

L

A

M

P

I

R

A

N

**MainMenu.cs**

```

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

public class Mainmenu : MonoBehaviour
{
    public AudioSource buttonsound;

    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame

    public void Level1()
    {
        buttonsound.PlayOneShot(buttonsound.clip);
        SceneManager.LoadScene("Level 1");
    }
    public void Menuutama()
    {
        buttonsound.PlayOneShot(buttonsound.clip);
        SceneManager.LoadScene("MenuUtama1");
    }

}

```

**MainMenu2.cs**

```

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

public class MainMenu2 : MonoBehaviour

```

```
{  
    public AudioSource buttonsound;  
    public Canvas BukaMenu;  
    public Canvas Menu_1;  
    public Canvas Menu_2;  
    public Canvas Menu_3;  
    public Canvas Menu_4;  
  
    void Start()  
    {  
        BukaMenu.enabled = true;  
        Menu_1.enabled = false;  
        Menu_2.enabled = false;  
        Menu_3.enabled = false;  
        Menu_4.enabled = false;  
    }  
    public void menu_1()  
    {  
        buttonsound.PlayOneShot(buttonsound.clip);  
        BukaMenu.enabled = false;  
        Menu_1.enabled = true;  
        Menu_2.enabled = false;  
        Menu_3.enabled = false;  
    }  
}
```

```
        Menu_4.enabled = false;
    }
    public void menu_2()
    {

        buttonsound.PlayOneShot(buttonsound.clip);
        BukaMenu.enabled = false;
        Menu_1.enabled = false;
        Menu_2.enabled = true;
        Menu_3.enabled = false;
        Menu_4.enabled = false;
    }
    public void menu_3()
    {

        buttonsound.PlayOneShot(buttonsound.clip);
        BukaMenu.enabled = false;
        Menu_1.enabled = false;
        Menu_2.enabled = false;
        Menu_3.enabled = true;
        Menu_4.enabled = false;
    }
    public void menu_4()
    {
        buttonsound.PlayOneShot(buttonsound.clip);
```

```
BukaMenu.enabled = false;

Menu_1.enabled = false;

Menu_2.enabled = false;

Menu_3.enabled = false;

Menu_4.enabled = true;

}

public void bukamenu()

{

    buttonsound.PlayOneShot(buttonsound.clip);

    BukaMenu.enabled = true;

    Menu_1.enabled = false;

    Menu_2.enabled = false;

    Menu_3.enabled = false;

}

public void Menuutama()

{

    buttonsound.PlayOneShot(buttonsound.clip);

    SceneManager.LoadScene("MenuUtama 1");

}

public void Lv11()

{

    buttonsound.PlayOneShot(buttonsound.clip);

    SceneManager.LoadScene("Level 1");
```

```

    }
    public void Lvl2()
    {
        buttonsound.PlayOneShot(buttonsound.clip);
        SceneManager.LoadScene("Level 2");
    }
    public void Lvl3()
    {
        buttonsound.PlayOneShot(buttonsound.clip);
        SceneManager.LoadScene("Level 3");
    }
    public void Lvl4()
    {
        buttonsound.PlayOneShot(buttonsound.clip);
        SceneManager.LoadScene("Level 4");
    }
}

```

### **MenuUtama.cs**

```

using System.Collections;
using System.Collections.Generic;
using UnityEditor;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UIElements;

public class MenuUtama : MonoBehaviour
{
    public GameObject[] images; // Array untuk menyimpan gambar-gambar
    private int currentIndex = 0; // Indeks gambar saat ini
    public AudioSource buttonsound;

```

```

public Canvas Menu_utama;
public Canvas Menu_Profil;
public Canvas Menu_petunjuk;
// Start is called before the first frame update
void Start()
{
    Menu_utama.enabled = true;
    Menu_Profil.enabled = false;
    Menu_petunjuk.enabled = false;

    for (int i = 0; i < images.Length; i++)
    {
        images[i].SetActive(i == currentIndex);
    }
}
public void Profil()
{
    buttonsound.PlayOneShot(buttonsound.clip);
    Menu_utama.enabled = true;
    Menu_Profil.enabled = true;
    Menu_petunjuk.enabled = false;
}
public void Petunjuk()
{
    buttonsound.PlayOneShot(buttonsound.clip);
    Menu_utama.enabled = false;
    Menu_Profil.enabled = false;
    Menu_petunjuk.enabled = true;
}
public void Utama()
{
    buttonsound.PlayOneShot(buttonsound.clip);
    Menu_utama.enabled = true;
    Menu_Profil.enabled = false;
    Menu_petunjuk.enabled = false;
}
public void NextImage()
{
    // Nonaktifkan gambar saat ini
    images[currentIndex].SetActive(false);
}

