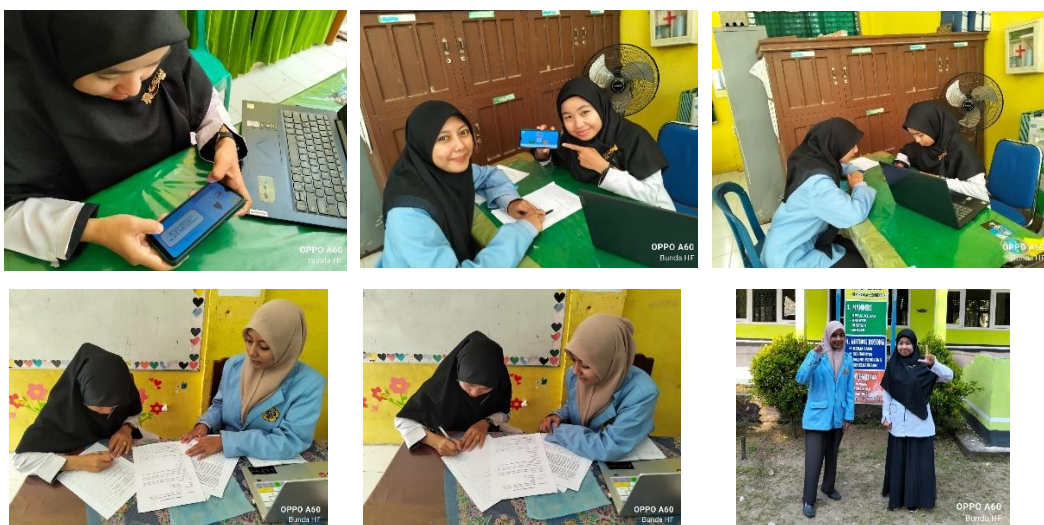


LAMPIRAN I

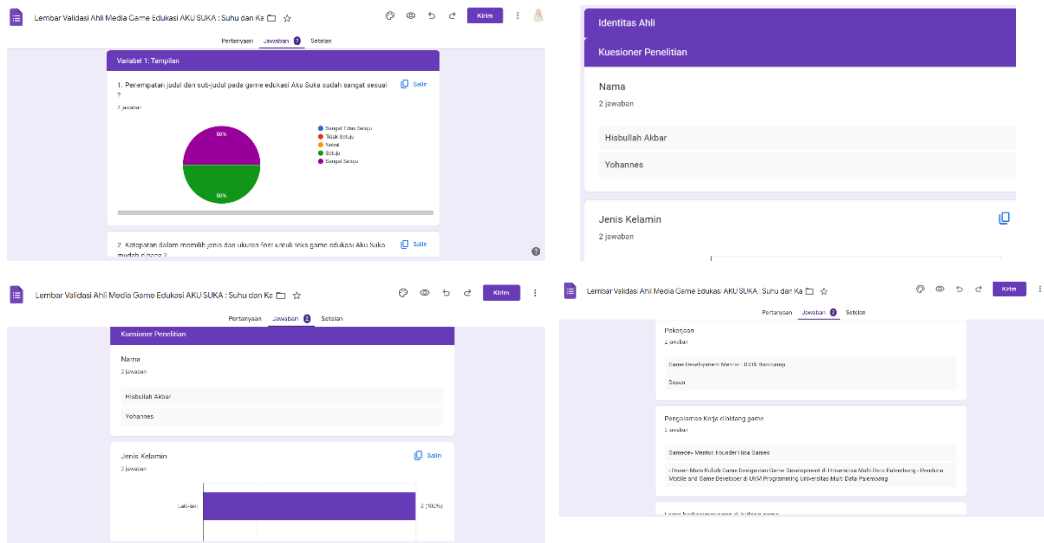
DOKUMENTASI



Pengujian Siswa Kelas V di SDN 06 Rantau Panjang



Pengujian Validasi Ahli Materi Bersama ibu Nurhidayati



Pengujian Validasi Ahli Media Bersama Bapak Hisbullah Akbar dan Bapak Yohannes

LAMPIRAN II

Hasil Instrumen Pertanyaan Ahli Materi

Hasil Instrumen Pertanyaan Ahli Media 1

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Tampilan						
P1.	Penempatan judul dan sub-judul pada game edukasi Aku Suka sudah sangat sesuai				✓	
P2.	Ketepatan dalam memilih jenis dan ukuran font untuk teks game edukasi Aku Suka mudah dibaca				✓	
P3.	Kejelasan gambar yang ada sudah sesuai dan menarik				✓	
P4.	Suara musik instrumen pada game edukasi suhu dan kalor sudah sesuai				✓	

P5.	Tampilan Warna pada background dan tombol navigasi game edukasi suhu dan kalor sesuai dan menarik				✓	
Aspek Kemudahan Penggunaan						
P6	Mudah memahami ikon tombol navigasi pada game edukasi Aku Suka					✓
P7.	Penyajian game edukasi Aku Suka dilakukan secara sistematis					✓
P8.	Petunjuk dalam penggunaan game edukasi Aku Suka disusun dengan jelas dan mudah dipahami				✓	
Aspek Kemanfaatan						
P9.	Game edukasi Aku Suka dirancang dengan mudah dijalankan				✓	
P.10	Game edukasi Aku Suka dirancang untuk menarik perhatian dan minat siswa				✓	

Hasil Instrumen Pertanyaan Ahli Materi 2

No	Pertanyaan	Skor				
		1	2	3	4	5
Aspek Tampilan						
P1.	Penempatan judul dan sub-judul pada game edukasi Aku Suka sudah sangat sesuai					✓
P2.	Ketepatan dalam memilih jenis dan ukuran font untuk teks game edukasi Aku Suka mudah dibaca			✓		

P3.	Kejelasan gambar yang ada sudah sesuai dan menarik					✓
P4.	Suara musik instrumen pada game edukasi suhu dan kalor sudah sesuai					✓
P5.	Tampilan Warna pada background dan tombol navigasi game edukasi suhu dan kalor sesuai dan menarik					✓
