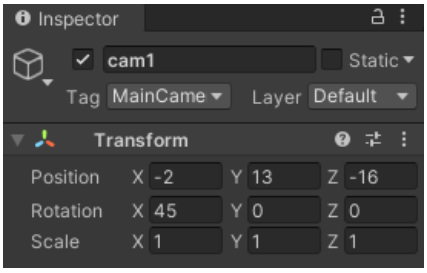
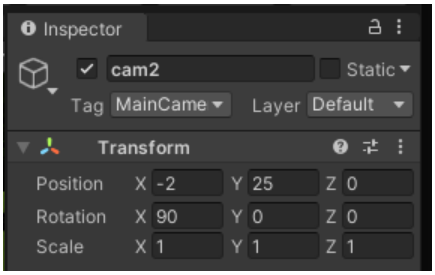
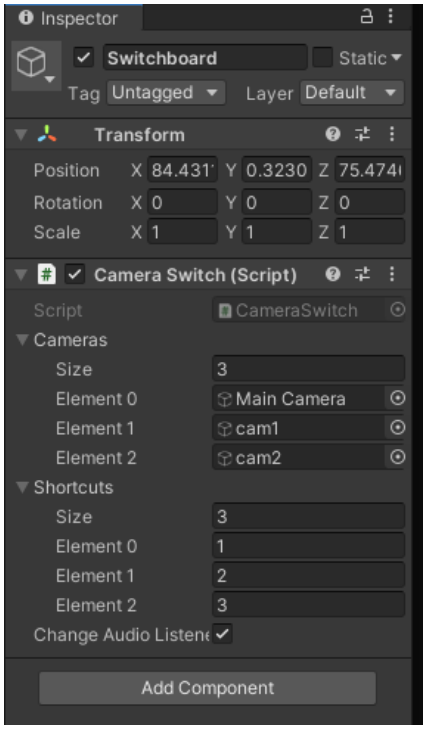
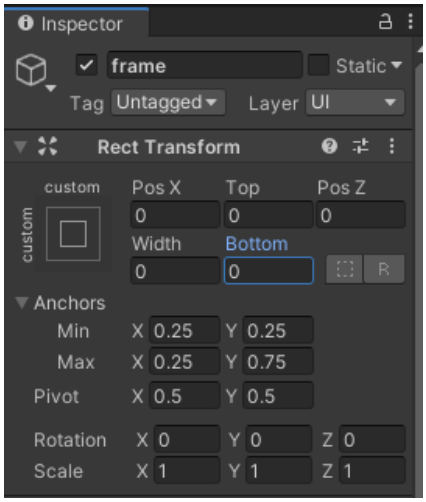
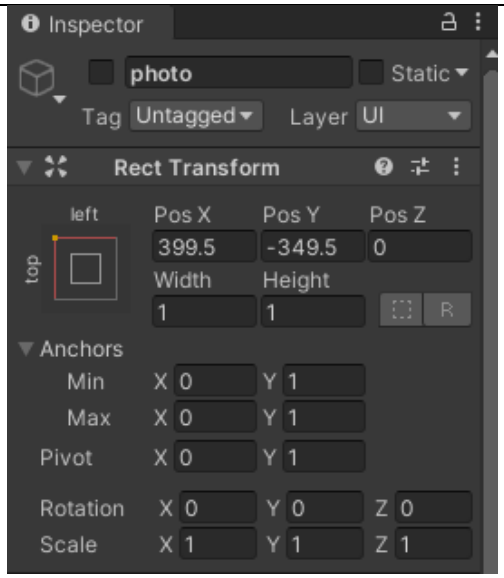
Pertemuan Ke- 6

Laporan

LANGKAH-LANGKAH PENGERJAAN	
1	Mendownload asset yang dibutuhkan.
2	Menambahkan ke dalam folder asset.
3	Menambahkan
4	Tambahkan New Terrain dan Secondary Terrain ke hierarchy dari scene NatureStartKit2
5	Setting letak main camera
6	Tambahkan Prefab DogKnight ke Hierarchy, dan atur tempat nya.
7	Tambahkan monster sebagai lawan nya nanti ke dalam hierarchy dari folder RPG Monster Duo PBR Polyart.
8	Tambahkan camera dan setting camera dengan depthmenjadi 1
9	Buat folder Scripts dan buat Script C# baru bernama PictureInPicture.cs.. lalu drag script tersebut ke Camera.
10	Pada inspector Camera, uncentang pada Audio Listener dan rubah parameter PictureInPicture Hor Algn menjadi right, Ver Aligh menjadi top, width menjadi 400, height menjadi 200.
11	Kemudian tambahkan lagi 2 Camera bernama cam1 dan cam2.
12	Rubah posisi cam1 dan 2 pada inspector menjadi: Cam 1:  Cam2:  Dan hilangkan centang pada komponen Camera dan Audio Listener.
13	Tambahkan Create Empety dan rubah namanya menjadi Switchboard