```

```

// Hitung indeks berikutnya
currentIndex = (currentIndex + 1) % images.Length;

// Aktifkan gambar berikutnya
images[currentIndex].SetActive(true);
}

// Method untuk gambar sebelumnya
public void PreviousImage()
{
// Nonaktifkan gambar saat ini
images[currentIndex].SetActive(false);

// Hitung indeks sebelumnya
currentIndex = (currentIndex - 1 + images.Length) % images.Length;

// Aktifkan gambar sebelumnya
images[currentIndex].SetActive(true);
}
}

```

### **MiniMap.cs**

```

using System.Collections;

using System.Collections.Generic;
using UnityEngine;

public class MiniMap : MonoBehaviour
{
public Transform Player;

private void LateUpdate()
{
Vector3 newPos = transform.position;
newPos.y = transform.position.y;
transform.position = newPos;

transform.rotation = Quaternion.Euler(90, Player.eulerAngles.y, 0);
}
}

```

### **Quest.cs**

```

using UnityEngine;

```

```
[System.Serializable]

public class Quest
{
    public string itemName;

    public string questDescription;
}

```

### **QuestManager.cs**

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

public class QuestManager : MonoBehaviour
{
    public TMP_Text questNotificationText;
    public List<Quest> quests;
    private int currentQuestIndex;
    private Quest currentQuest;

    private void Start()
    {
        currentQuestIndex = 0;
        StartCoroutine(ShowAndHideQuestNotification(8f));
    }

    private IEnumerator ShowAndHideQuestNotification(float duration)
    {
        // Tampilkan pesan
        ShowQuestNotification();

        // Tunggu selama 'duration' detik
        yield return new WaitForSeconds(duration);

        // Sembunyikan pesan
        HideQuestNotification();
    }

    private void ShowQuestNotification()
    {
        if (currentQuestIndex < quests.Count)

```

```

    {
        currentQuest = quests[currentQuestIndex];
        questNotificationText.text = currentQuest.questDescription;
        currentQuestIndex++;
    }
    else
    {
        questNotificationText.text = "All items found! Quest complete!";
    }
}

private void HideQuestNotification()
{
    // Hapus teks pesan
    questNotificationText.text = "";
}

public void CheckItem(string itemName)
{
    if (currentQuest != null && itemName == currentQuest.itemName)
    {
        StartCoroutine(ShowAndHideQuestNotification(8f));
    }
}
}
}

```

### **SuaraMute.cs**

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UIElements;
public class Suaramute : MonoBehaviour
{
    private int VolumeLevel = 50;
    public AudioSource suara1;
    public AudioSource suara2;
    public AudioSource suara3;
    public AudioSource suara4;
    public AudioSource suara5;
    public AudioSource suara6;
    public AudioSource suara7;
    public AudioSource suara8;
    public AudioSource suara9;
    public AudioSource suara10;
    public Button tombolSuara;
    // Start is called before the first frame update

```

```
public void mutesuara()
{
    suara1.mute = true;
    suara2.mute = true;
    suara3.mute = true;
    suara4.mute = true;
    suara5.mute = true;
    suara6.mute = true;
    suara7.mute = true;
    suara8.mute = true;
    suara9.mute = true;
    suara10.mute = true;
}

public void anmutesuara()
{
    suara1.mute = false;
    suara2.mute = false;
    suara3.mute = false;
    suara4.mute = false;
    suara5.mute = false;
    suara6.mute = false;
    suara7.mute = false;
    suara8.mute = false;
    suara9.mute = false;
    suara10.mute = false;
}
}
```

### **SuccesTextManager.cs**

```
using UnityEngine;

public class SuccessTextManager : MonoBehaviour
{
    public GameObject successText;

    private void Start()
    {
        // Aktifkan teks "Berhasil!" saat permainan dimulai
        successText.SetActive(true);
    }
}
```

