

LAMPIRAN

LAMPIRAN -1 Foto Penelitian Di Sekolah



LAMPIRAN -2 Kartu Monitoring



KARTU MONITORING BIMBINGAN
 MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
 FAKULTAS TEKNIK
 UNIVERSITAS MUHAMMADIYAH PAREPARE

PROPOSAL

MAHASISWA : NURAI SYAH	Pembimbing I : Wahyuddin, S.Kom., M.Kom.
NIM : 220280121	Pembimbing II : Mughaffir Yunus, S.T., M.T.
Judul Skripsi : PERANCANGAN MEDIA PEMBELAJARAN INTERAKTIF SISTEM TATA SURYA	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1 - Tambahkan teori pembelajaran pada Unity	f	Konsultasi 1	Wf
Konsultasi 2 - Desain sistem diperbaiki	f	Konsultasi 2	Wf
Konsultasi 3 - Tidak menggunakan Marker	f	Konsultasi 3 Acc	Wf
Konsultasi 4 Acc	f	Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatian :

1. Minimalisir waktu konsultasi minimal 5 kali
2. Kartu ini wajib diisi oleh mahasiswa setiap konsultasi dan diisi oleh Pembimbing
3. Kartu ini wajib ditempelkan pada laporan skripsi dan menjadi salah satu persyaratan untuk ikut seminar proposal/ujian skripsi
4. Kartu ini dicetak di atas kertas karton berwarna hijau muda dan dicetak, limbal balik



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

HASIL

MAHASISWA :	NURAIYAH	Pembimbing I :	Wahyuddin, S.Kom., M.Kom.
NIM :	220280121	Pembimbing II :	Mughaffir Yunus, S.T., M.T.
Judul Skripsi :	PERANCANGAN MEDIA PEMBELAJARAN INTERAKTIF SISTEM TATA SURYA		

ARAHAN PEMBIMBING I	HARITGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARITGL & PARAF PEMBIMBING
Konsultasi 1 Acc f.	f.	Konsultasi 1	Ref
Konsultasi 2 Acc usang puly	f.	Konsultasi 2 Acc Aplikasi	Ref
Konsultasi 3		Konsultasi 3	Ref
Konsultasi 4		Konsultasi 4	Ref
Konsultasi 5		Konsultasi 5 Acc Semesta hasil	Ref

Lampir ke halaman sebelah.

Petunjuk :

1. Mahasiswa wajib melakukan minimal 5 kali
2. Kartu ini wajib diisi oleh mahasiswa setelah konsultasi dan disetujui Pembimbing
3. Kartu ini wajib diserahkan pada dosen yang telah mengisi untuk bisa diserahkan untuk hal selanjutnya jika diperlukan
4. Kartu ini dibuat di atas kertas berjenis kertas putih dan ditulis cetak tebal



KARTU MONITORING BIMBINGAN
MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH PAREPARE

SKRIPSI

MAHASISWA : NURAI SYAH	Pembimbing I : Wahyuddin, S.Kom, M.Kom.
NIM : 220280121	Pembimbing II : Mughaffir Yunus, S.T., M.T.
Judul Skripsi : PERANCANGAN MEDIA PEMBELAJARAN INTERAKTIF SISTEM TATA SURYA	

ARAHAN PEMBIMBING I	HARI/TGL & PARAF PEMBIMBING	ARAHAN PEMBIMBING II	HARI/TGL & PARAF PEMBIMBING
Konsultasi 1		Konsultasi 1	
Konsultasi 2		Konsultasi 2	
Konsultasi 3		Konsultasi 3 Acc Ujian Putar	
Konsultasi 4		Konsultasi 4	
Konsultasi 5		Konsultasi 5	

Lanjut ke halaman sebelah...

Perhatikan :

1. Mahasiswa wajib konsultasi minimal 5 kali

LAMPIRAN -3 Kuesioner

KUESIONER

- 1) Jawaban kamu tidak untuk menentukan nilai.
- 2) Kuesioner ini untuk mendapatkan informasi. Jawaban kamu sangat berarti untuk memperbaiki dan membangun aplikasi pelajaran tata surya yang sesuai dan lebih baik.

