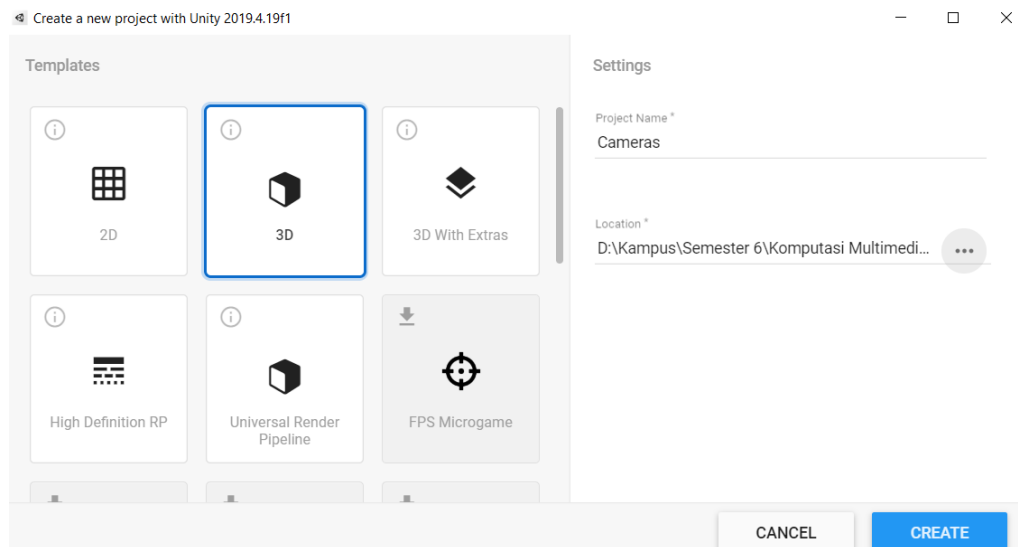
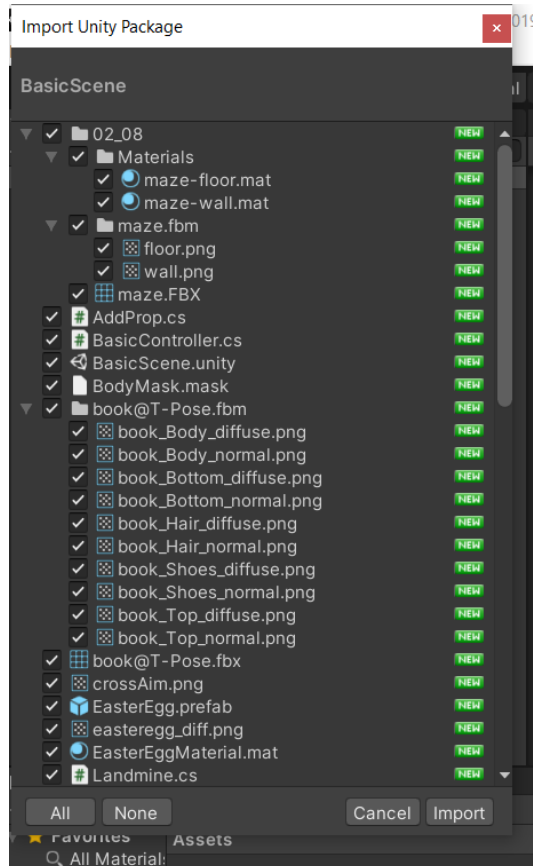
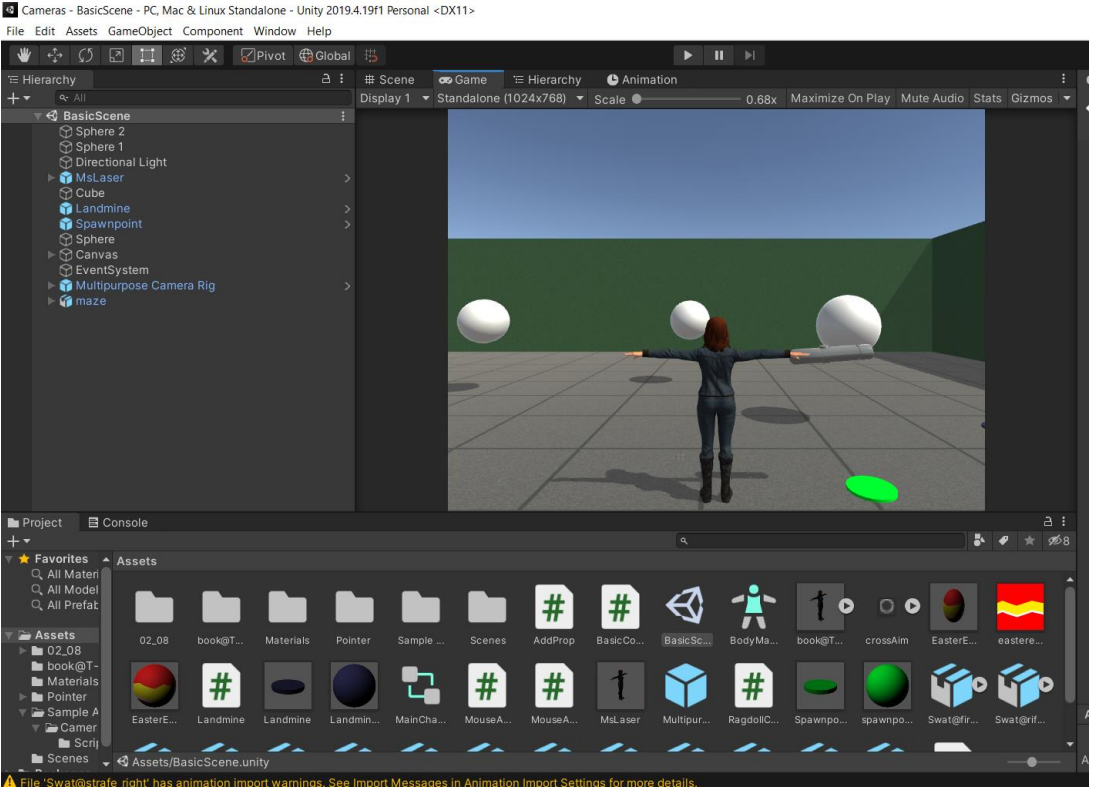
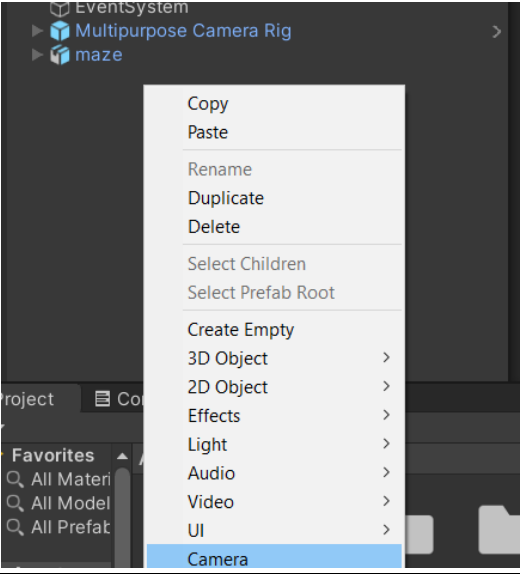
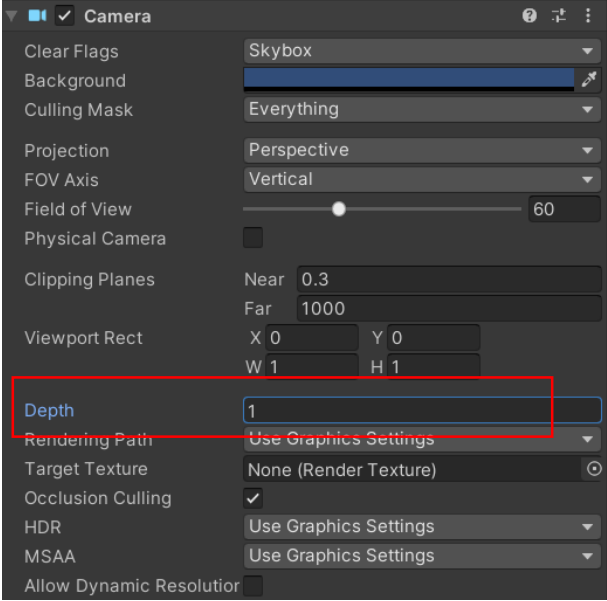