### VrLookWalk.cs

```

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

public class VrLookWalk : MonoBehaviour
{
    public TMP_Text timerText;
    public TMP_Text questNotificationText;
    public Transform vrCamera;
    public float toggleAngle = 30.0f;
    public float speed = 3.0f;
    public bool moveForward;

    private CharacterController cc;
    private ItemManager itemManager;
    private QuestManager questManager;

    private void Start()
    {
        cc = GetComponent<CharacterController>();
        itemManager = FindObjectOfType<ItemManager>();
        questManager = FindObjectOfType<QuestManager>();
    }

    private void Update()
    {
        if (vrCamera.eulerAngles.x >= toggleAngle && vrCamera.eulerAngles.x
        < 90.0f)
        {
            moveForward = true;
        }
        else
        {
            moveForward = false;
        }

        if (moveForward)
        {
            Vector3 forward = vrCamera.TransformDirection(Vector3.forward);
            cc.SimpleMove(forward * speed);
        }

        PickupItem();
    }
}

```

```

}

private void PickupItem()
{
    RaycastHit hit;
    if (Physics.Raycast(vrCamera.position, vrCamera.forward, out hit, 3.0f))
    {
        GameObject hitObject = hit.collider.gameObject;

        if (itemManager != null)
        {
            itemManager.PickUpItem(hitObject);

            // Update quest
            if (questManager != null)
            {
                questManager.CheckItem(hitObject.name);
            }
        }
    }
}
}
}
}

```

### **RaycastButtonManager.cs**

```

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

public class RaycastButtonManager : MonoBehaviour
{
    public Transform vrCamera; // Transform kamera VR
    public List<ButtonTextPair> buttonTextPairs; // Daftar pasangan teks dan tombol

    void Update()
    {
        RaycastHit hit;
        if (Physics.Raycast(vrCamera.position, vrCamera.forward, out hit, 3.0f))
        {
            foreach (var pair in buttonTextPairs)
            {
                if (hit.collider.gameObject == pair.button.gameObject)
                {
                    pair.text.gameObject.SetActive(true);
                }
            }
        }
    }
}

```

```

        }
        else
        {
            pair.text.gameObject.SetActive(false);
        }
    }
}
else
{
    foreach (var pair in buttonTextPairs)
    {
        pair.text.gameObject.SetActive(false);
    }
}
}
}

```

```

[System.Serializable]
public class ButtonTextPair
{
    public Button button;
    public Text text;
}

```

### **LevelManager.cs**

```

using UnityEngine;
using UnityEngine.SceneManagement;

public class LevelManager : MonoBehaviour
{
    public void LoadNextLevel()
    {
        // Mendapatkan indeks scene saat ini
        int currentSceneIndex = SceneManager.GetActiveScene().buildIndex;

        // Memuat scene berikutnya jika ada
        SceneManager.LoadScene(currentSceneIndex + 1);
    }
}

```

### **ItemManager.cs**

```

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

```

```

using TMPro;
using UnityEngine;
using UnityEngine.UI;

public class ItemManager : MonoBehaviour
{
    public GameObject[] items;
    public Dictionary<string, int> foundItems = new Dictionary<string, int>();
    public Button nextLevelButton;
    public TMP_Text successText;
    public InventoryManager inventoryManager;

    private bool allItemsFound = false;

    private void Start()
    {
        nextLevelButton.gameObject.SetActive(false);
        successText.gameObject.SetActive(false);

        // Cari referensi ke InventoryManager
        inventoryManager = FindObjectOfType<InventoryManager>();
    }

    private void Update()
    {
        if (!allItemsFound && AllItemsFound())
        {
            successText.gameObject.SetActive(true);
            successText.text = "Berhasil!!! Semua item telah ditemukan!";
            nextLevelButton.gameObject.SetActive(true);

            allItemsFound = true;
        }
    }

    private bool AllItemsFound()
    {
        foreach (var item in items)
        {
            if (!foundItems.ContainsKey(item.name))
            {
                return false;
            }
        }
        return true;
    }
}