Aspek Kemudahan Penggunaan						
P6	Mudah memahami ikon tombol navigasi pada game edukasi Aku Suka					✓
P7.	Penyajian game edukasi Aku Suka dilakukan secara sistematis					✓
P8.	Petunjuk dalam penggunaan game edukasi Aku Suka disusun dengan jelas dan mudah dipahami					✓
Aspek Kemanfaatan						
P9.	Game edukasi Aku Suka dirancang dengan mudah dijalankan					✓
P.10	Game edukasi Aku Suka dirancang untuk menarik perhatian dan minat siswa					✓

LAMPIRAN III

Data Hasil Pretest dan Posttest Siswa Kelas V

Kode Siswa	Pre Test	Post Test
S-1	55	80
S-2	50	75
S-3	65	85
S-4	75	100
S-5	65	85

S-6	80	100
S-7	45	75
S-8	40	70
S-9	50	80
S-10	55	80
S-11	65	85
S-12	60	80
S-13	70	90
S-14	45	75
S-15	50	80
S-16	65	85
S-17	75	95
S-18	70	90
Rata-rata	60	83,88889

Lampiran Kode Pemrograman

1. Loading Bar

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

public class loading : MonoBehaviour
{
    public void load()
    {
        Debug.Log("load method called");
        SceneManager.LoadScene("S.Mulai");
    }
}
```

2. Loading Scene

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

public class loadscene : MonoBehaviour
{
    public float delayTime = 0.5f;

    public void LoadScene(string sceneName)
    {
        if (sceneName == "Game Basket")
            AudioController.instance.PlayGameBGM();
        StartCoroutine(DelayedLoadScene(sceneName));
    }

    private IEnumerator DelayedLoadScene(string sceneName)
    {
        yield return new WaitForSeconds(delayTime);
        SceneManager.LoadScene(sceneName);
    }

    public void TombolKeluar()
    {
        Application.Quit();
        Debug.Log("Game Close");
    }

    public void Mainkan()
    {
        SceneManager.LoadScene("S.Menu");
    }
}
```

```

    public void Home()
    {
        SceneManager.LoadScene("mainmenu");
    }
}

