LAMPIRAN

LAMPIRAN SCRIPT

```
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.ARFoundation;
using UnityEngine.XR.ARSubsystems;

/// <summary>
/// LAMPIRAN KODINGAN — ARPlacementController (Markerless via AR Foundation)
/// ------------------------------------------------------------------------
/// Fitur:
/// - Tap pada permukaan (plane) untuk menempatkan prefab furnitur.
/// - Menyimpan referensi objek aktif agar bisa di-rotate/scale (lihat ObjectManipulator).
/// - Integrasi dengan ProductInfoPanel untuk menampilkan detail.
///
/// Dependensi:
/// - AR Session Origin + ARRaycastManager + ARPlaneManager pada scene.
/// - Event System aktif untuk Input UI.
/// </summary>
public class ARPlacementController : MonoBehaviour
{
    [Header("Komponen ARFoundation")]
    public ARRaycastManager raycastManager;

    [Header("Prefab Default (opsional)")]
    public GameObject defaultPrefab;

    [Header("Referensi UI")]
    public ProductInfoPanel infoPanel;

    private static List<ARRaycastHit> s_Hits = new();
```

```
private GameObject currentObject;

/// <summary>
/// Panggil dari UI Catalog untuk mengganti prefab yang akan diletakkan.
/// </summary>
public void PlaceFromItem(FurnitureItem item)
{
    if (item == null || item.prefab3D == null) return;
    defaultPrefab = item.prefab3D;
    infoPanel?.Show(item);
}

void Update()
{
    if (Input.touchCount == 0) return;
    var touch = Input.GetTouch(0);
    if (touch.phase != TouchPhase.Began) return;

    if (raycastManager.Raycast(touch.position, s_Hits, TrackableType.PlaneWithinPolygon))
    {
        Pose pose = s_Hits[0].pose;
        SpawnOrMove(pose.position, pose.rotation);
    }
}

private void SpawnOrMove(Vector3 position, Quaternion rotation)
{
    if (currentObject == null)
    {
        var prefab = defaultPrefab != null ? defaultPrefab : infoPanel?.GetActivePrefab();
        if (prefab == null)
        {
```

```
            Debug.LogWarning("[ARPlacement] Tidak ada prefab yang diset.");
            return;
        }
        currentObject = Instantiate(prefab, position, rotation);
        currentObject.AddComponent<ObjectManipulator>(); // agar bisa pinch/rotate
    }
    else
    {
        currentObject.transform.SetPositionAndRotation(position, rotation);
    }
  }
}
```

```
using UnityEngine;

/// <summary>
/// LAMPIRAN KODINGAN — ObjectManipulator
/// ---------------------------------------
/// Gestur dua jari untuk ROTATE dan PINCH untuk SCALE pada objek 3D.
/// Tempelkan ke setiap objek 3D yang ingin dimanipulasi.
/// </summary>
public class ObjectManipulator : MonoBehaviour
{
    [Header("Pengaturan Manipulasi")]
    public float rotateSpeed = 0.2f;   // kepekaan rotasi
    public float scaleSpeed  = 0.01f;  // kepekaan skala
    public float minScale = 0.1f;
    public float maxScale = 5f;

    private float lastDistance;
```

```
private bool rotating;


void Update()
{
    if (Input.touchCount == 1)
    {
        // Satu jari: rotasi horizontal (drag)
        var t = Input.GetTouch(0);
        if (t.phase == TouchPhase.Moved)
        {
            float delta = t.deltaPosition.x * rotateSpeed;
            transform.Rotate(0f, -delta, 0f, Space.World);
            rotating = true;
        }
        else if (t.phase == TouchPhase.Ended)
        {
            rotating = false;
        }
    }
    else if (Input.touchCount >= 2)
    {
        // Dua jari: pinch untuk scale
        var t0 = Input.GetTouch(0);
        var t1 = Input.GetTouch(1);

        float curDist = Vector2.Distance(t0.position, t1.position);
        if (t1.phase == TouchPhase.Began)
            lastDistance = curDist;

        float diff = curDist - lastDistance;
        lastDistance = curDist;
```

```
            float newScale = Mathf.Clamp(transform.localScale.x + diff * scaleSpeed, minScale, maxScale);
            transform.localScale = new Vector3(newScale, newScale, newScale);
        }
    }
}


#if VUFORIA_PRESENT
using UnityEngine;
using Vuforia;

/// <summary>
/// LAMPIRAN KODINGAN — VuforiaGroundPlaneController (Opsional)
/// ------------------------------------------------------------
/// Jika menggunakan Vuforia (Ground Plane), aktifkan define VUFORIA_PRESENT
/// dan tempelkan script ini pada GameObject yang punya Plane Finder.
/// Pastikan Vuforia Engine sudah terpasang via Package Manager.
