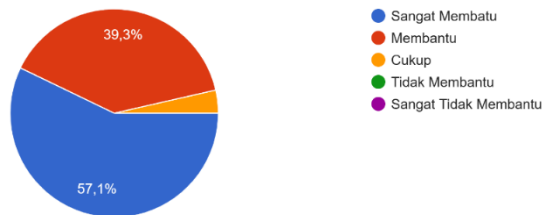
Bagaimana pendapat anda tentang cara game ini menyampaikan nilai-nilai sejarah?

28 jawaban



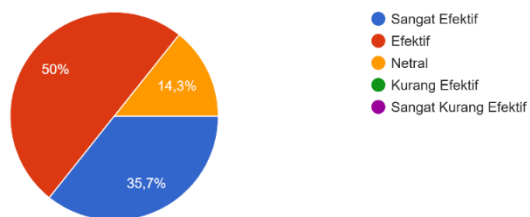
Apakah anda merasa game ini membantu anda belajar tentang tempat-tempat bersejarah yang ada di Indonesia?

28 jawaban



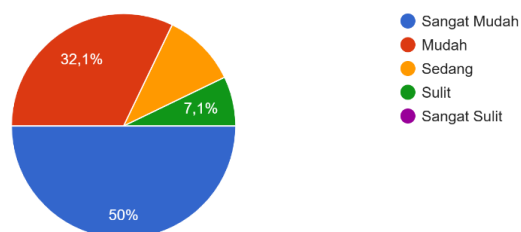
Apakah penggabungan antara elemen sejarah dengan hiburan dalam game ini efektif untuk membantu pemahaman tentang tempat-tempat bersejarah?

28 jawaban



Bagaimana pengalaman anda saat menggunakan kontrol dalam game?

28 jawaban



Lampiran 11 : Source code *Player*

```
using System.Collections;
using UnityEngine;

public class Player : MonoBehaviour
{
    public Animator animator;
    public GameObject bullet; // Bullet untuk menyerang
    public Transform firePoint; // Titik keluarnya peluru
    public CoinManager coinManager; // Pastikan CoinManager sudah
    didefinisikan
    private Rigidbody2D rb;
    private bool isGrounded;
    private bool jump;
    private bool facingRight = true;
    private bool isAttacking = false;

    [SerializeField] private Transform groundCheckCollider;
    [SerializeField] private LayerMask groundLayer;
    [SerializeField] private float speed = 10f;
    [SerializeField] private float jumpPower = 5f;
    [SerializeField] private float gravityScale = 3f;
    private const float groundCheckRadius = 0.2f;
    private float horizontalValue; // Ini akan diubah sesuai input
    dari tombol UI

    // Variabel health
    public int playerHealth = 5;
    public GameObject[] healthUnits; // Array untuk UI Health
    private PlayerHealth playerHealthScript; // Referensi ke
    PlayerHealth
    private Vector2 lastSafePosition; // Posisi terakhir yang aman
    sebelum jatuh
    [SerializeField] private float respawnOffsetX = 3f; // Offset
    horizontal untuk respawn (3 unit ke depan)

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
        animator = GetComponent<Animator>();
        rb.gravityScale = gravityScale;
        lastSafePosition = transform.position; // Set posisi
        terakhir yang aman
        playerHealthScript = GetComponent<PlayerHealth>(); //
        Mendapatkan referensi ke PlayerHealth
        coinManager = CoinManager.instance; // Jika CoinManager
        adalah singleton
    }

    private void Update()
    {
        // Tombol C digunakan untuk menyerang
        if (Input.GetKeyDown(KeyCode.C) && !isAttacking)
```

```

    {
        TriggerAttack(); // Jalankan serangan
    }

    // Cek jika player jatuh di luar ground
    if (transform.position.y < -10f)
    {
        ReduceHealth(); // Kurangi health saat jatuh
        // Reset posisi pemain ke posisi terakhir yang aman
        dengan offset
        transform.position = new Vector2(lastSafePosition.x +
respawnOffsetX, lastSafePosition.y);
        rb.velocity = Vector2.zero; // Reset kecepatan
    }
}

private void FixedUpdate()
{
    GroundCheck();
    Move(horizontalValue, jump); // Gerakan berdasarkan
horizontalValue
    animator.SetFloat("Blend", Mathf.Abs(rb.velocity.x));
    animator.SetFloat("Blend Jump", rb.velocity.y);

    if (jump)
    {
        jump = false;
    }

    // Update posisi terakhir yang aman jika pemain berada di
tanah
    if (isGrounded)
    {
        lastSafePosition = transform.position; // Simpan
posisi terakhir yang aman
    }
}

private void Move(float direction, bool jump)
{
    rb.velocity = new Vector2(direction * speed *
Time.fixedDeltaTime, rb.velocity.y);

    if (jump && isGrounded)
    {
        rb.velocity = new Vector2(rb.velocity.x, 0); //
Resetkan kecepatan Y
        rb.AddForce(Vector2.up * jumpPower,
ForceMode2D.Impulse);
        isGrounded = false;
    }

    if (direction > 0 && !facingRight)