```

3. Canvas Manager

```

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

public class CanvasManager : Singleton<CanvasManager>
{
    [SerializeField] private GameObject
    _openingPanel, _gamePanel, _gameOpeningPanel, _popUpGameOver, _popUpWin;

    //Function yang dipanggil pada saat game baru dimulai
    private void Start()
    {
        Time.timeScale = 1;
        TurnOffAllUI();
        _openingPanel.SetActive(true);
    }

    //Function yang dipanggil untuk menampilkan game panel
    public void OnDisplayGamePanel()
    {
        TurnOffAllUI();
        _gamePanel.SetActive(true);
        _gameOpeningPanel.SetActive(true);
    }

    //Function yang dipanggil untuk menampilkan panel game over
    public void OnDisplayGameOver()
    {
        Time.timeScale = 0;
        TurnOffAllUI();
        _popUpGameOver.SetActive(true);
    }

    //Function yang dipanggil untuk menampilkan panel win
    public void OnDisplayWin()
    {
        Time.timeScale = 0;
        TurnOffAllUI();
        _popUpWin.SetActive(true);
    }

    //Function yang dipanggil untuk mematikan semua UI panel sebelum
    mengaktifkan UI baru (agar menghindari overlap UI)
    private void TurnOffAllUI()
    {
        _openingPanel.SetActive(false);
        _gameOpeningPanel.SetActive(false);
        _gamePanel.SetActive(false);
    }
}

```

```

        _popUpGameOver.SetActive(false);
        _popUpWin.SetActive(false);
    }
}

```

4. Drag Controller

```

using UnityEngine;
using UnityEngine.EventSystems;

public class DragController : MonoBehaviour
{
    private bool _isDragActive = false;
    private Vector2 _screenPosition;
    private Vector3 _worldPosition;
    private Draggable _lastDragged;
    private Camera _mainCamera;
    private float _leftBound;
    private float _rightBound;

    //Function yang dipanggil pada saat game dimulai, tepat sebelum
    start
    private void Awake()
    {
        _mainCamera = Camera.main;
    }

    //Function yang dipanggil pada saat game dimulai
    private void Start()
    {
        _leftBound = _mainCamera.ScreenToWorldPoint(new Vector3(0, 0,
0)).x;

        _rightBound = _mainCamera.ScreenToWorldPoint(new
Vector3(Screen.width, 0, 0)).x;
    }

    //Function yang dipanggil setiap waktu selama game jalan
    private void Update()
    {
        if(_isDragActive && (Input.GetMouseButtonUp(0) ||
(Input.touchCount == 1 && Input.GetTouch(0).phase ==
TouchPhase.Ended)))
        {
            Drop();
            return;
        }

        if (Input.GetMouseButton(0))
        {
            Vector3 mousePos = Input.mousePosition;
            _screenPosition = new Vector2(mousePos.x, mousePos.y);
        }
        else if (Input.touchCount > 0)

```



```

        {
            _screenPosition = Input.GetTouch(0).position;
        }
        else return;

        _worldPosition =
_mainCamera.ScreenToWorldPoint(_screenPosition);
        _worldPosition.z = 0;

        if (_isDragActive)
        {
            Drag();
        }
        else
        {
            RaycastHit2D hit = Physics2D.Raycast(_worldPosition,
Vector2.zero);
            if(hit.collider != null)
            {
                if (!hit.transform.TryGetComponent(out Draggable
draggable)) return;
                _lastDragged = draggable;
                InitDrag();
            }
        }
    }

    //Function yang dipanggil pada saat drag pada game object dimulai
    private void InitDrag()
    {
        _isDragActive = true;
    }

    //Function yang dipanggil pada saat drag pada game object
    berlangsung
    private void Drag()
    {
        if (!_lastDragged.TryGetComponent(out BoxCollider2D
boxCollider2D)) return;
        _lastDragged.transform.position = new
Vector2(Mathf.Clamp(_worldPosition.x, _leftBound +
boxCollider2D.bounds.size.x/2 , _rightBound -
boxCollider2D.bounds.size.x/2), _lastDragged.transform.position.y);
    }

    //Function yang dipanggil pada saat drag pada game object selesai
    private void Drop()
    {
        _isDragActive = false;
    }
}