Pertemuan Ke- 6
Laporan

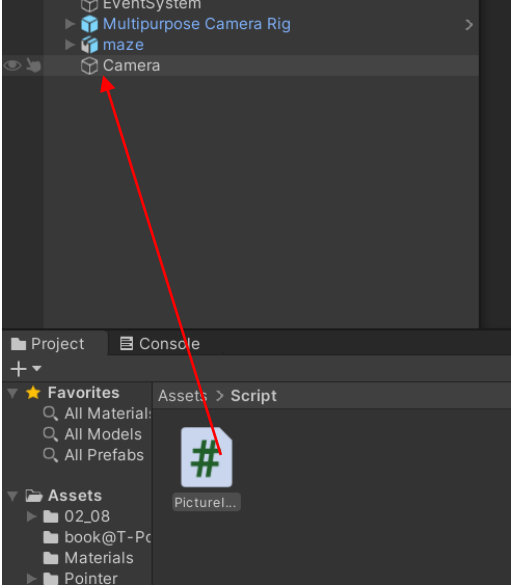
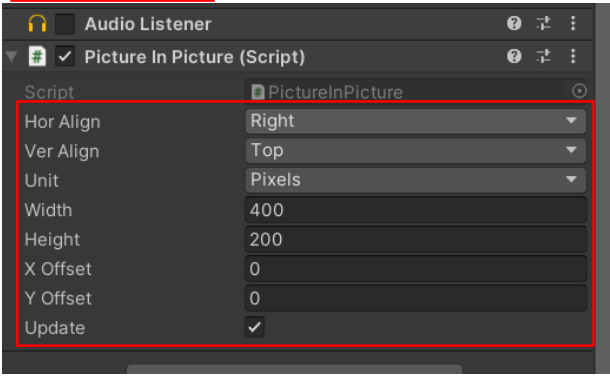
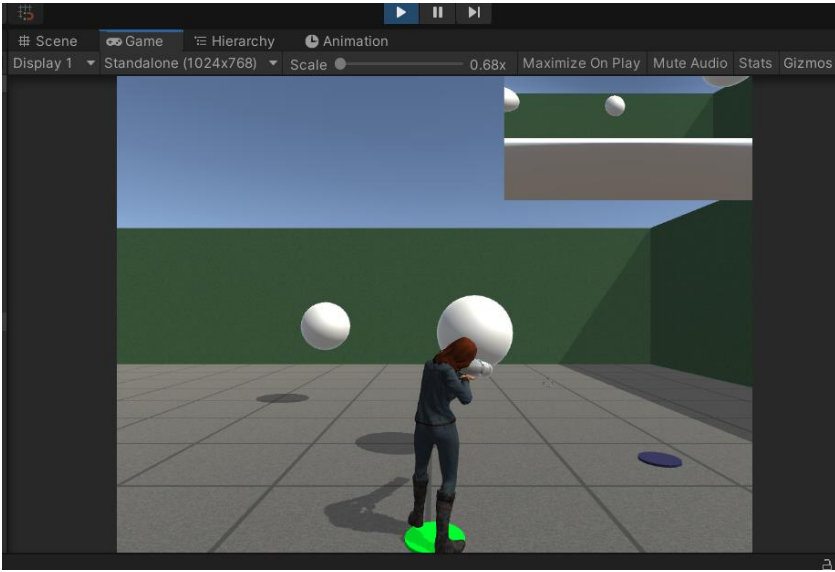
LANGKAH-LANGKAH Pengerjaan**1. Buatlah Project 3D baru pada Unity.****2. Masukkan paket BasicScene kedalam Assets.****3. Pada Project View, buka BasicScene kemudian akan muncul karakter animasi dan beberapa objek benda yang telah dibuat.**

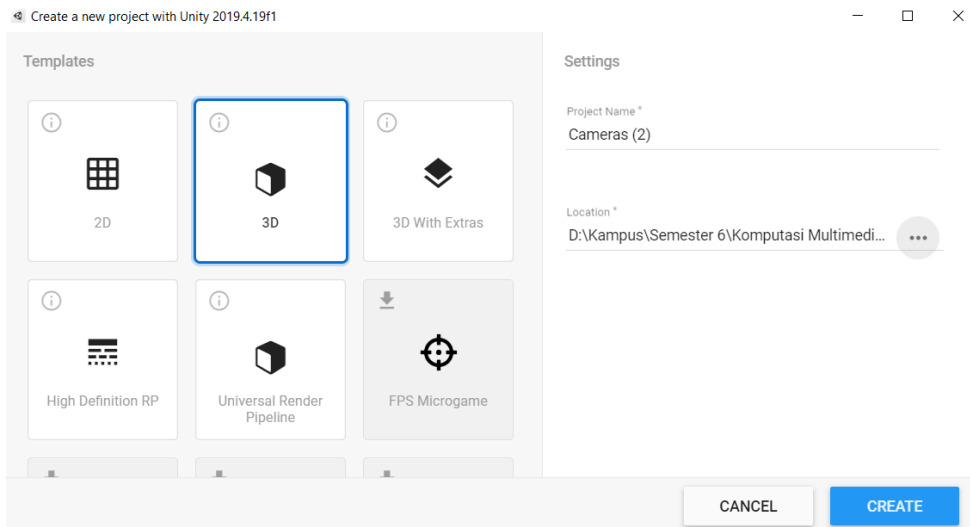
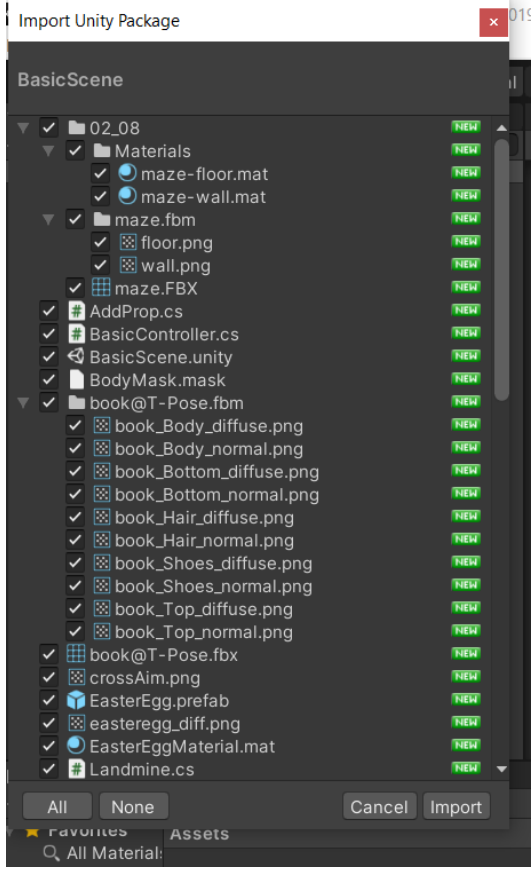
	
<p>4.</p>	<p>Tambahkan new Camera ke dalam scene melalui menu Create yang berada diatas Hierarchy, Pilih Create → Camera</p> 
<p>5.</p>	<p>Pilih Camera pada Hierarchy yang telah dibuat, dari Inspector rubahlah “Depth = 1” seperti gambar di bawah ini.</p>

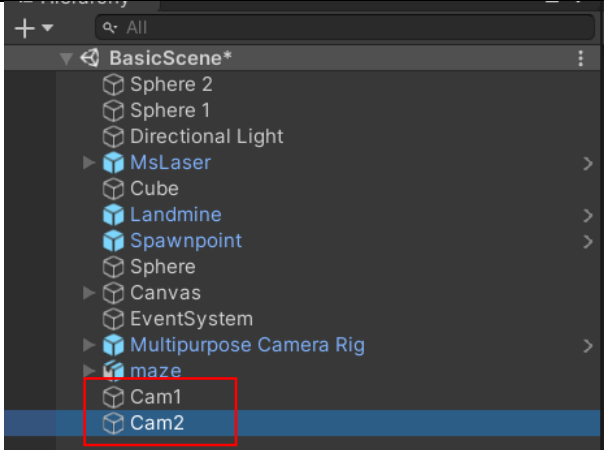
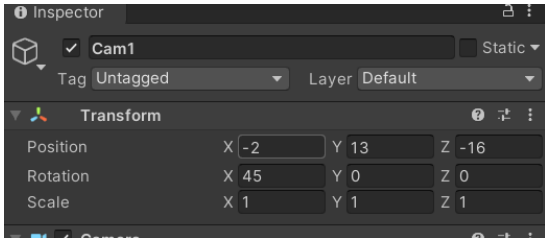
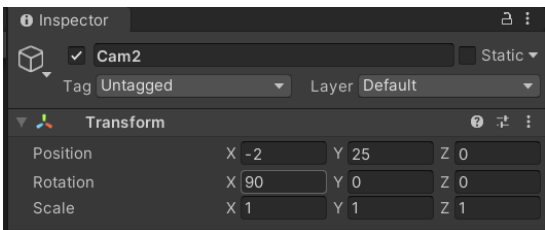
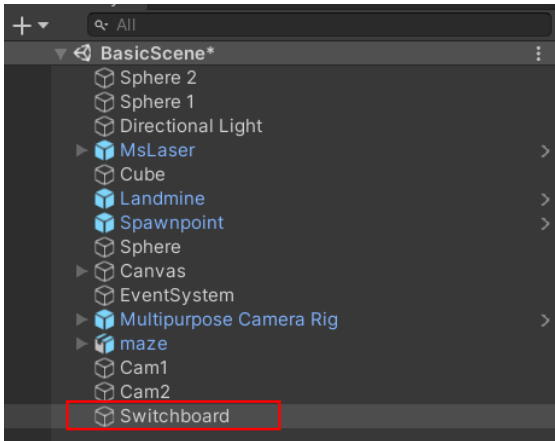
		
6.	<p>Pada Project View buatlah script C# baru dengan code seperti dibawah ini, dan berilah nama PictureInPicture.</p> <pre> PictureInPicture.cs Miscellaneous Files 1 using System.Collections; 2 using System.Collections.Generic; 3 using UnityEngine; 4 5 public class PictureInPicture : MonoBehaviour 6 { 7 public enum hAlignment { left, center, right }; 8 public enum vAlignment { top, middle, bottom }; 9 public hAlignment horAlign = hAlignment.left; 10 public vAlignment verAlign = vAlignment.top; 11 public enum UnitsIn { pixels, screen_percentage }; 12 public UnitsIn unit = UnitsIn.pixels; 13 public int width = 50; 14 public int height = 50; 15 public int xOffset = 0; 16 public int yOffset = 0; 17 public bool update = true; 18 private int hsize, vsize, hloc, vloc; 19 20 void Start() 21 { 22 AdjustCamera(); 23 } 24 void Update() 25 { 26 if (update) 27 AdjustCamera(); 28 } 29 </pre>	