```

```

        {
            Flip();
        }
        else if (direction < 0 && facingRight)
        {
            Flip();
        }
    }

    private void GroundCheck()
    {
        isGrounded = false;
        Collider2D[] colliders =
Physics2D.OverlapCircleAll(groundCheckCollider.position,
groundCheckRadius, groundLayer);
        if (colliders.Length > 0)
        {
            isGrounded = true;
        }
        animator.SetBool("Jumping", !isGrounded);
    }

    private void Flip()
    {
        facingRight = !facingRight;
        Vector3 scaler = transform.localScale;
        scaler.x *= -1;
        transform.localScale = scaler;
    }

    // Fungsi untuk mengurangi health
    public void ReduceHealth()
    {
        if (playerHealth > 0)
        {
            playerHealth--;
            playerHealthScript.TakeDamage(1); // Mengurangi health
di PlayerHealth
        }
        if (playerHealth <= 0)
        {
            // Player mati
            Debug.Log("Player mati!");
            // Tambahkan logika untuk respawn atau game over
        }
    }

    // Fungsi untuk bergerak ke kiri
    public void MoveLeft()
    {
        horizontalValue = -1f; // Gerak ke kiri
    }

    // Fungsi untuk bergerak ke kanan

```



```

public void MoveRight()
{
    horizontalValue = 1f; // Gerak ke kanan
}
// Fungsi untuk menghentikan pergerakan horizontal
public void StopMovement()
{
    horizontalValue = 0f; // Berhenti bergerak
    rb.velocity = new Vector2(0, rb.velocity.y); // Pastikan
kecepatan horizontal berhenti
}
// Fungsi untuk menangani tombol serang yang dipanggil melalui
UI Button
public void TriggerAttack()
{
    if (!isAttacking) // Pastikan tidak ada serangan bertumpuk
    {
        isAttacking = true;
        animator.SetTrigger("Attack"); // Menjalankan animasi
serangan

        float direction = facingRight ? 1f : -1f; // Tentukan
arah serangan
        GameObject bluefire = Instantiate(bullet,
firePoint.position, Quaternion.identity); // Membuat peluru

        // Meningkatkan kecepatan peluru untuk keluar lebih
cepat
        bluefire.GetComponent<Rigidbody2D>().velocity = new
Vector2(direction * 20f, 0); // Kecepatan peluru

        Destroy(bluefire, 2f); // Hancurkan peluru setelah
2 detik

        isAttacking = false; // Serangan selesai
    }
}
// Fungsi yang dipanggil ketika tombol kiri ditekan
public void OnLeftButtonPressed()
{
    MoveLeft();
}
// Fungsi yang dipanggil ketika tombol kiri dilepas
public void OnLeftButtonReleased()
{
    StopMovement();
}
// Fungsi yang dipanggil ketika tombol kanan ditekan
public void OnRightButtonPressed()
{
    MoveRight();
}
// Fungsi yang dipanggil ketika tombol kanan dilepas