Nama :

Kelas :

1. Apakah kamu tahu apa saja planet yang ada di dalam tata surya?
 - a. Ya
 - b. Tidak
2. Pernahkah kamu belajar tentang sistem tata surya sebelumnya?
 - a. Ya
 - b. Tidak
3. Apakah kamu merasa kesulitan mengingat urutan planet dalam tata surya?
 - a. Ya
 - b. Tidak
4. Apakah kamu pernah menggunakan media interaktif untuk belajar tata surya?
 - a. Ya
 - b. Tidak
5. Apakah kamu merasa pelajaran tentang tata surya sulit untuk dipahami?
 - a. Ya
 - b. Tidak

TERIMAKASIH

KUESIONER

- 1) Jawaban kamu tidak untuk menentukan nilai.
- 2) Kuesioner ini untuk mendapatkan informasi. Jawaban kamu sangat berarti untuk memperbaiki dan membangun aplikasi pelajaran tata surya yang sesuai dan lebih baik.

Nama :

Kelas :

1. Bagaimana menurut kamu tampilan aplikasi ini?
 - a. Buruk
 - b. Biasa saja
 - c. Bagus
 - d. Sangat Bagus
2. Seberapa mudah kamu memahami informasi tentang planet di aplikasi ini?
 - a. Sulit
 - b. Biasa saja
 - c. Mudah
 - d. Sangat Mudah
3. Apakah aplikasi ini membantu kamu belajar tentang tata surya?
 - a. Tidak Membantu
 - b. Biasa saja
 - c. Membantu
 - d. Sangat Membantu
4. Apakah kamu senang menggunakan aplikasi ini untuk belajar tentang tata surya?
 - a. Tidak Senang
 - b. Biasa saja
 - c. Senang
 - d. Sangat Senang
5. Apakah fitur interaktif di aplikasi ini membantu kamu belajar lebih baik?
 - a. Tidak Membantu
 - b. Biasa saja
 - c. Membantu
 - d. Sangat Membantu

TERIMA KASIH

LAMPIRAN -4 Script Aplikasi

1. PlanetRotation

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class PlanetRotation :
MonoBehaviour
{
    public float rotationSpeed = 10f; //
    Kecepatan rotasi planet
    public GameObject ringObject; //
    Objek tambahan, misalnya cincin
    Saturnus
    public float ringRotationSpeed = 5f; //
    Kecepatan rotasi cincin
```

2. InteratifPlanet

```
using UnityEngine;

public class InteratifPlanet :
MonoBehaviour
{
    [SerializeField] private float
    rotationSpeed = 10f; // Kecepatan rotasi
    (diatur di Inspector)
    [SerializeField] private float
    zoomSpeed = 0.2f; // Kecepatan zoom
    (diatur di Inspector)
    [SerializeField] private float minZoom
    = 0.3f; // Skala minimum zoom (diatur
    di Inspector)
    [SerializeField] private float
    maxZoom = 5f; // Skala maksimum
    zoom (diatur di Inspector)
    private PlanetRotation planetRotation;
    // Referensi ke script PlanetRotation
    private bool isTouched = false; //
    Apakah objek sedang disentuh
```

```
public Vector3 ringRotationAxis =
Vector3.right; // Sumbu rotasi cincin
public bool isAutoRotating = true; //
Kontrol rotasi otomatis
void Update()
{
    if (isAutoRotating)
        { // Rotasi planet di sekitar sumbu Y
        transform.Rotate(Vector3.up *
        rotationSpeed * Time.deltaTime);
        }
    if (ringObject != null)
        {
            ringObject.transform.Rotate(ring
            RotationAxis * ringRotationSpeed *
            Time.deltaTime); } } }
```