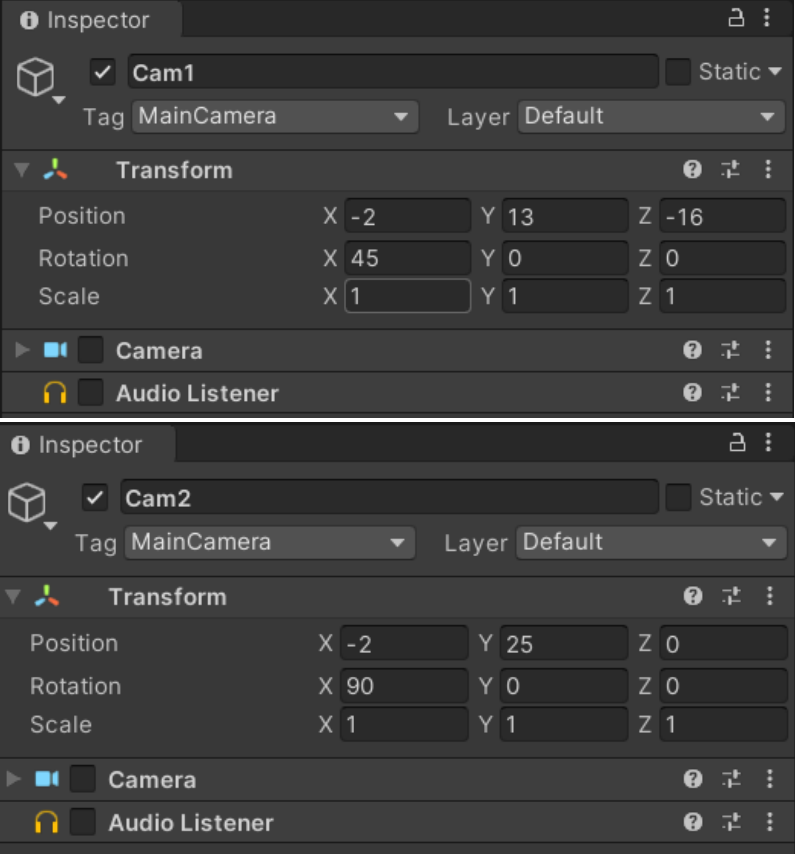
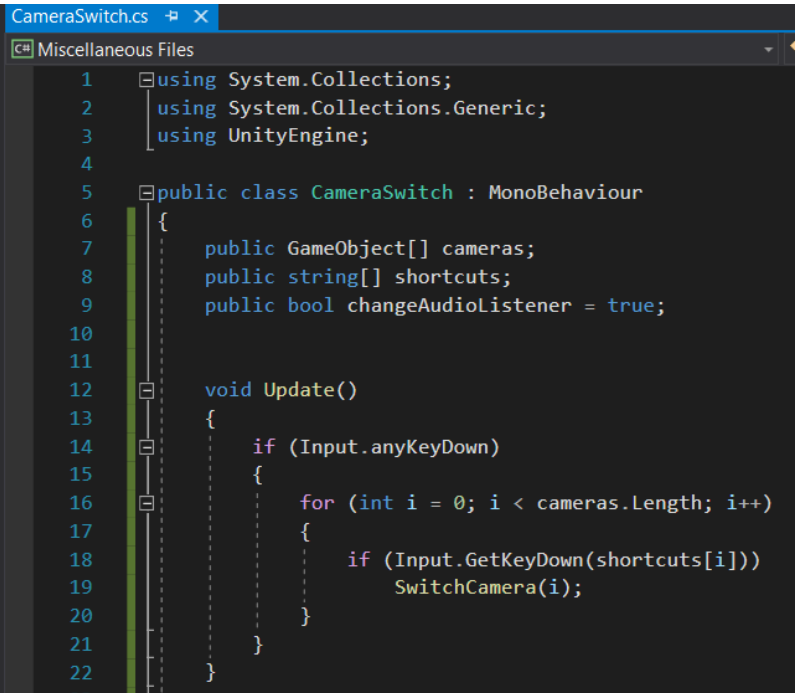
	<pre> 30 void AdjustCamera() 31 { 32 int sw = Screen.width; 33 int sh = Screen.height; 34 float swPercent = sw * 0.01f; 35 float shPercent = sh * 0.01f; 36 float xOffPercent = xOffset * swPercent; 37 float yOffPercent = yOffset * shPercent; 38 int xOff; 39 int yOff; 40 if (unit == UnitsIn.screen_percentage) 41 { 42 hsize = width * (int)swPercent; 43 vsize = height * (int)shPercent; 44 xOff = (int)xOffPercent; 45 yOff = (int)yOffPercent; 46 } 47 else 48 { 49 hsize = width; 50 vsize = height; 51 xOff = xOffset; 52 yOff = yOffset; 53 } </pre>	
7.	<pre> 54 switch (horAlign) 55 { 56 case hAlignment.left: 57 hloc = xOff; 58 break; 59 case hAlignment.right: 60 int justifiedRight = (sw - hsize); 61 hloc = (justifiedRight - xOff); 62 break; 63 case hAlignment.center: 64 float justifiedCenter = (sw * 0.5f) - (hsize * 0.5f); 65 hloc = (int)(justifiedCenter - xOff); 66 break; 67 } 68 switch (verAlign) 69 { 70 case vAlignment.top: 71 int justifiedTop = sh - vsize; 72 vloc = (justifiedTop - (yOff)); 73 break; 74 case vAlignment.bottom: 75 vloc = yOff; 76 break; 77 case vAlignment.middle: 78 float justifiedMiddle = (sh * 0.5f) - (vsize * 0.5f); 79 vloc = (int)(justifiedMiddle - yOff); 80 break; 81 } 82 GetComponent<Camera>().pixelRect = new Rect(hloc, vloc, hsize, vsize); 83 } 84 } </pre>	

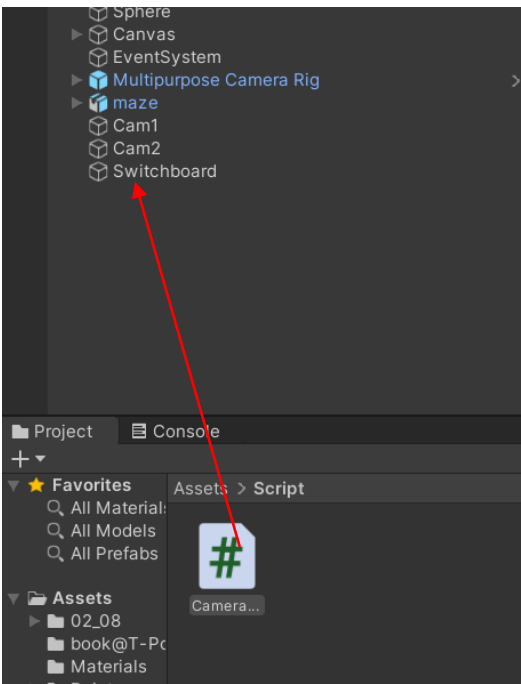
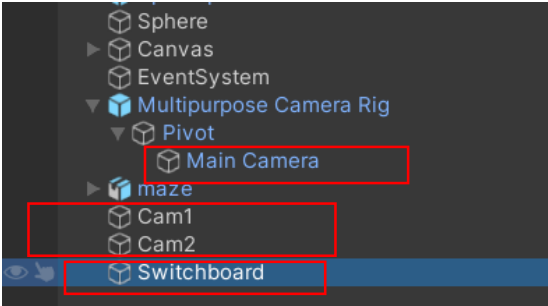
Setelah menyimpan script C# tersebut, drag script tersebut menuju Camera yang ada pada Hierarchy.

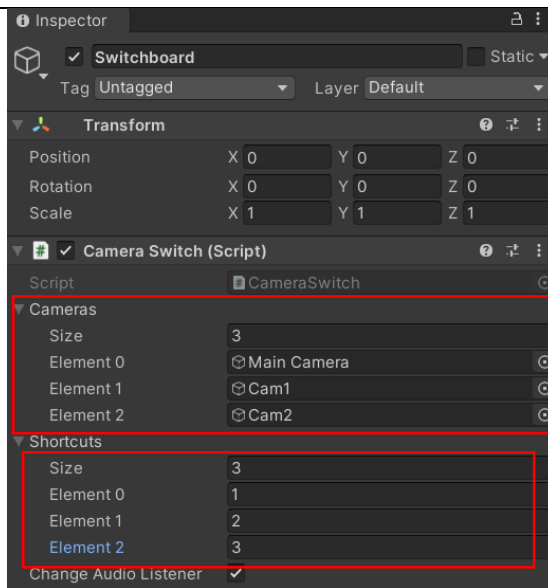
	
<p>8.</p>	<p>Pilih Camera lalu pada Inspector hilangkan centang pada komponen Audio Listener dan rubah beberapa parameter yang ada pada Script PictureInPicture seperti gambar dibawah ini:</p> 
<p>9.</p>	<p>Silahkan jalankan program, maka pada layar tampilan akan muncul camera dari arah sudut pandang berbeda yang terletak pada kanan atas layar</p> 
	<p>2. Membuat dan Melakukan Perpindahan Beberapa Sudut Pandang Camera</p>

1.	<p>Buatlah Project 3D baru pada Unity 3D.</p> 
2.	<p>Masukkan paket BasicScene kedalam Assets</p> 
3.	<p>Pada Project View, buka BasicScene kemudian akan muncul karakter animasi dan beberapa objek benda yang telah dibuat.</p>
4.	<p>Tambahkan 2 new Camera ke dalam scene melalui menu Create yang berada diatas Hierarchy, Pilih Create → Camera. Kemudian beri nama cam1 dan cam2.</p>

	
5.	<p>Pada Inspector, rubahlah posisi camera pada cam1 dan cam2 seperti dibawah ini.</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Cam 1</p>  </div> <div style="text-align: center;"> <p>Cam 2</p>  </div> </div>
6.	<p>Buatlah GameObject baru menggunakan menu Create pada Hierarchy. Pilih Create → Create Empty. Kemudian ganti nama menjadi Switchboard.</p> 
7.	<p>Dari Inspector pada cam1 dan cam2, hilangkan tanda centang pada komponen Camera dan Audio Listener. Serta atur kolom Tag menjadi MainCamera.</p>