14	Buat script C# baru bernama CameraSwitch. Kemudian masukkan ke Swicthboard dengan cara di drag.
15	Pada inspector Switchoard atur seperti berikut ini: 
16	Tambahkan UI Image ke hierarchy dang anti namanya menjadi frame.
17	Kemudian, pada inspector, pada komponen Image (Script) ubah menjadi InputFieldBackground.
18	Lalu ubah react transform menjadi seperti berikut:  Dan hilangkan centang pada Fill Center
19	Tambahkan Raw Image ke Canvas dan rename menjadi photo. Dan pada inspector cari komponen raw image dan atur kolom texture menjadi None. Kemudian hilangkan centang pada kolom photo yang berada di atas.
20	Ubah React Transform pada photo menjadi seperti berikut ini:



21	Buat script bernama ScreenTexture.
22	Kemudian drag ke Main camera yang berada di Multipurpose Camera Rig.
23	Pada Photo GUI tambahan photo dan pada FrameGUI tambahkan frame.

KODE PEMROGRAMAN

PictureInPicture.cs

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

public class PictureInPicture : MonoBehaviour
{
    public enum hAlignment { left, center, right };
    public enum vAlignment { top, middle, bottom };
    public hAlignment horAlign = hAlignment.left;
    public vAlignment verAlign = vAlignment.top;
    public enum UnitsIn { pixels, screen_percentage };
    public UnitsIn unit = UnitsIn.pixels;
    public int width = 50;
    public int height = 50;
    public int xOffset = 0;
    public int yOffset = 0;
    public bool update = true;
    private int hsize, vsize, hloc, vloc;
    void Start()
    {
        AdjustCamera();
    }
    void Update()
    {
        if (update)
            AdjustCamera();
    }
```

```
    void AdjustCamera()
    {
        int sw = Screen.width;
        int sh = Screen.height;
        float swPercent = sw * 0.01f;
        float shPercent = sh * 0.01f;
        float xOffPercent = xOffset * swPercent;
        float yOffPercent = yOffset * shPercent;
        int xOff;
        int yOff;
        if (unit == UnitsIn.screen_percentage)
        {
            hsize = width * (int)swPercent;
            vsize = height * (int)shPercent;
            xOff = (int)xOffPercent;
            yOff = (int)yOffPercent;
        }
        else
        {
            hsize = width;
            vsize = height;
            xOff = xOffset;
            yOff = yOffset;
        }
        switch (horAlign)
        {
            case hAlignment.left:
                hloc = xOff;
                break;
            case hAlignment.right:
                int justifiedRight = (sw - hsize);
                hloc = (justifiedRight - xOff);
                break;
            case hAlignment.center:
                float justifiedCenter = (sw * 0.5f) - (hsize * 0.5f);
                hloc = (int)(justifiedCenter - xOff);
                break;
        }
        switch (verAlign)
        {
            case vAlignment.top:
                int justifiedTop = sh - vsize;
                vloc = (justifiedTop - (yOff));
                break;
            case vAlignment.bottom:
                vloc = yOff;
                break;
            case vAlignment.middle:
                float justifiedMiddle = (sh * 0.5f) - (vsize * 0.5f);
                vloc = (int)(justifiedMiddle - yOff);
                break;
        }
```

```
        GetComponent<Camera>().pixelRect = new Rect(hloc, vloc,
hsize, vsize);
    }

}
```

CameraSwitch.cs

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

public class CameraSwitch : MonoBehaviour
{
    public GameObject[] cameras;
    public string[] shortcuts;
    public bool changeAudioListener = true;
    void Update()
    {
        if (Input.anyKeyDown)
        {
            for (int i = 0; i < cameras.Length; i++)
            {
                if (Input.GetKeyDown(shortcuts[i]))
                    SwitchCamera(i);
            }
        }
    }
    void SwitchCamera(int index)
    {
        for (int i = 0; i < cameras.Length; i++)
        {
            if (i != index)
            {
                cameras[i].GetComponent<Camera>().enabled = false;
                if (changeAudioListener)
                    cameras[i].GetComponent<AudioListener>().enabled
= false;
            }
            else
            {
                cameras[i].GetComponent<Camera>().enabled = true;
                if (changeAudioListener)
                    cameras[i].GetComponent<AudioListener>().enabled
= true;
            }
        }
    }

}
```