```

```
public void PickupItem(GameObject item)
{
    if (System.Array.Exists(items, i => i == item))
    {
        foundItems.Add(item.name, 1);
        item.SetActive(false);

        // Tentukan ukuran item berdasarkan jenisnya
        float size = 1.0f; // Ukuran default

        if (item.name == "Mainan")
        {
            size = 2.5f;
        }
        else if (item.name == "Kamera")
        {
            size = 2f;
        }
        else if (item.name == "Hp")
        {
            size = 25f;
        }
        else if (item.name == "Galon")
        {
            size = 2.5f;
        }
        else if (item.name == "Cctv")
        {
            size = 45f;
        }
        else if (item.name == "Sepatu")
        {
            size = 75f;
        }
        else if (item.name == "Basket")
        {
            size = 135f;
        }
        else if (item.name == "Botol")
        {
            size = 75f;
        }
        else if (item.name == "Jam")
        {
            size = 0.5f;
        }
    }
}
```

```

    }

    // Tambahkan item lain dengan ukuran yang sesuai

    inventoryManager.AddItemToInventoryUI(item.name, size);

    CheckAllItemsFound();
}
}

private void CheckAllItemsFound()
{
    foreach (var item in items)
    {
        if (!foundItems.ContainsKey(item.name))
        {
            return;
        }
    }
    // Jika semua item telah ditemukan, panggil fungsi AllItemsFound() dari
script InventoryManager
    inventoryManager.AllItemsFound();
}
}

```

### **InventoryManager.cs**

```

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

public class InventoryManager : MonoBehaviour
{
    public Dictionary<string, GameObject> itemPrefabs = new
Dictionary<string, GameObject>();
    public List<Image> inventoryImages = new List<Image>(); // List of
inventory images to store items

    public GameObject bears;
    public GameObject Topi;
    public GameObject Mainan;
    public GameObject Helm;
    public GameObject Kipas;
    public GameObject Tas;

```

```

public GameObject Laptop;
public GameObject Kamera;
public GameObject Hp;
public GameObject Galon;
public GameObject Cctv;
public GameObject Sepatu;
public GameObject Basket;
public GameObject Kursi;
public GameObject Botol;
public GameObject Jam;

public GameObject[] items; // Array of items to find

private Dictionary<string, Image> itemImages = new Dictionary<string,
Image>(); // Dictionary to track item images
private Dictionary<string, int> foundItems = new Dictionary<string, int>(); //
Dictionary to track found items

private void Start()
{
    if (inventoryImages.Count == 0)
    {
        Debug.LogError("Inventory Images belum diinisialisasi!");
        return;
    }

    itemPrefabs.Add("bears", bears);
    itemPrefabs.Add("Topi", Topi);
    itemPrefabs.Add("Mainan", Mainan);
    itemPrefabs.Add("Helm", Helm);
    itemPrefabs.Add("Kipas", Kipas);
    itemPrefabs.Add("Tas", Tas);
    itemPrefabs.Add("Laptop", Laptop);
    itemPrefabs.Add("Kamera", Kamera);
    itemPrefabs.Add("Hp", Hp);
    itemPrefabs.Add("Galon", Galon);
    itemPrefabs.Add("Cctv", Cctv);
    itemPrefabs.Add("Sepatu", Sepatu);
    itemPrefabs.Add("Basket", Basket);
    itemPrefabs.Add("Kursi", Kursi);
    itemPrefabs.Add("Botol", Botol);
    itemPrefabs.Add("Jam", Jam);

    // Populate itemImages dictionary
    foreach (var image in inventoryImages)
    {

```

```

        itemImages.Add(image.name, image);
    }
}

public void AddItemToInventoryUI(string itemName, float size)
{
    if (itemPrefabs.ContainsKey(itemName))
    {
        if (!foundItems.ContainsKey(itemName))
        {
            if (itemImages.ContainsKey(itemName))
            {
                Image targetImage = itemImages[itemName];
                GameObject newItem = Instantiate(itemPrefabs[itemName],
targetImage.transform);
                newItem.transform.SetParent(targetImage.transform, false);

                // Set the scale of the new item based on the provided size
                newItem.transform.localScale = Vector3.one * size;

                newItem.SetActive(true);

                foundItems.Add(itemName, 1);
            }
            else
            {
                Debug.LogWarning($"Image untuk {itemName} tidak
ditemukan!");
            }
        }
        else
        {
            Debug.LogWarning($"Item {itemName} sudah ada di inventaris!");
        }
    }
    else
    {
        Debug.LogWarning($"Prefab untuk {itemName} tidak ditemukan!");
    }
}

public void AllItemsFound()
{
    Debug.Log("Semua item ditemukan!");
}