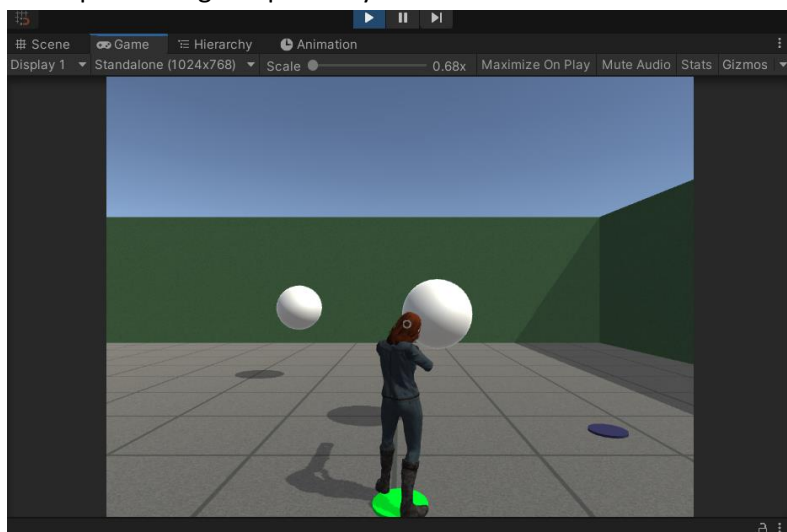
		
8.	<p>Kemudian buatlah script C# dengan code seperti dibawah ini kemudian berilah nama CameraSwitch</p>  <pre> 1 using System.Collections; 2 using System.Collections.Generic; 3 using UnityEngine; 4 5 public class CameraSwitch : MonoBehaviour 6 { 7 public GameObject[] cameras; 8 public string[] shortcuts; 9 public bool changeAudioListener = true; 10 11 12 void Update() 13 { 14 if (Input.anyKeyDown) 15 { 16 for (int i = 0; i < cameras.Length; i++) 17 { 18 if (Input.GetKeyDown(shortcuts[i])) 19 SwitchCamera(i); 20 } 21 } 22 } 23 </pre>	

	<pre> 25 void SwitchCamera(int index) 26 { 27 for (int i = 0; i < cameras.Length; i++) 28 { 29 if (i != index) 30 { 31 cameras[i].GetComponent<Camera>().enabled = false; 32 if (changeAudioListener) 33 cameras[i].GetComponent<AudioListener>().enabled = false; 34 } 35 else 36 { 37 cameras[i].GetComponent<Camera>().enabled = true; 38 if (changeAudioListener) 39 cameras[i].GetComponent<AudioListener>().enabled = true; 40 } 41 } 42 } </pre>	
9.	<p>Setelah itu drag script C# tersebut menuju GameObject Switchboard.</p> 	
10.	<p>Pada Inspector, atur ukuran Cameras dan Shortcuts dengan “sizes = 3”. Kemudian, drag dan arahkan secara urut mulai dari Main Camera (berada di Multipurpose Camera Rig → Pivot), cam1, dan cam2 menuju ke setiap slots yang ada pada Cameras. Kemudian pada Shortcuts ketikkan angka 1, 2, dan 3 secara urut pada slots yang tersedia. Seperti gambar di bawah ini:</p> 	

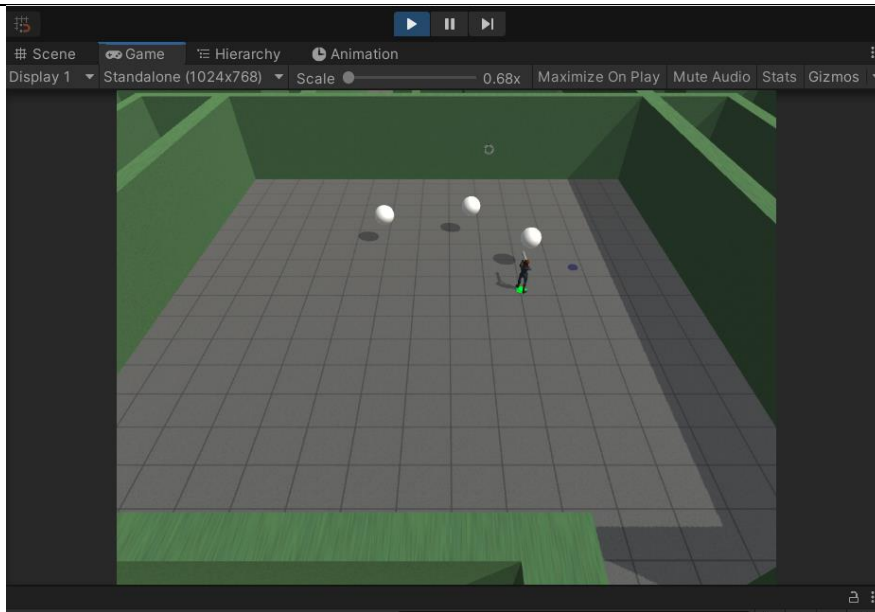


11. Jalankan program dan coba tekan angka 1, 2, dan 3 pada keyboard, maka akan terjadi perubahan pada sudut pandang camera utama.

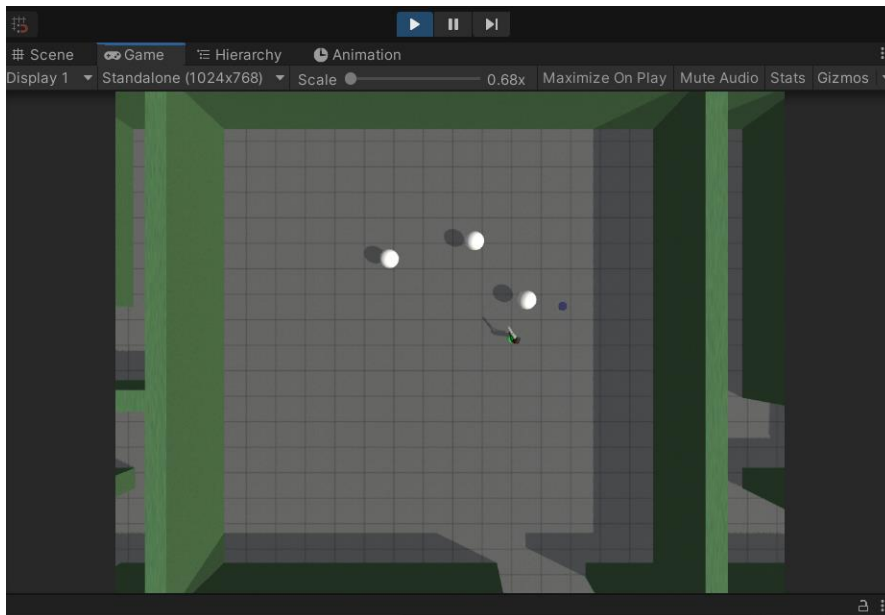
Jika dipencet angka 1 pada keyboard



Jika dipencet angka 2 pada keyboard

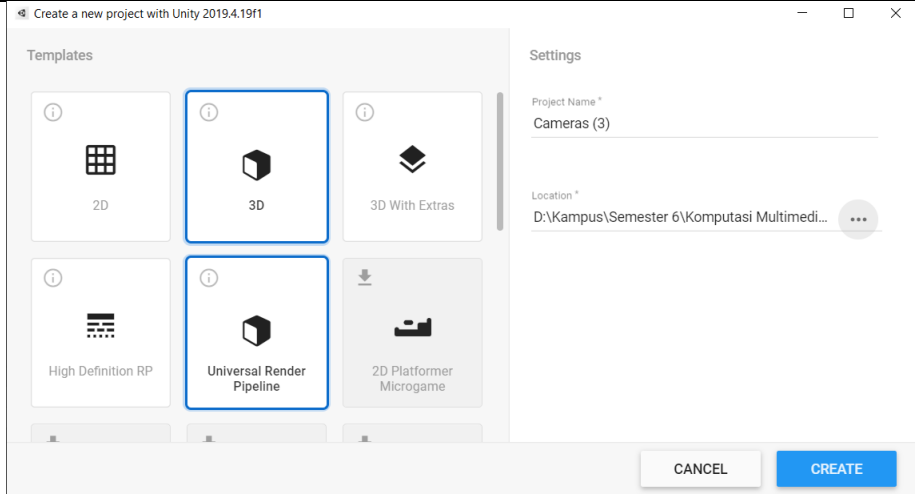
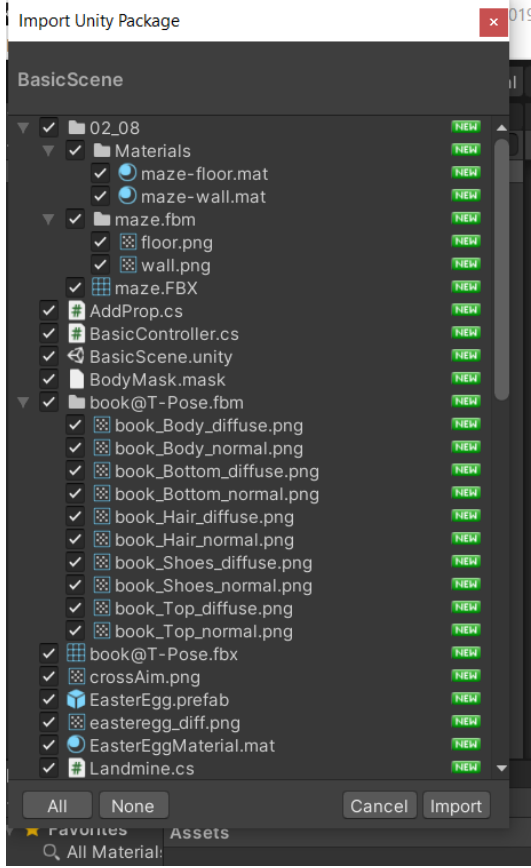