ScreenTexture.cs

```
using UnityEngine;
using UnityEngine.UI;
using System.Collections;
/* ----------------------------------------
 * class to demonstrate how to take snapshots
 * of the screen and use it as a GUI texture
 */
public class ScreenTexture : MonoBehaviour
{
	// Gameobject variable for the GUI object where to display the
texture
	public GameObject photoGUI;
	// Gameobject variable for the GUI object to be used as frame
	public GameObject frameGUI;
	// Float variable for the ratio between size of the snapshot
and displayed texture
	public float ratio = 0.25f;

	/* ----------------------------------------
	 * During Update, detect if the left mouse button was pressed,
	 * starting the CaptureScreen() coroutine, if so.
	 */
	void Update()
	{
		if (Input.GetKeyUp(KeyCode.Mouse0))
			// IF the left mouse button was pressed, THEN start
the CaptureScreen coroutine
			StartCoroutine(CaptureScreen());
	}

	/* ----------------------------------------
	 * A function to calculate the dimension and location of the
snapshot,
	 * capture it and apply it to its respective GUI element
	 */
	IEnumerator CaptureScreen()
	{
		// Disable GUI element for the last snapshot taken
(otherwise it will be superposed to the next snapshot)
		photoGUI.SetActive(false);

		// A shorthand for the screen's width
		int sw = Screen.width;

		// A shorthand for the screen's height
		int sh = Screen.height;

		// A shorthand for the Rect Transform settings of the GUI
element for the framing
		RectTransform frameTransform =
frameGUI.GetComponent<RectTransform>();
```

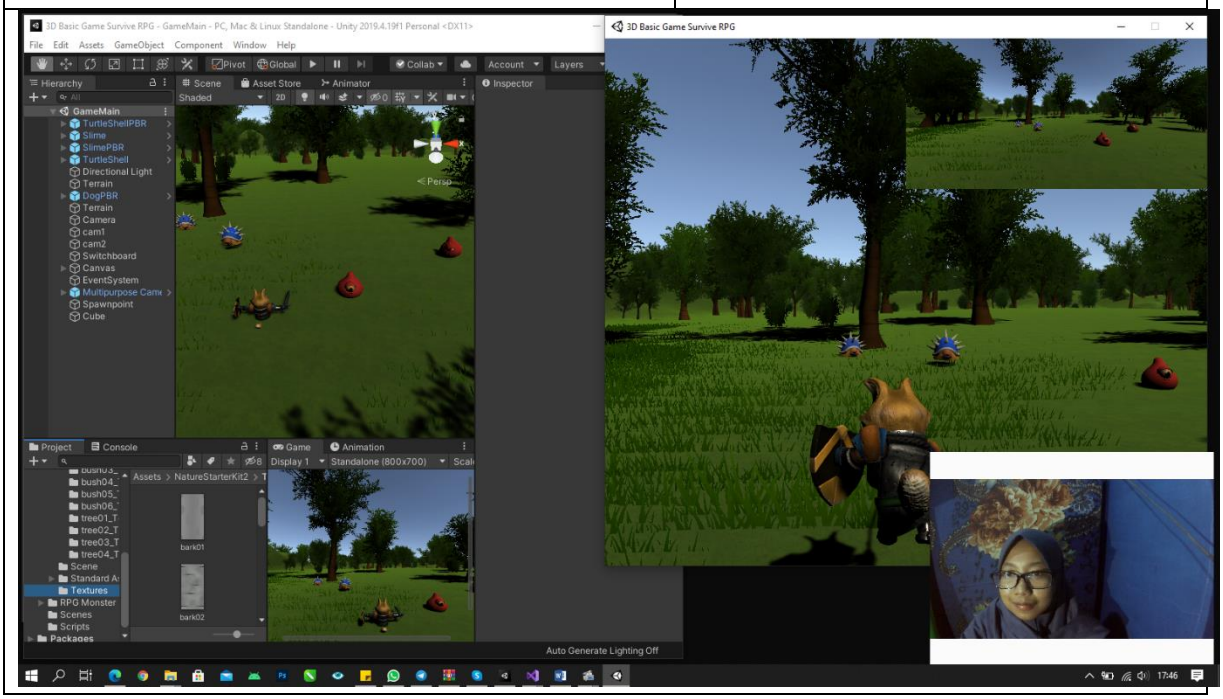
```
            // Rect for the snapshot area, initially based on the GUI
frame's the Rect Transform
            Rect framing = frameTransform.rect;

            // A shorthand for the coordinates of the GUI frame's
pivot
            Vector2 pivot = frameTransform.pivot;

            // A 2D vector for the Anchor Min (defines horizontal and
vertical origin of the frame)
            Vector2 origin = frameTransform.anchorMin;

            // Convert X coordinate of origin point to pixels by
multiplying it by screen's width
            origin.x *= sw;

            // Convert Y coordinate of origin point to pixels by
multiplying it by screen's height
            origin.y *= sh;

            // float var for horizontal offset of the frame, obtained
by multiplying horizontal pivot point by frame width
            float xOffset = pivot.x * framing.width;

            // Add horizontal offset to frame horizontal origin
            origin.x += xOffset;

            // float var for vertical offset of the frame, obtained
by multiplying vertical pivot point by frame height
            float yOffset = pivot.y * framing.height;

            // Add vertical offset to frame vertical origin
            origin.y += yOffset;

            // Offset framing horizontal location
            framing.x += origin.x;

            // Offset framing vertical location
            framing.y += origin.y;

            // Int variable for texture width, based on framing width
            int textWidth = (int)framing.width;

            // Int variable for texture height, based on framing
height
            int textHeight = (int)framing.height;

            // Create a new Texture measuring textWidth x textHeight
            Texture2D texture = new Texture2D(textWidth, textHeight);

            // Wait for the EndOfFrame before capturing snapshot
            yield return new WaitForEndOfFrame();
```

```
            //Read Pixels from screen
            texture.ReadPixels(framing, 0, 0);

            // Apply captured pixels onto texture
            texture.Apply();

            // Re-activate GUI element for displaying snapshot
            photoGUI.SetActive(true);

            // 3D Vector for the new snapshot dimension (based on
framing dimension multplied by selected ratio)
            Vector3 photoScale = new Vector3(framing.width * ratio,
framing.height * ratio, 1);

            // Resize GUI texture display to specified dimensions
            photoGUI.GetComponent<RectTransform>().localScale =
photoScale;

            // Set captured texture as GUI display's texture
            photoGUI.GetComponent<RawImage>().texture = texture;
      }
}
```