```

```

    }
}

```

### **GazeInteraksi.cs**

```

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

public class GazeInteraksi : MonoBehaviour
{
    public float waktulihat = 2f;
    private float waktu;
    private bool lihatobjek;

    // Use this for initialization
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

        if (lihatobjek)
        {
            waktu += Time.deltaTime;

            if (waktu >= waktulihat)
            {
                // execute pointerdown handler
                ExecuteEvents.Execute(gameObject, new
                PointerEventData(EventSystem.current), ExecuteEvents.pointerDownHandler);
                waktu = 0f;
            }
        }

    }

    public void PointerEnter()
    {
        lihatobjek = true;
        Debug.Log("PointerEnter");
    }
}

```

```

    }

    public void PointerExit()
    {
        lihatobjek = false;
        Debug.Log("PointerExit");
    }

    public void PointerDown()
    {
        Debug.Log("PointerDown");
    }
}

```

### **GameOverManager.cs**

```

using UnityEngine;
using UnityEngine.UI;
using TMPro;

public class GameOverManager : MonoBehaviour
{
    public TMP_Text messageText; // Referensi ke komponen TextMeshPro
    untuk menampilkan pesan
    public Button Kembali; // Referensi ke tombol kembali ke menu
    public Button Ulang;

    private void Start()
    {
        // Sembunyikan tombol kembali ke menu saat permainan dimulai
        Kembali.gameObject.SetActive(false);
        Ulang.gameObject.SetActive(false);
    }

    private void OnEnable()
    {
        // Mendaftarkan metode HandleGameEnd untuk dipanggil saat event
        OnGameEnd dipicu
        CountdownTimer.OnGameEnd += HandleGameEnd;
    }

    private void OnDisable()
    {
        // Membatalkan pendaftaran metode HandleGameEnd saat objek
        dimatikan
        CountdownTimer.OnGameEnd -= HandleGameEnd;
    }
}

```

```

    }

    private void HandleGameEnd()
    {
        // Tampilkan pesan game over
        messageText.text = "Game Over!";

        // Aktifkan tombol kembali ke menu
        Ulang.gameObject.SetActive(true);
        Kembali.gameObject.SetActive(true);
    }
    public void AllItemsFound()
    {
        // Tampilkan pesan dan aktifkan tombol saat semua item ditemukan
        messageText.text = "Selamat! Anda telah menemukan semua item!";
    }
}

```

### **CountdownTimer.cs**

```

using System.Collections;
using UnityEngine;
using UnityEngine.UI;
using System;
using TMPro;

public class CountdownTimer : MonoBehaviour
{
    public static event System.Action OnGameEnd;
    public int totalTime = 60; // Total waktu
    private int currentTime; // Sisa waktu

    public TMP_Text timerText; // komponen untuk menampilkan waktu
    public TMP_Text messageText; // komponen untuk menampilkan pesan

    public Button Ulang; // Untuk Button kembali ke menu
    public Button Kembali; // Untuk Button level berikutnya

    private bool gameEnded = false; // Untuk Status apakah permainan sudah berakhir

    private void Start()
    {
        currentTime = totalTime;
    }
}

```

```

        StartCoroutine(Countdown());
    }

private System.Collections.IEnumerator Countdown()
{
    while (currentTime > 0)
    {
        yield return new WaitForSeconds(1f);
        currentTime--;
        UpdateTimerDisplay();
    }
    EndGame();
}

private void UpdateTimerDisplay()
{
    int minutes = currentTime / 60;
    int seconds = currentTime % 60;
    timerText.text = string.Format("Waktu: {0:00}:{1:00}", minutes, seconds);
}

private void EndGame()
{
    messageText.text = "Game Over!";
    if (!gameEnded)
    {
        Ulang.gameObject.SetActive(true);
        gameEnded = true;
        OnGameEnd?.Invoke();
    }
}

public void AllItemsFound()
{
    // Aktifkan tombol level berikutnya jika semua item telah ditemukan
    Kembali.gameObject.SetActive(true);
}

public void ReturnToMainMenu()
{
    // Kode untuk kembali ke menu utama
    // Misalnya, jika menggunakan Scene Management API, Anda bisa
    memuat scene menu utama di sini
}
}