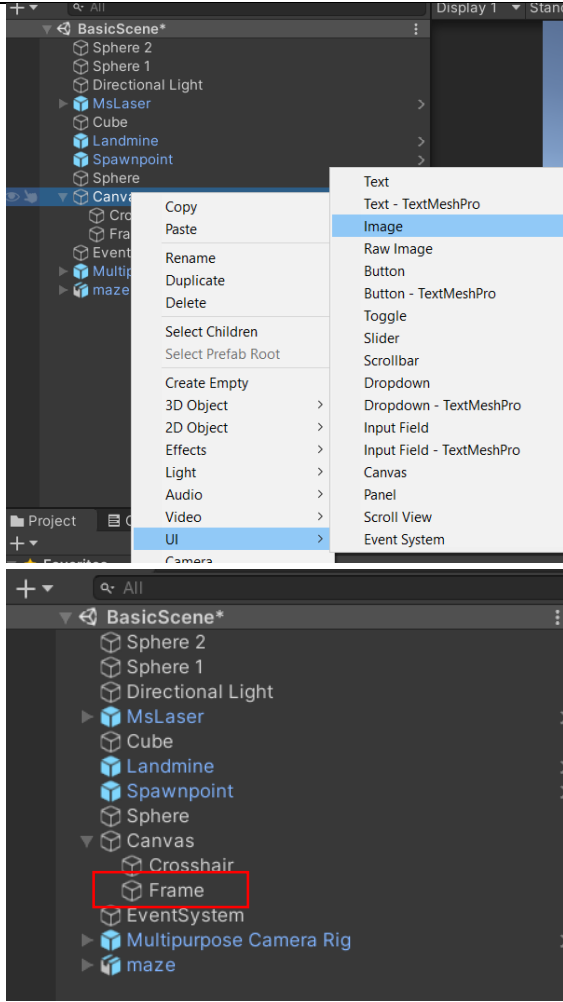
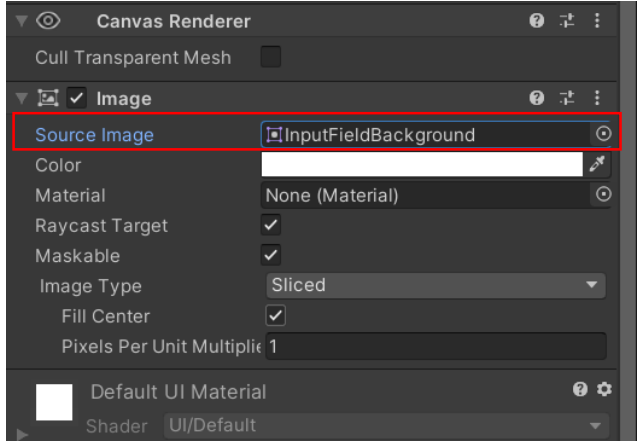
Jika dipencet angka 3 pada keyboard

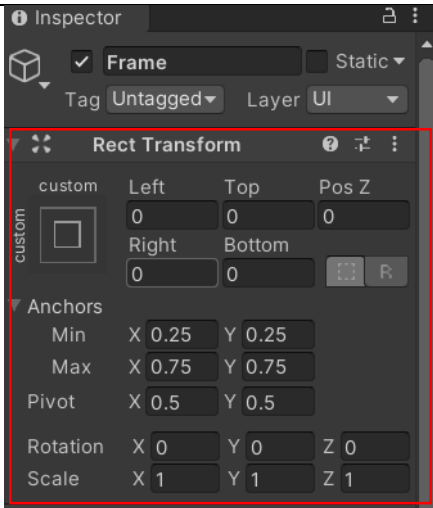

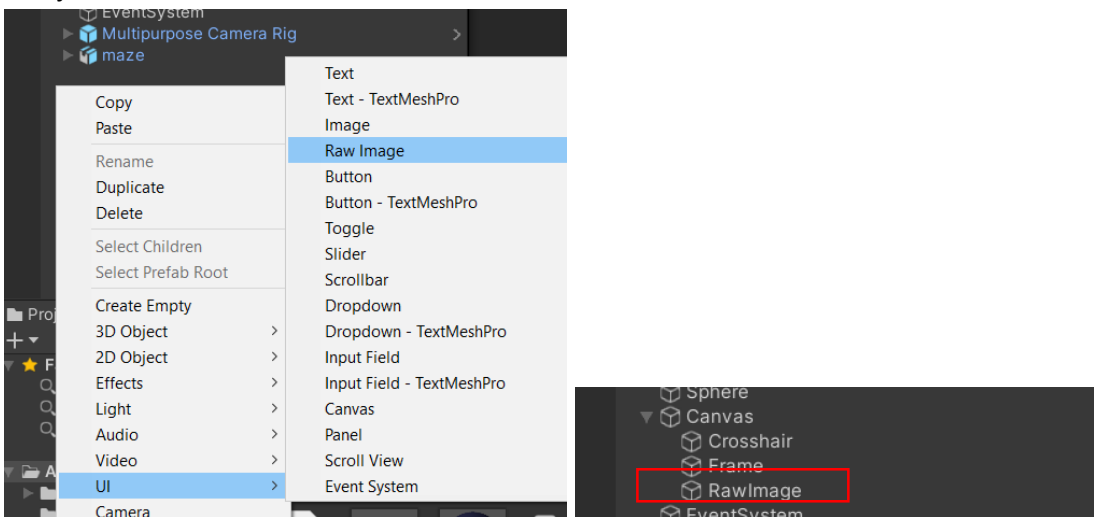
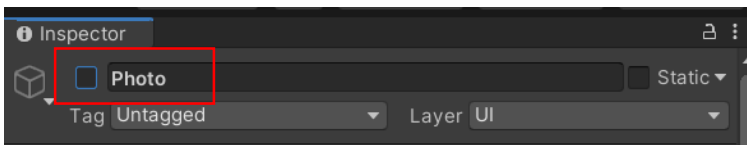


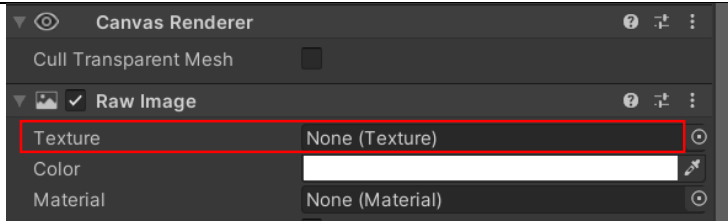
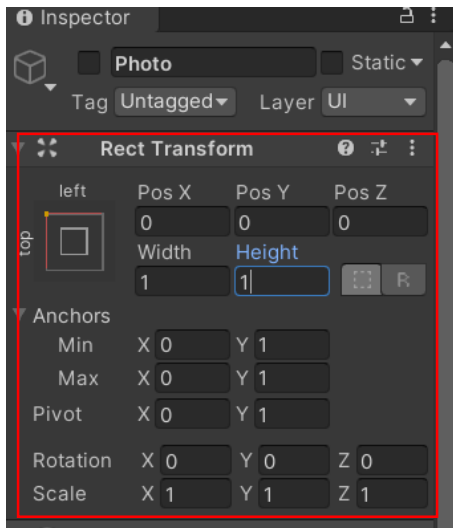
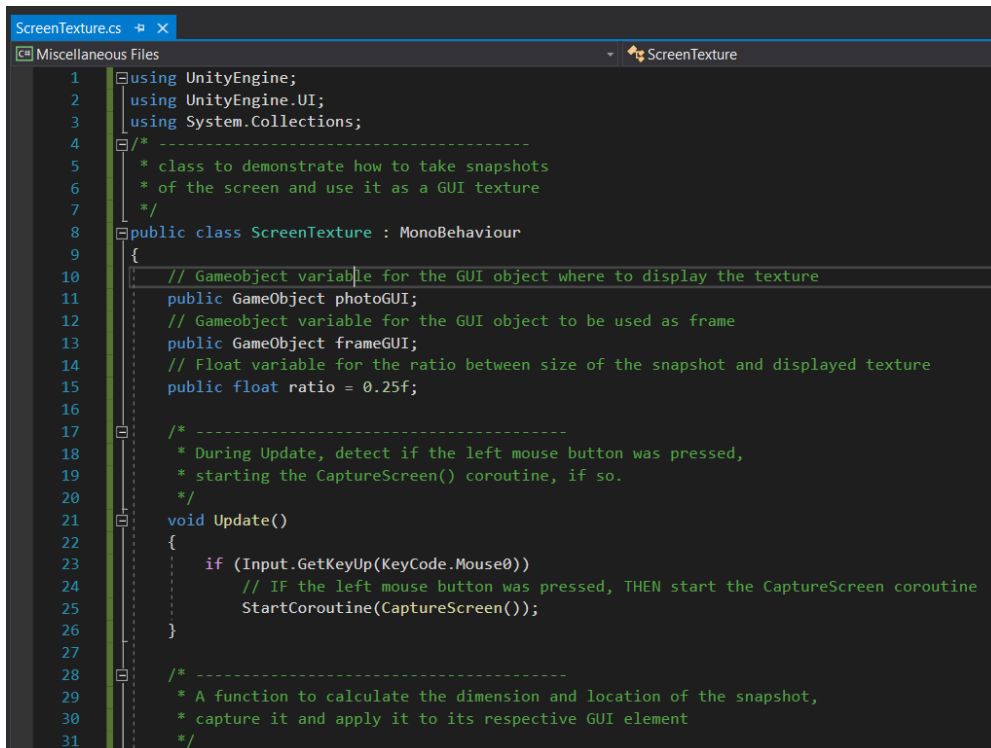
3. Membuat Hasil Tangkapan (Capture) Layar yang Berada di Dalam Frame