```

```

public void OnRightButtonReleased()
{
    StopMovement();
}
// Fungsi untuk melompat yang dipanggil melalui UI Button
public void TriggerJump()
{
    if (isGrounded) // Pastikan hanya lompat saat berada di
tanah
    {
        animator.SetBool("Jumping", true);
        jump = true;
    }
}
// Fungsi OnCollisionEnter2D untuk menangani tabrakan dengan
musuh
private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag("Enemy"))
    {
        ReduceHealth();
    }
}
// Fungsi untuk menangani trigger dengan garis finish
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.CompareTag("GarisFinish"))
    {
        Debug.Log("Player reached the finish line.");
        GameWinManager.instance.ShowWinPanel(100,
coinManager.GetTotalCoins()); // Sesuaikan dengan logika game
Anda
    }
    if (collision.gameObject.CompareTag("Enemy"))
    {
        ReduceHealth();
    }
}
}

```

Lampiran 12 : *Source code LevelManager*

```

using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class LevelManager : MonoBehaviour
{
    public Button level1Button;
    public Button level2Button;
    public Button level3Button;

    public GameObject level2ButtonObject;
    public GameObject level3ButtonObject;
    private void Start()

```

```

    {
        // Set Level 1 always active
        level1Button.interactable = true;
        level2ButtonObject.SetActive(false);
        level3ButtonObject.SetActive(false);
        // Load saved level completion status
        LoadLevelStatus();
        // Add listeners for buttons
        level1Button.onClick.AddListener(OnLevel1ButtonClick);
        level2Button.onClick.AddListener(OnLevel2ButtonClick); //
Add listener for Level 2 button
    }
    private void OnLevel1ButtonClick()
    {
        Debug.Log("Level 1 button clicked.");
        PlayerPrefs.SetInt("Level1Pressed", 1); // simpan Level 1
button has been pressed
        PlayerPrefs.Save();
        UpdateButtonStates(); // Refresh button states
    }

    private void OnLevel2ButtonClick()
    {
        Debug.Log("Level 2 button clicked.");
        PlayerPrefs.SetInt("Level2Pressed", 1); // simpan Level 2
button has been pressed
        PlayerPrefs.Save();
        UpdateButtonStates(); // Refresh button states
    }

    private void LoadLevelStatus()
    {
        UpdateButtonStates();
    }

    private void UpdateButtonStates()
    {
        int level1Pressed = PlayerPrefs.GetInt("Level1Pressed",
0);
        if (level1Pressed == 1)
        {
            level2ButtonObject.SetActive(true); // Show Level 2
button
            level2Button.interactable = true; // Enable Level
2 button
        }

        int level2Pressed = PlayerPrefs.GetInt("Level2Pressed",
0);
        if (level2Pressed == 1)
        {
            level3ButtonObject.SetActive(true); // Show Level 3
button
            level3Button.interactable = true; // Enable Level
3 button
        }
        int levelCompleted = PlayerPrefs.GetInt("LevelCompleted",
0);
        if (levelCompleted >= 3)

```

```

        {
            level3ButtonObject.SetActive(true); // Ensure Level
3 button is shown if already completed
            level3Button.interactable = true; // Enable Level
3 button
        }
    }

    public void CompleteLevel(int level)
    {
        PlayerPrefs.SetInt("LevelCompleted",
Mathf.Max(PlayerPrefs.GetInt("LevelCompleted", 0), level));
        PlayerPrefs.Save();
        SceneManager.LoadScene("LevelMenuScene");
    }

    public void ResetProgress()
    {
        PlayerPrefs.DeleteKey("Level1Pressed");
        PlayerPrefs.DeleteKey("Level2Pressed");
        PlayerPrefs.DeleteKey("LevelCompleted");
        SceneManager.LoadScene("LevelMenuScene");
    }
}

```

Lampiran 13: *Source code PopUpController*

```

using UnityEngine;
using UnityEngine.UI;

public class PopUpController : MonoBehaviour
{
    public GameObject[] popUpEdukasi; // Array untuk PopUpEdukasi
    public GameObject[] rollPapers; // Array untuk RollPaper
yang terkait
    public Button[] closeButtons; // Array untuk tombol close
    public Button buttonLevel2; // Referensi untuk Button
Level 2

    public int rollPaperUnit = 0; // Menyimpan jumlah unit
roll paper yang dimiliki oleh pemain
    public int maxRollPaperUnit = 5; // Batas maksimum roll
paper unit

    private void Start()
    {
        Debug.Log("Initializing PopUpController...");

        // Menonaktifkan semua pop-up edukasi dan tombol close
pada awal permainan
        foreach (var popUp in popUpEdukasi)
        {
            if (popUp != null)
            {
                popUp.SetActive(false); // Pastikan pop-up tidak
aktif saat mulai
                Debug.Log($"Pop-up {popUp.name} is set to
inactive.");
            }
        }
    }
}