```

5. Basket (Mengatur objek keranjang)

```

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

public class Basket : MonoBehaviour
{
    //Function untuk membuat basket dapat mendeteksi object yang
    collide/berkontak fisik dengan basket (Dalam kasus ini, basket
    mendeteksi tag game object yang tipenya sesuai dengan stage, yang tidak
    sesuai dengan stage, dan bom)
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if(collision.CompareTag(ReturnTagByCurrentStageType()))
        {
            GameManager.GetInstance().OnCollideWithCorrectObject();
        }
        else if(collision.CompareTag("Bomb"))
        {
            GameManager.GetInstance().OnCollideWithBomb();
        }
        else
        {
            GameManager.GetInstance().OnCollideWithIncorrectObject();
        }

        Destroy(collision.gameObject);
    }

    //Function untuk return tag sesuai dengan tipe stage yang sekarang
    private string ReturnTagByCurrentStageType()
    {
        switch (GameManager.GetInstance().CurrentStageType)
        {
            case StageType.Mencair:
                return "Mencair";
            case StageType.Menyublim:
                return "Menyublim";
            case StageType.Membeku:
                return "Membeku";
            case StageType.Menguap:
                return "Menguap";
            case StageType.Mengkristal:
                return "Mengkristal";
            case StageType.Mengembun:
                return "Mengembun";
            case StageType.Konduktor:
                return "Konduktor";
            case StageType.Isolator:
                return "Isolator";
            default:
                return null;
        }
    }
}

```

6. Falling Object (Mengatur objek jatuh kebawah)

```

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

public class FallingObject : MonoBehaviour
{
    [SerializeField] private float _fallSpeed = 2.0f;
    [SerializeField] private float _objLifetime = 10f;

    //Function yang dipanggil pada saat game dimulai
    private void Start()
    {
        StartCoroutine(DelayOnDestroy());
    }

    //Function yang dipanggil setiap saat selama game berlangsung
    void Update()
    {
        transform.position += Vector3.down * _fallSpeed *
Time.deltaTime;
    }

    //Function yang dipanggil untuk menghilangkan game object dalam
waktu tertentu
IEnumerator DelayOnDestroy()
    {
        yield return new WaitForSeconds(_objLifetime);
        Destroy(gameObject);
    }

    //Function yang dipanggil untuk menentukkan kecepatan jatuh dari
object
    public void SetFallSpeed(float newValue)
    {
        _fallSpeed = newValue;
    }
}

```

7. Health (Mengatur nyawa)

```

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

public class Health : MonoBehaviour
{
    [SerializeField] private List<GameObject> _healthIconList = new
List<GameObject>();
    [SerializeField] private GameEventNoParam _onGameOver;
    public int _totalHealth = 5;
    public int _howManyHealthInThisLevel = 5;

    //Function yang dipanggil pada saat game object in aktif atau pada
saat game basket dimulai
    private void OnEnable()