/// </summary>
public class VuforiaGroundPlaneController : MonoBehaviour
{
    public GameObject defaultPrefab;
    public ProductInfoPanel infoPanel;

    public void OnContentPlaced(Transform anchor)
    {
        var prefab = infoPanel?.GetActivePrefab() ?? defaultPrefab;
        if (prefab == null)
        {
            Debug.LogWarning("[Vuforia] Prefab tidak tersedia.");
            return;
        }
        var go = Instantiate(prefab, anchor.position, anchor.rotation);
```

```
        go.AddComponent<ObjectManipulator>();
    }
}
#endif



using UnityEngine;

/// <summary>
/// LAMPIRAN KODINGAN — AudioManager (Simple Singleton)
/// ----------------------------------------------------
/// Memutar SFX klik dan efek lain secara global.
/// </summary>
public class AudioManager : SimpleSingleton<AudioManager>
{
    public AudioSource sfxSource;
    public AudioClip sfxClick;

    public void PlayClick()
    {
        if (sfxSource != null && sfxClick != null) sfxSource.PlayOneShot(sfxClick);
    }
}



using System.Collections.Generic;
using UnityEngine;

/// <summary>
/// LAMPIRAN KODINGAN — CatalogDatabase
/// -----------------------------------
```

```
/// Menyimpan daftar item furnitur untuk dijadikan sumber data UI katalog.
/// Tempelkan komponen ini pada GameObject (mis. Systems) lalu isi daftar di Inspector.
/// </summary>
public class CatalogDatabase : MonoBehaviour
{
    public List<FurnitureItem> items = new List<FurnitureItem>();

    public FurnitureItem GetByName(string name)
    {
        return items.Find(i => i != null && i.nama == name);
    }

    public IReadOnlyList<FurnitureItem> All() => items;
}




using UnityEngine;
using UnityEngine.UI;

/// <summary>
/// LAMPIRAN KODINGAN — FurnitureItem (ScriptableObject)
/// ----------------------------------------------------
/// Berisi metadata 1 item furnitur untuk katalog AR.
/// Dipakai oleh CatalogDatabase, CatalogController, dan ProductInfoPanel.
/// Buat asset: Right Click Project > Create > AR Catalog > Furniture Item.
/// </summary>
[CreateAssetMenu(menuName = "AR Catalog/Furniture Item", fileName = "FurnitureItem")]
public class FurnitureItem : ScriptableObject
{
    [Header("Identitas")]
    public string nama;
```

```
    public string kategori;
    public string warna;

    [TextArea(3, 6)]
    public string deskripsi;

    [Header("Visual")]
    public Sprite thumbnail;
    public GameObject prefab3D;
    [Range(0.01f, 10f)]
    public float skalaDefault = 1f;

    [Header("Audio (opsional)")]
    public AudioClip sfxDeskripsi;
}



using UnityEngine;
using UnityEngine.SceneManagement;

/// <summary>
/// LAMPIRAN KODINGAN — SceneLoader
/// --------------------------------
/// Untuk berpindah scene dari menu utama ke scene AR/katalog/halaman tentang.
/// Kaitkan ke tombol UI via OnClick().
/// </summary>
public class SceneLoader : MonoBehaviour
{
    public void Load(string sceneName)
    {
        if (!string.IsNullOrEmpty(sceneName))
```

```
        SceneManager.LoadScene(sceneName);
    }

    public void Quit()
    {
#if UNITY_EDITOR
        UnityEditor.EditorApplication.isPlaying = false;
#else
        Application.Quit();
#endif
    }
}
```

```
using UnityEngine;

/// <summary>
/// LAMPIRAN KODINGAN — SimpleSingleton
/// -----------------------------------
/// Util sederhana untuk komponen singleton.
/// </summary>
public class SimpleSingleton<T> : MonoBehaviour where T : MonoBehaviour
{
    private static T _instance;
    public static T Instance
    {
        get
        {
            if (_instance == null)
            {
                _instance = FindObjectOfType<T>();
```

```
            if (_instance == null)
            {
                var go = new GameObject(typeof(T).Name);
                _instance = go.AddComponent<T>();
                DontDestroyOnLoad(go);
            }
        }
        return _instance;
    }
}

protected virtual void Awake()
{
    if (_instance == null)
    {
        _instance = this as T;
        DontDestroyOnLoad(gameObject);
    }
    else if (_instance != this)
    {
        Destroy(gameObject);
    }
}
```

```
using UnityEngine;

/// <summary>
/// LAMPIRAN KODINGAN — CatalogController
/// ---------------------------------------
```

```
/// Mengisi UI katalog dari CatalogDatabase dan memberikan event ke ARPlacementController.