```
void Start()
{ // Ambil script PlanetRotation dari
planet planetRotation =
GetComponent<PlanetRotation>(); }
void Update()
{ if (Input.touchCount > 0)
    { Touch touch = Input.GetTouch(0);
    // Lakukan Raycast dari posisi sentuhan
    Ray ray =
    Camera.main.ScreenPointToRay(touch.
    position);
    RaycastHit hit;
    if (Physics.Raycast(ray, out hit))
    { // Periksa apakah objek yang disentuh
    adalah objek ini
    if (hit.transform == transform)
    { isTouched = true;
    if (Input.touchCount == 1) // Satu jari:
    Rotasi manual
```



```

        {if (touch.phase ==
TouchPhase.Began && planetRotation
!= null)

    {// Hentikan rotasi otomatis saat disentuh
planetRotation.isAutoRotating = false;

        if (touch.phase == TouchPhase.Moved)
            {float rotationX =
touch.deltaPosition.y * rotationSpeed *
Time.deltaTime; float rotationY = -
touch.deltaPosition.x * rotationSpeed *
Time.deltaTime;
            transform.Rotate(rotationX, rotationY,
0, Space.World); }}

        else if (Input.touchCount ==
2) // Dua jari: Zoom
            {Touch touch1 = Input.GetTouch(0);
            Touch touch2 = Input.GetTouch(1);
            // Hitung jarak antara dua jari sekarang
            dan sebelumnya
                float currentDistance =
Vector2.Distance(touch1.position,
touch2.position);
                float previousDistance =
Vector2.Distance(touch1.position -
touch1.deltaPosition,
touch2.position - touch2.deltaPosition);
                float zoomDelta =
currentDistance - previousDistance;
                float newScale =
Mathf.Clamp(transform.localScale.x +
zoomDelta * zoomSpeed *
Time.deltaTime, minZoom, maxZoom);
                // Terapkan skala baru ke
planet transform.localScale = new
Vector3(newScale,newScale, newScale);
            }}

```

```

        else{ isTouched = false; //
Objek lain disentuh }} }
        else if (Input.touchCount == 0 &&
planetRotation != null && isTouched) //
Tidak ada sentuhan
            {// Aktifkan kembali rotasi otomatis
planetRotation.isAutoRotating = true;
            isTouched = false; // Reset status
sentuhan }}}

```

3. PindahScene

```

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

public class PindahScene :
MonoBehaviour
{ public string sceneName = "planet"; //
Nama scene kedua
    public GameObject
selectedPlanetObject; // Referensi ke
objek planet yang dipilih
    public string planetNameText; // Teks
untuk nama planet (judul)
    [TextArea(3, 10)] public string
planetDescriptionText; // Teks untuk
deskripsi planet (detail)

    public void LoadScene()
    {if (selectedPlanetObject != null &&
!string.IsNullOrEmpty(planetNameText
)&&
!string.IsNullOrEmpty(planetDescriptio
nText))
        {
            // Simpan data nama planet dan
deskripsi ke PlayerPrefs
            PlayerPrefs.SetString("SelectedP
lanetName",
selectedPlanetObject.name);
            PlayerPrefs.SetString("PlanetNa
meText", planetNameText);
        }
    }
}

```

```
PlayerPrefs.SetString("PlanetDescriptionText", planetDescriptionText);
```

```

// Pindah ke Scene 2
SceneManager.LoadScene(sceneName); }
else
{Debug.LogError("Data tidak lengkap! Pastikan semua data planet terisi."); }}

```

```

void Start()
{ // Menambahkan event listener untuk Button
    Button btn = GetComponent<Button>();
    if (btn != null)
    { btn.onClick.AddListener(LoadScene); } }

```

4. ButtonPindah

```

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

```

```

public class ButtonPindah : MonoBehaviour
{ public Button pindahSceneButton; // Referensi ke tombol yang akan digunakan untuk pindah scene

```

```

    public string sceneName = "Scene2";
    // Nama scene tujuan

```

```

void Start()
{ if (pindahSceneButton != null)
    { // Tambahkan listener ke tombol untuk memanggil fungsi LoadScene
        pindahSceneButton.onClick.AddListener(() => LoadScene(sceneName)); } }

    public void LoadScene(string sceneName)
    { SceneManager.LoadScene(sceneName); // Pindah ke scene yang ditentukan } }

```

5. GlobalAudioSettings