```

### GameManager.cs

```

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

public class GameManager : MonoBehaviour
{
    public void AllItemsFound()
    {
        Debug.Log("Semua item ditemukan!");
        // Tambahkan logika untuk menyelesaikan level atau melakukan tindakan
        tertentu
        LoadNextLevel();
    }

    private void LoadNextLevel()
    {
        // Mendapatkan indeks scene saat ini
        int currentSceneIndex = SceneManager.GetActiveScene().buildIndex;

        // Memuat scene berikutnya jika ada
        SceneManager.LoadScene(currentSceneIndex + 1);
    }
}

```

### Lvl 2/ InventoryManager2.cs

```

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

public class InventoryManager2 : MonoBehaviour
{
    public Dictionary<string, GameObject> itemPrefabs = new
    Dictionary<string, GameObject>();
    public List<Image> inventoryImages = new List<Image>(); // List of
    inventory images to store items

    public GameObject Helm;
    public GameObject Kipas;
    public GameObject Tas;
    public GameObject Laptop;
}

```

```

// Start is called before the first frame update
public GameObject[] items; // Array of items to find

private Dictionary<string, Image> itemImages = new Dictionary<string,
Image>(); // Dictionary to track item images
private Dictionary<string, int> foundItems = new Dictionary<string, int>(); //
Dictionary to track found items

private void Start()
{
    if (inventoryImages.Count == 0)
    {
        Debug.LogError("Inventory Images belum diinisialisasi!");
        return;
    }
    itemPrefabs.Add("Helm", Helm);
    itemPrefabs.Add("Kipas", Kipas);
    itemPrefabs.Add("Tas", Tas);
    itemPrefabs.Add("Laptop", Laptop);

    foreach (var image in inventoryImages)
    {
        itemImages.Add(image.name, image);
    }
}

public void AddItemToInventoryUI(string itemName, float size)
{
    if (itemPrefabs.ContainsKey(itemName))
    {
        if (!foundItems.ContainsKey(itemName))
        {
            if (itemImages.ContainsKey(itemName))
            {
                Image targetImage = itemImages[itemName];
                GameObject newItem = Instantiate(itemPrefabs[itemName],
targetImage.transform);
                newItem.transform.SetParent(targetImage.transform, false);

                // Set the scale of the new item based on the provided size
                newItem.transform.localScale = Vector3.one * size;

                newItem.SetActive(true);

                foundItems.Add(itemName, 1);
            }
        }
    }
}

```

```

        else
        {
            Debug.LogWarning($"Image untuk {itemName} tidak
ditemukan!");
        }
    }
    else
    {
        Debug.LogWarning($"Item {itemName} sudah ada di inventaris!");
    }
}
else
{
    Debug.LogWarning($"Prefab untuk {itemName} tidak ditemukan!");
}
}

public void AllItemsFound()
{
    Debug.Log("Semua item ditemukan!");
}
}
}

```

### Lvl 2/ ItemManager2.cs

```

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

public class ItemManager2 : MonoBehaviour
{
    public GameObject[] items;
    public Dictionary<string, int> foundItems = new Dictionary<string, int>();
    public Button nextLevelButton;
    public TMP_Text successText;
    public InventoryManager2 inventoryManager2;

    private bool allItemsFound = false;

    private void Start()
    {
        nextLevelButton.gameObject.SetActive(false);
    }
}

```

```

successText.gameObject.SetActive(false);

// Cari referensi ke InventoryManager
inventoryManager2 = FindObjectOfType<InventoryManager2>();
}

private void Update()
{
    if (!allItemsFound && AllItemsFound())
    {
        successText.gameObject.SetActive(true);
        successText.text = "Berhasil!!! Semua item telah ditemukan!";
        nextLevelButton.gameObject.SetActive(true);

        allItemsFound = true;
    }
}

private bool AllItemsFound()
{
    foreach (var item in items)
    {
        if (!foundItems.ContainsKey(item.name))
        {
            return false;
        }
    }
    return true;
}

public void PickupItem(GameObject item)
{
    if (System.Array.Exists(items, i => i == item))
    {
        foundItems.Add(item.name, 1);
        item.SetActive(false);

        // Tentukan ukuran item berdasarkan jenisnya
        float size = 1.0f; // Ukuran default
        if (item.name == "Helm")
        {
            size = 2.5f;
        }
        else if (item.name == "Kipas")
        {
            size = 0.5f;
        }
    }
}