/// </summary>
public class CatalogController : MonoBehaviour
{
    [Header("Sumber Data")]
    public CatalogDatabase database;

    [Header("Prefab UI Item")]
    public GameObject catalogItemUIPrefab;

    [Header("Kontainer Grid/Scroll Content")]
    public Transform contentRoot;

    [Header("Integrasi")]
    public ProductInfoPanel infoPanel;
    public ARPlacementController arPlacement;

    void Start()
    {
        BuildCatalog();
    }

    private void BuildCatalog()
    {
        if (database == null || database.All().Count == 0 || catalogItemUIPrefab == null || contentRoot == null)
        {
            Debug.LogWarning("[Catalog] Konfigurasi belum lengkap.");
            return;
        }

        foreach (Transform child in contentRoot)
            Destroy(child.gameObject);
```

```
        foreach (var item in database.All())
        {
            if (item == null) continue;
            var go = Instantiate(catalogItemUIPrefab, contentRoot);
            var ui = go.GetComponent<CatalogItemUI>();
            ui.Setup(item, this);
        }
    }

    public void OnItemSelected(FurnitureItem item)
    {
        infoPanel?.Show(item);
        arPlacement?.PlaceFromItem(item);
        AudioManager.Instance?.PlayClick();
    }
}
```

```
using UnityEngine;

/// <summary>
/// LAMPIRAN KODINGAN — CatalogController
/// ---------------------------------------
/// Mengisi UI katalog dari CatalogDatabase dan memberikan event ke ARPlacementController.
/// </summary>
public class CatalogController : MonoBehaviour
{
    [Header("Sumber Data")]
    public CatalogDatabase database;
```

```
[Header("Prefab UI Item")]
public GameObject catalogItemUIPrefab;

[Header("Kontainer Grid/Scroll Content")]
public Transform contentRoot;

[Header("Integrasi")]
public ProductInfoPanel infoPanel;
public ARPlacementController arPlacement;

void Start()
{
    BuildCatalog();
}

private void BuildCatalog()
{
    if (database == null || database.All().Count == 0 || catalogItemUIPrefab == null || contentRoot == null)
    {
        Debug.LogWarning("[Catalog] Konfigurasi belum lengkap.");
        return;
    }

    foreach (Transform child in contentRoot)
        Destroy(child.gameObject);

    foreach (var item in database.All())
    {
        if (item == null) continue;
        var go = Instantiate(catalogItemUIPrefab, contentRoot);
        var ui = go.GetComponent<CatalogItemUI>();
        ui.Setup(item, this);
```

```
        }
    }

    public void OnItemSelected(FurnitureItem item)
    {
        infoPanel?.Show(item);
        arPlacement?.PlaceFromItem(item);
        AudioManager.Instance?.PlayClick();
    }
}



using UnityEngine;
using UnityEngine.UI;
using TMPro;

/// <summary>
/// LAMPIRAN KODINGAN — CatalogItemUI
/// ---------------------------------
/// Komponen UI untuk 1 kartu item pada katalog (thumbnail + nama + tombol pilih).
/// </summary>
public class CatalogItemUI : MonoBehaviour
{
    public Image thumbnail;
    public TMP_Text txtNama;
    public Button btnPilih;

    private FurnitureItem item;
    private CatalogController owner;

    public void Setup(FurnitureItem item, CatalogController owner)
```

```
    {
        this.item = item;
        this.owner = owner;
        if (thumbnail) thumbnail.sprite = item.thumbnail;
        if (txtNama) txtNama.text = item.nama;

        if (btnPilih != null)
        {
            btnPilih.onClick.RemoveAllListeners();
            btnPilih.onClick.AddListener(() => owner?.OnItemSelected(item));
        }
    }
}
```

```
using UnityEngine;
using UnityEngine.UI;
using TMPro;

/// <summary>
/// LAMPIRAN KODINGAN — ProductInfoPanel
/// ------------------------------------
/// Panel UI untuk menampilkan info item furnitur (nama, kategori, warna, deskripsi, thumbnail).
/// Juga menyediakan referensi prefab aktif untuk ARPlacement.
/// </summary>
public class ProductInfoPanel : MonoBehaviour
{
    [Header("UI Komponen")]
    public TMP_Text txtNama;
    public TMP_Text txtKategori;
    public TMP_Text txtWarna;
```

```
public TMP_Text txtDeskripsi;
public Image imgThumbnail;
public Button btnPutarDiAR;

[Header("Audio (opsional)")]
public AudioSource audioSource;

private FurnitureItem activeItem;

void Awake()
{
    if (btnPutarDiAR != null)
        btnPutarDiAR.onClick.AddListener(OnPutarDiARClicked);
}

public void Show(FurnitureItem item)
{
    activeItem = item;
    if (item == null) { gameObject.SetActive(false); return; }

    if (txtNama) txtNama.text = item.nama;
    if (txtKategori) txtKategori.text = item.kategori;
    if (txtWarna) txtWarna.text = item.warna;
    if (txtDeskripsi) txtDeskripsi.text = item.deskripsi;
    if (imgThumbnail) imgThumbnail.sprite = item.thumbnail;

    if (audioSource && item.sfxDeskripsi)
    {
        audioSource.clip = item.sfxDeskripsi;
        audioSource.Play();
    }
```

```
        gameObject.SetActive(true);
    }

    public GameObject GetActivePrefab()
    {
        return activeItem != null ? activeItem.prefab3D : null;
    }

    private void OnPutarDiARClicked()
    {
        // Tombol "Putar di AR" hanya menutup panel (opsional)
        gameObject.SetActive(false);
    }
}
```