```

```

    {
        _totalHealth = _howManyHealthInThisLevel;
        foreach (GameObject icon in _healthIconList)
        {
            icon.SetActive(false);
        }
        for (int i = 0; i < _totalHealth; i++)
        {
            _healthIconList[i].SetActive(true);
        }
    }

    //Function yang dipanggil untuk mereset health player (biasanya
    dipanggil pada saat game dimulai lagi atau restart)
    public void OnResetHealth()
    {
        _totalHealth = _howManyHealthInThisLevel;
        foreach (GameObject icon in _healthIconList)
        {
            icon.SetActive(false);
        }
        for (int i = 0; i < _totalHealth; i++)
        {
            _healthIconList[i].SetActive(true);
        }
    }

    //Function yang dipanggil untuk mengurangi health player (Apabila
    basket collide dengan bomb)
    public void OnDecreaseHealth()
    {
        _totalHealth -= 1;
        _healthIconList[_totalHealth].gameObject.SetActive(false);

        if (_totalHealth <= 0) _onGameOver.Raise();
    }
}

```

8. Timer (Mengatur Waktu)

```

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

public class Timer : MonoBehaviour
{
    [SerializeField] private TextMeshProUGUI _timerText;
    [SerializeField] private GameEventNoParam _onGameOver;

    public bool stopTimer = false;
    [SerializeField] private float _countdownTime = 60;

    //Function yang dipanggil pada saat gamenya dimulai
    private void Start()
    {

```

```

        stopTimer = true;
    }

    //Function yang dipanggil untuk mereset timer (biasanya diapnggil
pada saat ingin restart level)
    public void OnResetTimer()
    {
        _countdownTime = 60;
        stopTimer = false;
        UpdateTimerText();
    }

    //Function yang dipanggil setiap saat selama game berlangsung
    void Update()
    {
        if (stopTimer) return;

        if (_countdownTime > 0)
        {
            _countdownTime -= Time.deltaTime;
            UpdateTimerText();
        }
        else
        {
            Debug.Log("Timer expired!");
            _onGameOver.Raise();
            _countdownTime = 0;
        }
    }

    //Function yang dipanggil untuk mengupdate timer text yang ada di
UI
    void UpdateTimerText()
    {
        int minutes = Mathf.FloorToInt(_countdownTime / 60);
        int seconds = Mathf.FloorToInt(_countdownTime % 60);
        string timerString = string.Format("{0:00}:{1:00}", minutes,
seconds);

        _timerText.SetText(timerString);
    }
}

```

9. Score

```

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

public class Score : MonoBehaviour
{
    [SerializeField] private TextMeshProUGUI _scoreText;
    [SerializeField] int _currentScore = 0;
    [SerializeField] int _totalScore = 0;
}

```

```

//Function yang dipanggil pada saat game dimulai
private void Start()
{
    _currentScore = 0;
    _totalScore = 0;
}

//Function yang dipanggil untuk menaikkan score stage dan total
score dengan jumlah 10
public void IncreaseScore()
{
    _totalScore += 10;
    _currentScore += 10;
    _scoreText.SetText(_currentScore.ToString());
}

//Function yang dipanggil untuk mereset score stage dari 0
(dipanggil pada saat transisi stage)
public void ResetScore()
{
    _currentScore = 0;
    _scoreText.SetText(_currentScore.ToString());
}
}

```

10. Game Manager

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;
using Unity.VisualScripting;

public enum StageType
{
    Mencair,
    Menyublim,
    Membeku,
    Menguap,
    Mengkristal,
    Mengembun,
    Konduktor,
    Isolator,
    None
}

public class GameManager : Singleton<GameManager>
{
    [SerializeField] private GameEventNoParam
_onCollideWithCorrectObject;
    [SerializeField] private GameEventNoParam
_onCollideWithIncorrectObject;
    [SerializeField] private GameEventNoParam _onCollideWithBomb;
    [SerializeField] private GameEventNoParam _onGameStart;
    [SerializeField] private GameEventNoParam _onWin;
    [SerializeField] private int _score;
}

```

```

        [SerializeField] private int _totalScore;
        [SerializeField] private TextMeshProUGUI
_scoreText, _totalScoreText;
        [SerializeField] private List<GameObject> _spawnerList = new();
        [SerializeField] private List<GameObject> _gameOpeningPanelList =
new();
        [SerializeField] private GameObject _gameRelatedObjects;
        [SerializeField] private StageType _currentStageType;

        public StageType CurrentStageType => _currentStageType;