1. Buatlah Project 3D baru pada Unity 3D

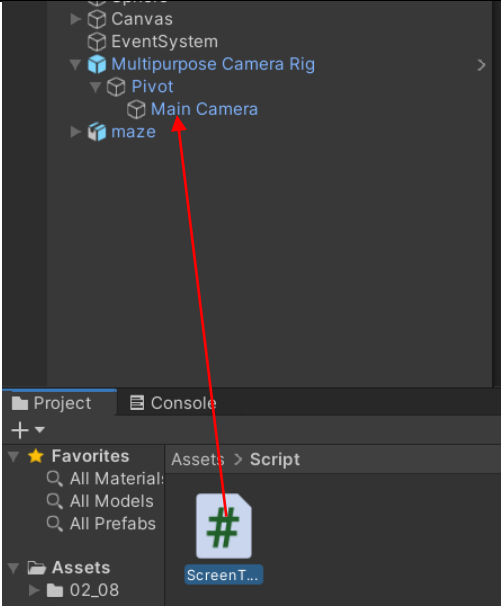
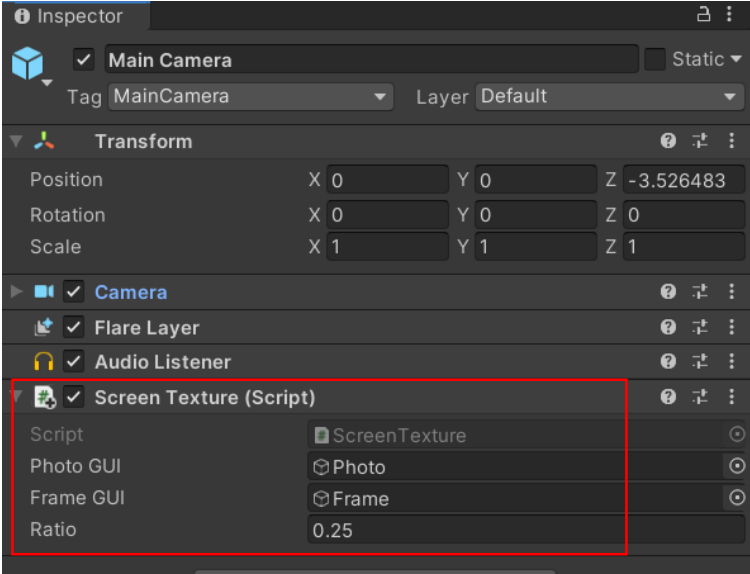
	
2.	<p>Masukkan paket BasicScene ke dalam Assets.</p> 
3.	<p>Pada Project View, buka BasicScene kemudian akan muncul karakter animasi dan beberapa objek benda yang telah dibuat. Didalam paket tersebut juga sudah terdapat fitur Canvas untuk UI elements.</p>
4.	<p>Buatlah UI Image dari menu Create yang ada pada Hierarchy, Pilih Create → UI → Image. Setelah itu akan muncul Image yang berada pada Canvas kemudian ganti nama tersebut menjadi frame.</p>

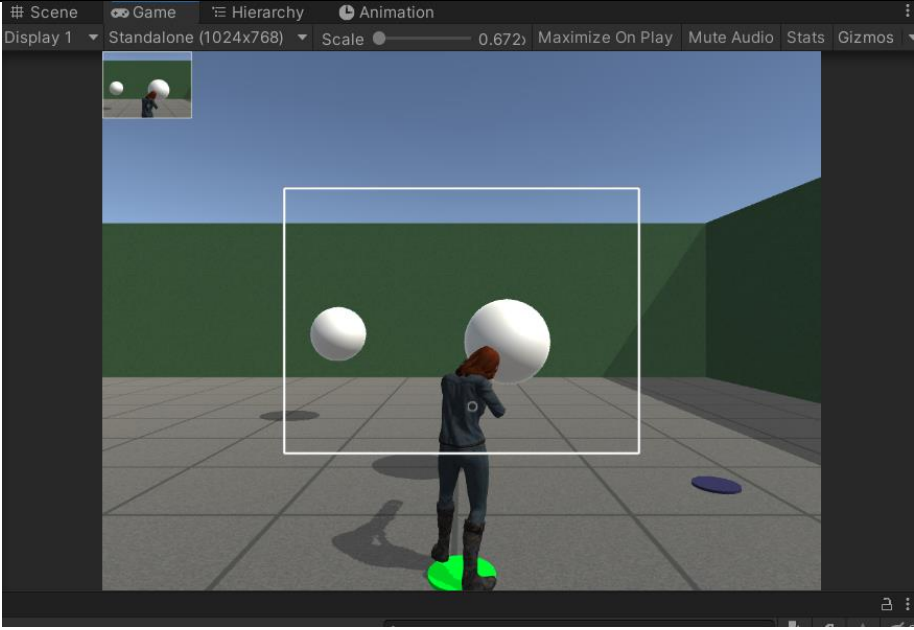
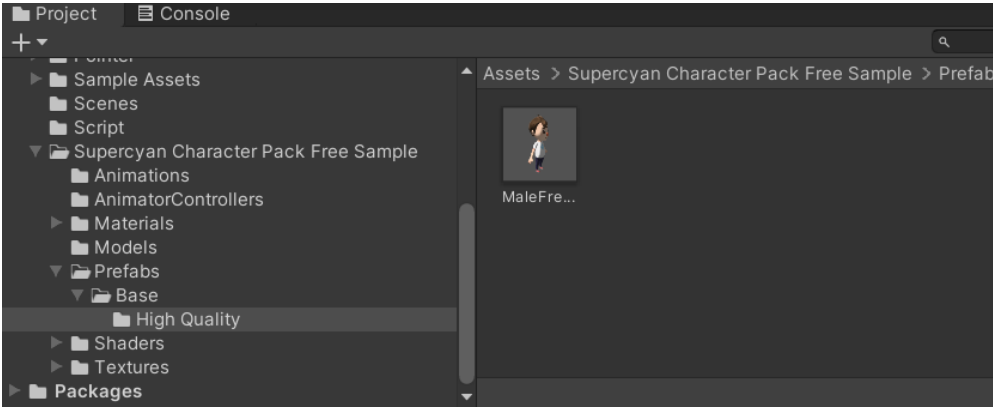
	
5.	<p>Pilih frame pada Hierarchy, kemudian pada komponen Image (Script) yang berada di Inspector atur kolom Source Image menjadi InputFieldBackground. Seperti pada gambar di bawah ini:</p> 
6.	<p>Pada Inspector milik Frame, Rubahlah Rect Transform sesuai dengan gambar dibawah ini:</p>

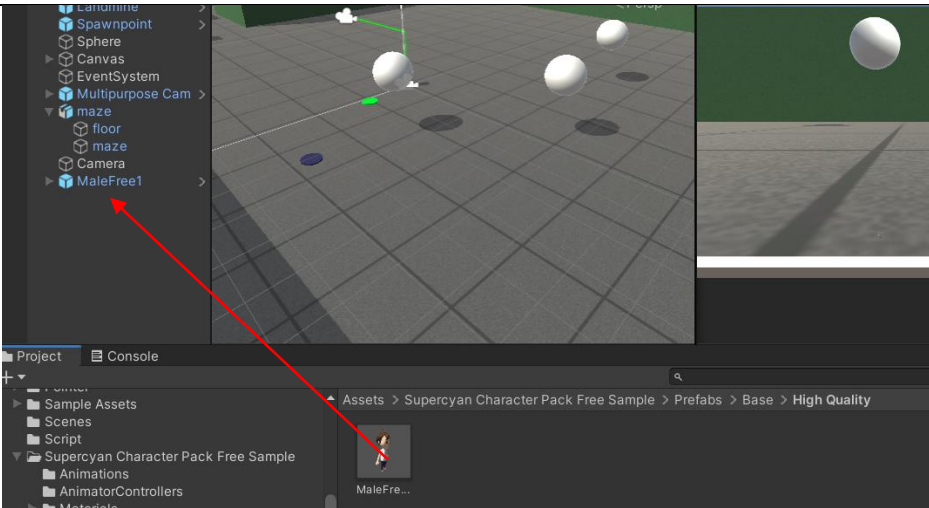
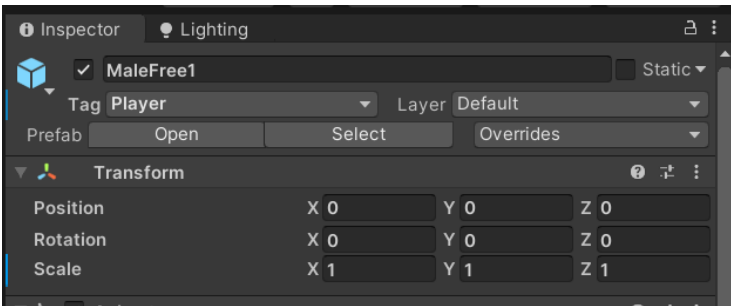
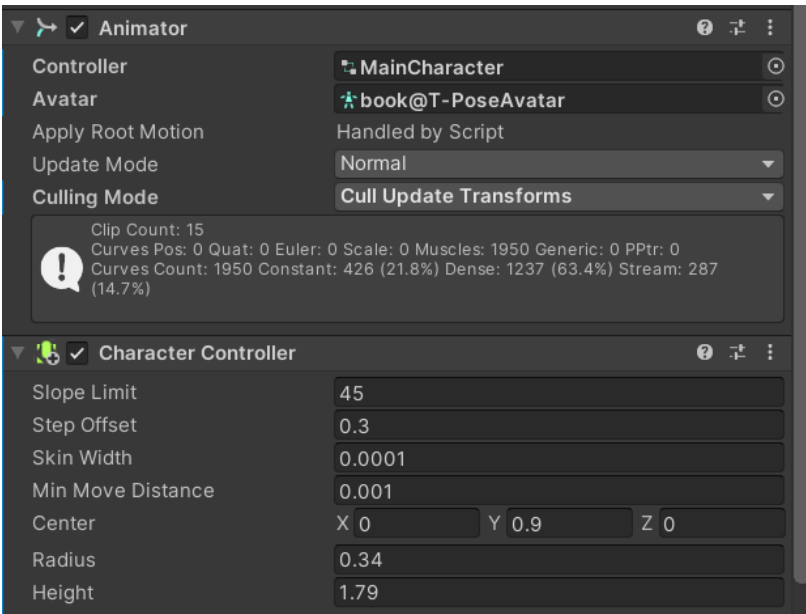
	