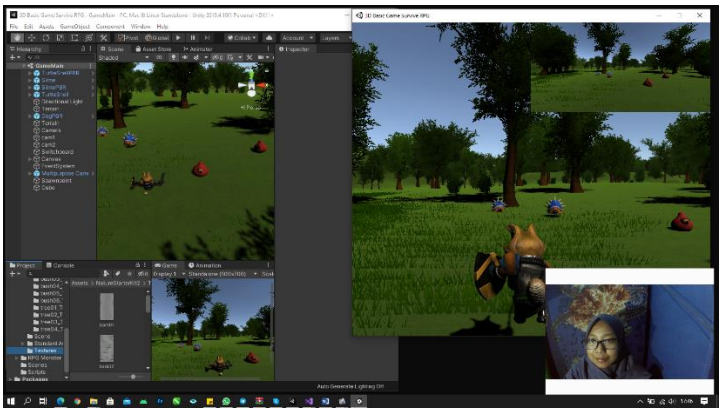
KESIMPULAN

Pada pembelajaran kali ini, kita mengetahui cara membuat 3 fitur camera yaitu camera dari arah sudut pandang lain di dalam camera, Perpindahan Beberapa Sudut Pandang Camera, dan Membuat Hasil Tangkapan (Capture) Layar yang Berada di Dalam Frame.

SWAFOTO MAHASISWA+PROJECT



Format laporan:

NIM	:	1841720004
Nama	:	Bella Setyowati
Kelas	:	TI-3H
Swafoto	:	

Deskripsi Game :

a. Nama Game : Survival Dog

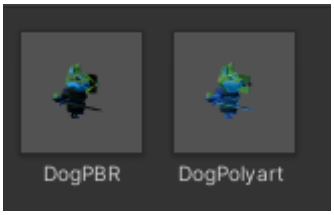
b. Alur Game (print screen dan penjelasan) : Player akan menyerang moinster yang berada di hutan.

c. Komponen materi yang dipakai (disertakan screenshoot) : Object beruapa character

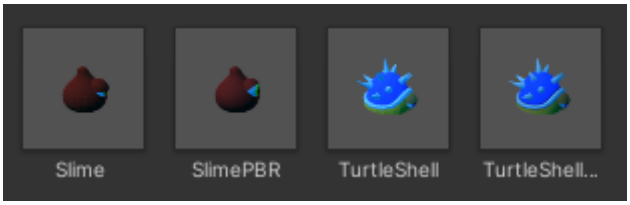



d. Asset yang dipakai (sertakan screenshoot, jika terdapat asset yang digunakan berasal dari internet atau sumber lain, cantumkan link) :

Dog Knight: Dog Knight PBR Polyart | 3D Animals | Unity Asset Store



Monster: RPG Monster Duo PBR Polyart | 3D Creatures | Unity Asset Store



Nature: Nature Starter Kit 2 | 3D Environments | Unity Asset Store