```
using UnityEngine;
```

```

public class GlobalAudioSettings : MonoBehaviour
{ public static GlobalAudioSettings Instance;

```

```

    void Awake()
    { // Singleton untuk memastikan hanya satu instance yang ada
        if (Instance == null)
        { Instance = this; DontDestroyOnLoad(gameObject); // Jangan hancurkan saat pindah scene }
        else
        { Destroy(gameObject); // Hancurkan duplikat } }

```

```

    // Fungsi untuk set volume ke PlayerPrefs

```

```

    public void SetGlobalVolume(float volume)
    { PlayerPrefs.SetFloat("MusicVolume", volume); // Simpan volume ke PlayerPrefs

```

```

        PlayerPrefs.Save(); }
    // Fungsi untuk mengambil volume dari PlayerPrefs

```

```

    public float GetGlobalVolume()
    { return PlayerPrefs.GetFloat("MusicVolume", 1f); // Default volume adalah 1.0 (max) }
    // Fungsi untuk menyimpan status musik (aktif/mati)

```

```

    public void SetMusicState(bool isOn)
    { PlayerPrefs.SetInt("MusicState", isOn ? 1 : 0); // 1 = Musik On, 0 = Musik Off

```

```

        PlayerPrefs.Save(); }
    // Fungsi untuk mengambil status musik dari PlayerPrefs
    public bool GetMusicState()

```

```

        {return
PlayerPrefs.GetInt("MusicState", 1) ==
1; // Default musik dalam keadaan On}}

```

6. AstronautJump

```
using UnityEngine;
```

```

public class AstronautJump :
MonoBehaviour
{public float jumpForce = 5f; // Tinggi
lompatan
    public float gravityScale = 1f; //
Gravitasi
    public AudioSource jumpSound; //
Referensi untuk suara lompatan

    private Rigidbody2D rb;
    private bool isGrounded = true; //
Status apakah astronot di tanah

    void Start()
    {rb = GetComponent<Rigidbody2D>();
    if (rb == null)
        {Debug.LogError("Rigidbody2D
tidak ditemukan! Tambahkan
Rigidbody2D ke astronot.");}

    rb.gravityScale = gravityScale; // Set
gravitasi}

    public bool CanJump()
    {return isGrounded; // Hanya bisa
melompat jika di tanah}

    public void Jump()
    {if (isGrounded && rb != null)
        {rb.velocity = new
Vector2(rb.velocity.x, jumpForce); //
Lompatan isGrounded = false; // Tidak
lagi di tanah

        // Mainkan suara lompatan jika
AudioSource tersedia
        if (jumpSound != null)
            {jumpSound.Play();}

```

```

        } }}

    void OnCollisionEnter2D(Collision2D
collision)
        {if
(collision.collider.CompareTag("Planet"
))
            {isGrounded = true; // Astronot
kembali menyentuh tanah }}

    void OnCollisionExit2D(Collision2D
collision)
        {if
(collision.collider.CompareTag("Planet"
))
            {isGrounded = false; // Astronot
meninggalkan tanah }}}

```

7. PlanetClick

```

using UnityEngine;
public class PlanetClick :
MonoBehaviour
{public GameObject astronaut;
void OnMouseDown()
    {AstronautJump jumpScript =
astronaut.GetComponent<AstronautJum
p>();
    if (jumpScript != null)
        {if (jumpScript.CanJump())
            {jumpScript.Jump();}
        else
            {Debug.Log("Astronot sedang
melompat, tunggu hingga menyentuh
planet.");}}
    else
        {Debug.LogError("Script
AstronautJump tidak ditemukan di
astronot!"); }}

```

8. Keluar

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class keluar : MonoBehaviour

```

```
{ public void ExitGame()                Debug.Log("APLIKASI KELUAR");
  {Application.Quit();                  }}
```