7.	<p>Pada komponen Image (Script), hilangkan centang pada Fill Center.</p> 
8.	<p>Buatlah UI Raw Image dari menu Create yang terdapat pada Hierarchy, Pilih Create → UI → RawImage. Setelah itu akan muncul pada Canvas GameObject kemudian ganti nama menjadi Photo.</p> 
9.	<p>Pada Hierarchy pilih Photo, kemudian pada Inspector carilah komponen Raw Image (Script) dan atur kolom Texture menjadi None (Texture). Setelah itu hilangkan centang pada kolom Photo yang berada di atas.</p> 

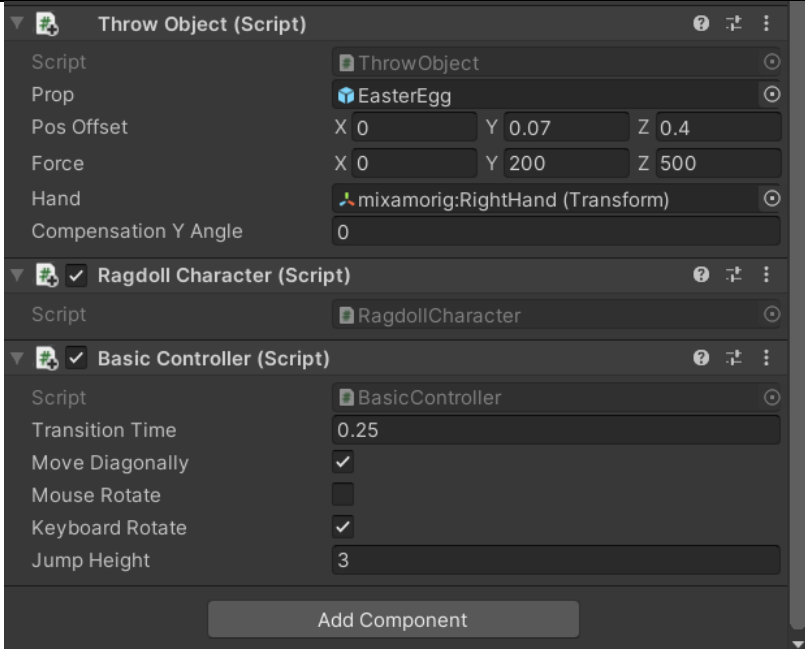
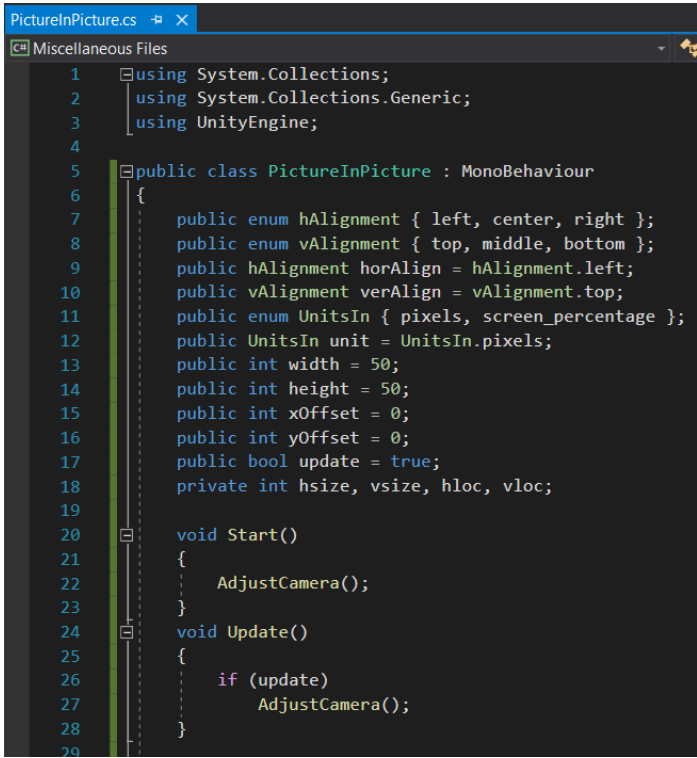
	
10.	<p>Langkah selanjutnya atur Rect Transform pada Photo sesuai gambar dibawah ini:</p> 
11.	<p>Kemudian buatlah script C# dengan code seperti dibawah ini kemudian berilah nama ScreenTexture.</p>  <pre> 1 using UnityEngine; 2 using UnityEngine.UI; 3 using System.Collections; 4 /* ----- 5 * class to demonstrate how to take snapshots 6 * of the screen and use it as a GUI texture 7 */ 8 public class ScreenTexture : MonoBehaviour 9 { 10 // GameObject variable for the GUI object where to display the texture 11 public GameObject photoGUI; 12 // GameObject variable for the GUI object to be used as frame 13 public GameObject frameGUI; 14 // Float variable for the ratio between size of the snapshot and displayed texture 15 public float ratio = 0.25f; 16 17 /* ----- 18 * During Update, detect if the left mouse button was pressed, 19 * starting the CaptureScreen() coroutine, if so. 20 */ 21 void Update() 22 { 23 if (Input.GetKeyUp(KeyCode.Mouse0)) 24 // IF the left mouse button was pressed, THEN start the CaptureScreen coroutine 25 StartCoroutine(CaptureScreen()); 26 } 27 28 /* ----- 29 * A function to calculate the dimension and location of the snapshot, 30 * capture it and apply it to its respective GUI element 31 */ </pre>

	<pre> 32 IEnumerator CaptureScreen() 33 { 34 // Disable GUI element for the last snapshot taken (otherwise it will be superposed to the next snapshot) 35 photoGUI.SetActive(false); 36 37 // A shorthand for the screen's width 38 int sw = Screen.width; 39 40 // A shorthand for the screen's height 41 int sh = Screen.height; 42 43 // A shorthand for the Rect Transform settings of the GUI element for the framing 44 RectTransform frameTransform = frameGUI.GetComponent<RectTransform>(); 45 46 47 // Rect for the snapshot area, initially based on the GUI frame's the Rect Transform 48 Rect framing = frameTransform.rect; 49 50 // A shorthand for the coordinates of the GUI frame's pivot 51 Vector2 pivot = frameTransform.pivot; 52 53 // A 2D vector for the Anchor Min (defines horizontal and vertical origin of the frame) 54 Vector2 origin = frameTransform.anchorMin; 55 56 // Convert X coordinate of origin point to pixels by multiplying it by screen's width 57 origin.x *= sw; 58 59 // Convert Y coordinate of origin point to pixels by multiplying it by screen's height 60 origin.y *= sh; 61 62 // float var for horizontal offset of the frame, obtained by multiplying horizontal pivot point by frame width 63 float xOffset = pivot.x * framing.width; 64 65 // Add horizontal offset to frame horizontal origin 66 origin.x += xOffset; 67 68 // float var for vertical offset of the frame, obtained by multiplying vertical pivot point by frame height 69 float yOffset = pivot.y * framing.height; 70 71 // Add vertical offset to frame vertical origin 72 origin.y += yOffset; 73 74 // Offset framing horizontal location 75 framing.x += origin.x; 76 77 // Offset framing vertical location 78 framing.y += origin.y; 79 80 // Int variable for texture width, based on framing width 81 int texWidth = (int)framing.width; 82 83 // Int variable for texture height, based on framing height 84 int texHeight = (int)framing.height; 85 86 // Create a new Texture measuring texWidth x texHeight 87 Texture2D texture = new Texture2D(texWidth, texHeight); 88 89 // Wait for the EndOfFrame before capturing snapshot 90 yield return new WaitForEndOfFrame(); 91 92 // Read Pixels from screen 93 texture.ReadPixels(framing, 0, 0); 94 95 // Apply captured pixels onto texture 96 texture.Apply(); 97 98 // Apply captured pixels onto texture 99 texture.Apply(); 100 101 // Re-activate GUI element for displaying snapshot 102 photoGUI.SetActive(true); 103 104 // 3D Vector for the new snapshot dimension (based on framing dimension multiplied by selected ratio) 105 Vector3 photoScale = new Vector3(framing.width * ratio, framing.height * ratio, 1); 106 107 // Resize GUI texture display to specified dimensions 108 photoGUI.GetComponent<RectTransform>().localScale = photoScale; 109 110 // Set captured texture as GUI display's texture 111 photoGUI.GetComponent<RawImage>().texture = texture; 112 } </pre>
12.	Setelah itu drag Script C# tersebut menuju Main Camera yang berada pada Multipurpose Camera Rig -> Pivot

	
13.	<p>Pada Inspector milik Main Camera, carilah komponen Screen Texture (Script) kemudian atur kolom Photo GUI dengan memilih Photo. Sama halnya dengan kolom Frame GUI dengan memilih Frame.</p> 
14.	<p>Jalankan program, maka anda akan bisa meng-capture layar yang berada dalam frame dengan cara menekan klik kiri pada mouse. Hasil capture akan tampil pada pojok kiri atas layar.</p>