        //Function yang dipanggil pada saat game dimulai
        private void Start()
        {
            _gameRelatedObjects.SetActive(false);
            TurnOffSpawners();
            _currentStageType = StageType.Mencair;
            _totalScore = 0;
        }

        //Funtion yang dipanggil pada saat sesi game drag basket dimulai
        public void OnStartGame()
        {
            Time.timeScale = 1;

            _onGameStart.Raise();
            _score = 0;
            _scoreText.SetText(_score.ToString());
            _totalScoreText.SetText(_totalScore.ToString());
            _gameRelatedObjects.SetActive(true);
            _spawnerList[(int)_currentStageType].SetActive(true);
        }

        //Funtion yang dipanggil bila health player sudah 0 atau sudah game
over
        public void OnGameOver()
        {
            _gameRelatedObjects?.SetActive(false);
        }

        //Function yang dipanggil bila basket collide dengan objek yang
sesuai di stage tersebut
        public void OnCollideWithCorrectObject()
        {
            _onCollideWithCorrectObject.Raise();
            _score += 10;
            _totalScore += 10;
            _scoreText.SetText(_score.ToString());
            _totalScoreText.SetText(_totalScore.ToString());

            if(_score >= 100)
            {
                if(_currentStageType == StageType.Isolator)
                {
                    _onWin.Raise();
                }
            }
        }

```

```

        _gameRelatedObjects?.SetActive(false);
    }
    else
    {
        OnProceedNextStage();
    }
}

//Function yang dipanggil bila basket collide dengan objek yang
tidak sesuai di stage tersebut
public void OnCollisionWithIncorrectObject()
{
    _onCollideWithIncorrectObject.Raise();
    if (_score > 0)
    {
        _score -= 5;
        _scoreText.SetText(_score.ToString());
    }

    if(_totalScore > 0)
    {
        _totalScore -= 5;
        _totalScoreText.SetText(_totalScore.ToString());
    }
}

//Function yang dipanggil bila basket collide dengan bomb
public void OnCollisionWithBomb()
{
    _onCollideWithBomb.Raise();
}

//Function yang dipanggil bila game sudah masuk ke stage
selanjutnya
private void OnProceedNextStage()
{
    Time.timeScale = 0;
    _currentStageType++;
    TurnOffSpawners();
    _gameRelatedObjects.gameObject.SetActive(false);
    _gameOpeningPanelList[(int)_currentStageType].SetActive(true);
}

//Function yang dipanggil untuk menonaktifkan seluruh spawners
(isolator,konduktor,menguap spawners, dll) sebelum menonaktifkan
spawner baru agar memastikan spawner yang aktif hanya 1 setiap stage
private void TurnOffSpawners()
{
    foreach (GameObject spawner in _spawnerList)
    {
        spawner.gameObject.SetActive(false);
    }
}
}

```


11. SFX Manager

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

public class SFXManager : MonoBehaviour
{
    [SerializeField] private AudioSource _audioSource;
    [SerializeField] private AudioClip _onCorrectSFX;
    [SerializeField] private AudioClip _onIncorrectSFX;
    [SerializeField] private AudioClip _onBombSFX;
    [SerializeField] private AudioClip _onClickSFX;

    //Function yang dipanggil untuk ngeplay sfx correct
    public void OnCorrectAnswer()
    {
        _audioSource.PlayOneShot(_onCorrectSFX);
    }

    //Function yang dipanggil untuk ngeplay sfx incorrect
    public void OnIncorrectAnswer()
    {
        _audioSource.PlayOneShot(_onIncorrectSFX);
    }

    //Function yang dipanggil untuk ngeplay sfx bom
    public void OnCollisionWithBombs()
    {
        _audioSource.PlayOneShot(_onBombSFX);
    }

    //Function yang dipanggil untuk ngeplay sfx button bila klik button
    public void OnClickButtonSFX()
    {
        _audioSource.PlayOneShot(_onClickSFX);
    }
}
```