```

```

    }
    else if (item.name == "Tas")
    {
        size = 7.5f;
    }
    else if (item.name == "Laptop")
    {
        size = 9.5f;
    }

    // Tambahkan item lain dengan ukuran yang sesuai

    inventoryManager2.AddItemToInventoryUI(item.name, size);

    CheckAllItemsFound();
}

private void CheckAllItemsFound()
{
    foreach (var item in items)
    {
        if (!foundItems.ContainsKey(item.name))
        {
            return;
        }
    }
    // Jika semua item telah ditemukan, panggil fungsi AllItemsFound() dari
script InventoryManager
    inventoryManager2.AllItemsFound();
}
}

```

### **Lvl 2/ VrLookWalk2.cs**

```

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

public class VrLookWalk2 : MonoBehaviour
{
    public TMP_Text timerText;
    public Transform vrCamera;
}

```

```

public Transform head;
public float toggleAngle = 30.0f;
public float speed = 3.0f;
public bool moveForward;

private CharacterController cc;
private ItemManager2 itemManager2;

private void Start()
{
    cc = GetComponent<CharacterController>();
    itemManager2 = FindObjectOfType<ItemManager2>();

    // Set initial position of the camera to the head position
    vrCamera.position = head.position;
    vrCamera.rotation = head.rotation;
}

private void Update()
{
    if (vrCamera.eulerAngles.x >= toggleAngle && vrCamera.eulerAngles.x
< 90.0f)
    {
        moveForward = true;
    }
    else
    {
        moveForward = false;
    }

    if (moveForward)
    {
        Vector3 forward = vrCamera.TransformDirection(Vector3.forward);
        cc.SimpleMove(forward * speed);
    }

    PickupItem();
}

private void PickupItem()
{
    RaycastHit hit;
    if (Physics.Raycast(vrCamera.position, vrCamera.forward, out hit, 3.0f))
    {
        GameObject hitObject = hit.collider.gameObject;

```

```

        if (itemManager2 != null)
        {
            itemManager2.PickUpItem(hitObject);
        }
    }
}

```

### Lvl 3/ InventoryManager3.cs

```

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

public class InventoryManager3 : MonoBehaviour
{
    public Dictionary<string, GameObject> itemPrefabs = new
Dictionary<string, GameObject>();
    public List<Image> inventoryImages = new List<Image>(); // List of
inventory images to store items

    public GameObject Kamera;
    public GameObject Radio;
    public GameObject Galon;
    public GameObject Cctv;
    // Start is called before the first frame update
    public GameObject[] items; // Array of items to find

    private Dictionary<string, Image> itemImages = new Dictionary<string,
Image>(); // Dictionary to track item images
    private Dictionary<string, int> foundItems = new Dictionary<string, int>(); //
Dictionary to track found items

    private void Start()
    {
        if (inventoryImages.Count == 0)
        {
            Debug.LogError("Inventory Images belum diinisialisasi!");
            return;
        }
        itemPrefabs.Add("Kamera", Kamera);
        itemPrefabs.Add("Radio", Radio);
        itemPrefabs.Add("Galon", Galon);
    }
}

```

```

itemPrefabs.Add("Cctv", Cctv);

foreach (var image in inventoryImages)
{
    itemImages.Add(image.name, image);
}

}
public void AddItemToInventoryUI(string itemName, float size)
{
    if (itemPrefabs.ContainsKey(itemName))
    {
        if (!foundItems.ContainsKey(itemName))
        {
            if (itemImages.ContainsKey(itemName))
            {
                Image targetImage = itemImages[itemName];
                GameObject newItem = Instantiate(itemPrefabs[itemName],
targetImage.transform);
                newItem.transform.SetParent(targetImage.transform, false);

                // Set the scale of the new item based on the provided size
                newItem.transform.localScale = Vector3.one * size;

                newItem.SetActive(true);

                foundItems.Add(itemName, 1);
            }
            else
            {
                Debug.LogWarning($"Image untuk {itemName} tidak
ditemukan!");
            }
        }
        else
        {
            Debug.LogWarning($"Item {itemName} sudah ada di inventaris!");
        }
    }
    else
    {
        Debug.LogWarning($"Prefab untuk {itemName} tidak ditemukan!");
    }
}

public void AllItemsFound()