```

```

    }
    else
    {
        Debug.LogWarning("Pop-up edukasi is null.");
    }
}

// Pastikan setiap tombol close terhubung dengan fungsi
OnCloseButtonPressed
for (int i = 0; i < closeButtons.Length; i++)
{
    int index = i; // Penting untuk menyalin indeks untuk
digunakan dalam lambda
        closeButtons[i].onClick.AddListener(() =>
OnCloseButtonPressed(index));
        closeButtons[i].interactable = false; // Menonaktifkan
tombol close pada awalnya
        Debug.Log($"Close button {closeButtons[i].name} is
connected to OnCloseButtonPressed with index {index}.");
    }

    // Pastikan Button Level 2 dimulai dalam keadaan non-
aktif
    buttonLevel2.interactable = false;
}

// Fungsi untuk membuka pop-up tertentu
public void ShowPopUp(int index)
{
    Debug.Log($"Attempting to show pop-up at index
{index}...");

    if (index >= 0 && index < popUpEdukasi.Length)
    {
        // Menambah roll paper unit jika belum mencapai batas
        if (rollPaperUnit < maxRollPaperUnit)
        {
            rollPaperUnit++; // Menambah roll paper unit
            Debug.Log($"Roll Paper Unit Increased:
{rollPaperUnit}");
        }

        // Menampilkan pop-up yang sesuai setelah unit roll
paper bertambah
        if (popUpEdukasi[index] != null)
        {
            popUpEdukasi[index].SetActive(true);
            Debug.Log($"Popup {index} is now visible.");
            closeButtons[index].interactable = true; //
Mengaktifkan tombol close untuk pop-up yang ditampilkan
        }
        else
        {
            Debug.LogWarning($"Pop-up edukasi at index {index}
is null.");
        }
    }
}

```

```

// Mengaktifkan Button Level 2 setelah pop-up
ditampilkan
if (index == 1) // Misalnya pop-up yang kedua
mengaktifkan Level 2
{
    buttonLevel2.interactable = true;
    Debug.Log("Button Level 2 is now active.");
}
else
{
    Debug.LogError("Index ShowPopUp tidak valid: " +
index);
}

// Fungsi untuk menutup pop-up dan menghilangkan roll paper
public void OnCloseButtonPressed(int index)
{
    Debug.Log($"Attempting to close pop-up and destroy roll
paper at index {index}...");

    if (index >= 0 && index < popUpEdukasi.Length)
    {
        // Menonaktifkan pop-up edukasi
        if (popUpEdukasi[index] != null)
        {
            popUpEdukasi[index].SetActive(false);
            Debug.Log($"Popup {index} is now hidden.");
        }
        else
        {
            Debug.LogWarning($"Pop-up edukasi at index {index}
is null.");
        }

        // Menghilangkan roll paper dan mengurangi unit roll
paper
        if (index < rollPapers.Length && rollPapers[index] !=
null)
        {
            Destroy(rollPapers[index]);
            rollPapers[index] = null; // Menghapus referensi
roll paper setelah dihancurkan
            Debug.Log($"Roll Paper {index} destroyed.");
        }

        // Mengurangi unit roll paper setelah menutup pop-up
        if (rollPaperUnit > 0)
        {
            rollPaperUnit--; // Mengurangi roll paper unit
setelah menutup pop-up
            Debug.Log($"Roll Paper Unit Decreased:
{rollPaperUnit}");
        }

        // Menonaktifkan tombol close setelah pop-up ditutup
closeButtons[index].interactable = false;

```



```
    }  
    else  
    {  
        Debug.LogError("Index OnCloseButtonPressed tidak  
valid: " + index);  
    }  
}  
}
```