```

```

    {
        Debug.Log("Semua item ditemukan!");
    }
}

```

### Lvl 3/ ItemManager3.cs

```

using System.Collections;
using System.Collections.Generic;
using TMPro;
using UnityEngine;
using UnityEngine.UI;
public class ItemManager3 : MonoBehaviour
{
    public GameObject[] items;
    public Dictionary<string, int> foundItems = new Dictionary<string, int>();
    public Button nextLevelButton;
    public TMP_Text successText;
    public InventoryManager3 inventoryManager3;

    private bool allItemsFound = false;

    private void Start()
    {
        nextLevelButton.gameObject.SetActive(false);
        successText.gameObject.SetActive(false);

        // Cari referensi ke InventoryManager
        inventoryManager3 = FindObjectOfType<InventoryManager3>();
    }

    private void Update()
    {
        if (!allItemsFound && AllItemsFound())
        {
            successText.gameObject.SetActive(true);
            successText.text = "Berhasil!!! Semua item telah ditemukan!";
            nextLevelButton.gameObject.SetActive(true);

            allItemsFound = true;
        }
    }

    private bool AllItemsFound()

```

```

{
    foreach (var item in items)
    {
        if (!foundItems.ContainsKey(item.name))
        {
            return false;
        }
    }
    return true;
}

public void PickupItem(GameObject item)
{
    if (System.Array.Exists(items, i => i == item))
    {
        foundItems.Add(item.name, 1);
        item.SetActive(false);

        // Tentukan ukuran item berdasarkan jenisnya
        float size = 1.0f; // Ukuran default
        if (item.name == "Kamera")
        {
            size = 2f;
        }
        else if (item.name == "Radio")
        {
            size = 2.5f;
        }
        else if (item.name == "Galon")
        {
            size = 2.5f;
        }
        else if (item.name == "Cctv")
        {
            size = 45f;
        }

        // Tambahkan item lain dengan ukuran yang sesuai

        inventoryManager3.AddItemToInventoryUI(item.name, size);

        CheckAllItemsFound();
    }
}

private void CheckAllItemsFound()

```

```

    {
        foreach (var item in items)
        {
            if (!foundItems.ContainsKey(item.name))
            {
                return;
            }
        }
        // Jika semua item telah ditemukan, panggil fungsi AllItemsFound() dari
script InventoryManager
        inventoryManager3.AllItemsFound();
    }
}

```

### **Lvl 3/ VrLookWalk3.cs**

```

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

public class VrLookWalk3 : MonoBehaviour
{
    public TMP_Text timerText;
    public Transform vrCamera;
    public Transform head;
    public float toggleAngle = 30.0f;
    public float speed = 3.0f;
    public bool moveForward;

    private CharacterController cc;
    private ItemManager3 itemManager3;

    private void Start()
    {
        cc = GetComponent<CharacterController>();
        itemManager3 = FindObjectOfType<ItemManager3>();

        // Set initial position of the camera to the head position
        vrCamera.position = head.position;
        vrCamera.rotation = head.rotation;
    }

    private void Update()
    {

```

```
        if (vrCamera.eulerAngles.x >= toggleAngle && vrCamera.eulerAngles.x
< 90.0f)
        {
            moveForward = true;
        }
        else
        {
            moveForward = false;
        }

        if (moveForward)
        {
            Vector3 forward = vrCamera.TransformDirection(Vector3.forward);
            cc.SimpleMove(forward * speed);
        }

        PickupItem();
    }

private void PickupItem()
{
    RaycastHit hit;
    if (Physics.Raycast(vrCamera.position, vrCamera.forward, out hit, 3.0f))
    {
        GameObject hitObject = hit.collider.gameObject;

        if (itemManager3 != null)
        {
            itemManager3.PickUpItem(hitObject);
        }
    }
}
```