	
	<p style="text-align: center;">Tugas Praktikum</p> <p style="text-align: center;">1. Membuat tampilan Camera dari Arah Sudut Pandang lain di dalam Layar</p> <p>Buatlah sebuah project 3D yang didalamnya terdapat 3 Fitur Using Cameras pada modul diatas.</p>
1.	Buat projet baru untuk pengerjaan tugas
2.	Masukkan package dari pengerjaan praktikum sebelumnya.
3.	<p>Kita download dulu asset freepack karakter sample, lali import ke project</p> 
4.	Selanjutnya kita drag game object maleFree1 kedalam hirarki

	
5.	<p>Atur transform male free nya seperti berikut ini</p> 
6.	<p>Tambahkan beberapa komponen juga untuk menunjang pergerakan objectnya seperti pada gambar</p> 

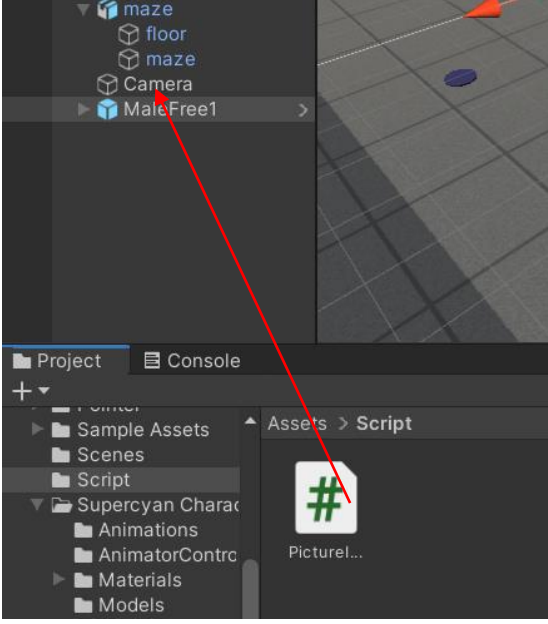
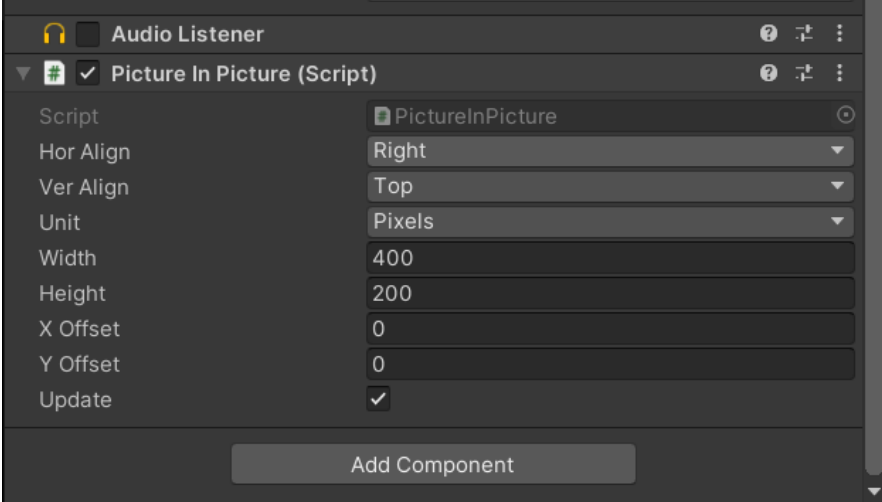
	
7.	<p>Pada Project View buatlah script C# baru dengan code seperti dibawah ini, dan berilah nama PictureInPicture.</p>  <pre> 1 using System.Collections; 2 using System.Collections.Generic; 3 using UnityEngine; 4 5 public class PictureInPicture : MonoBehaviour 6 { 7 public enum hAlignment { left, center, right }; 8 public enum vAlignment { top, middle, bottom }; 9 public hAlignment horAlign = hAlignment.left; 10 public vAlignment verAlign = vAlignment.top; 11 public enum UnitsIn { pixels, screen_percentage }; 12 public UnitsIn unit = UnitsIn.pixels; 13 public int width = 50; 14 public int height = 50; 15 public int xOffset = 0; 16 public int yOffset = 0; 17 public bool update = true; 18 private int hsize, vsize, hloc, vloc; 19 20 void Start() 21 { 22 AdjustCamera(); 23 } 24 void Update() 25 { 26 if (update) 27 AdjustCamera(); 28 } 29 </pre>

```

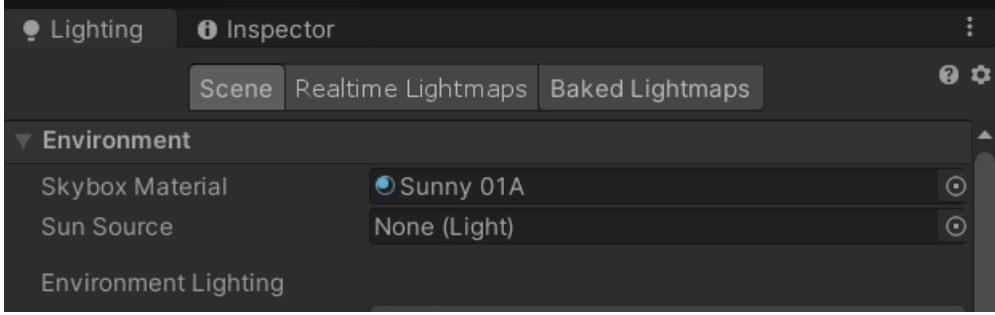


30 void AdjustCamera()
31 {
32     int sw = Screen.width;
33     int sh = Screen.height;
34     float swPercent = sw * 0.01f;
35     float shPercent = sh * 0.01f;
36     float xOffPercent = xOffset * swPercent;
37     float yOffPercent = yOffset * shPercent;
38     int xOff;
39     int yOff;
40     if (unit == UnitsIn.screen_percentage)
41     {
42         hsize = width * (int)swPercent;
43         vsize = height * (int)shPercent;
44         xOff = (int)xOffPercent;
45         yOff = (int)yOffPercent;
46     }
47     else
48     {
49         hsize = width;
50         vsize = height;
51         xOff = xOffset;
52         yOff = yOffset;
53     }
54
55     switch (horAlign)
56     {
57         case hAlignment.left:
58             hloc = xOff;
59             break;
60         case hAlignment.right:
61             int justifiedRight = (sw - hsize);
62             hloc = (justifiedRight - xOff);
63             break;
64         case hAlignment.center:
65             float justifiedCenter = (sw * 0.5f) - (hsize * 0.5f);
66             hloc = (int)(justifiedCenter - xOff);
67             break;
68     }
69     switch (verAlign)
70     {
71         case vAlignment.top:
72             int justifiedTop = sh - vsize;
73             vloc = (justifiedTop - (yOff));
74             break;
75         case vAlignment.bottom:
76             vloc = yOff;
77             break;
78         case vAlignment.middle:
79             float justifiedMiddle = (sh * 0.5f) - (vsize * 0.5f);
80             vloc = (int)(justifiedMiddle - yOff);
81             break;
82     }
83     GetComponent<Camera>().pixelRect = new Rect(hloc, vloc, hsize, vsize);
84 }

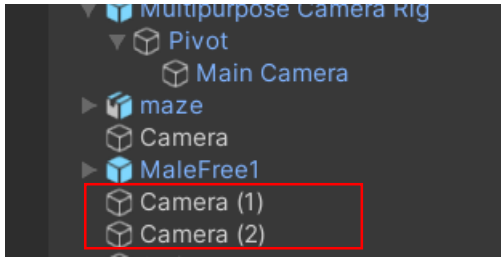
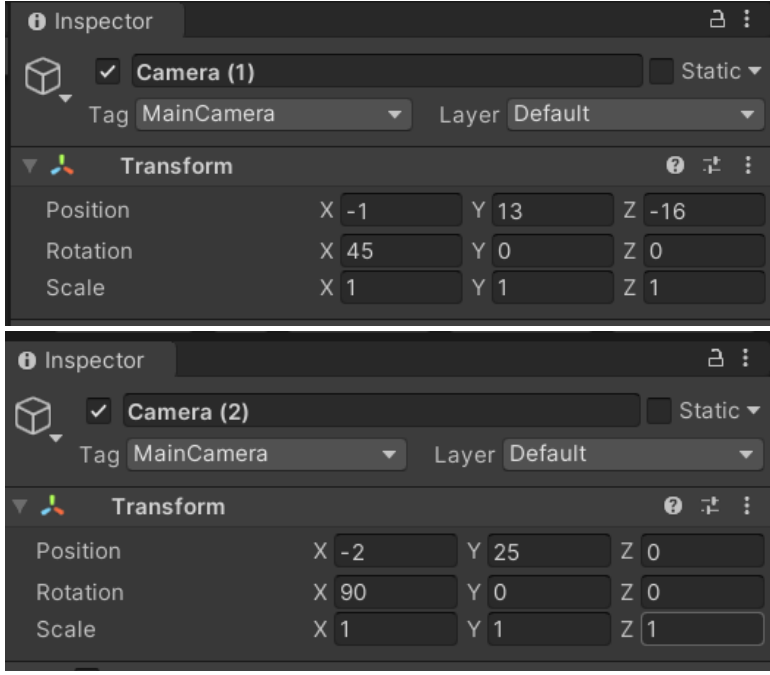
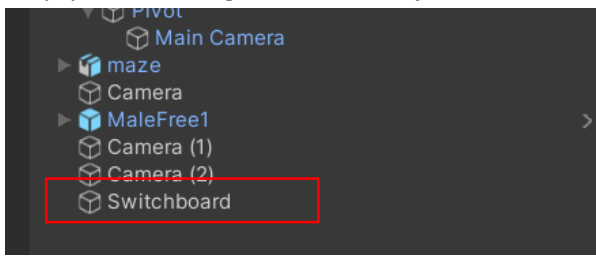
```

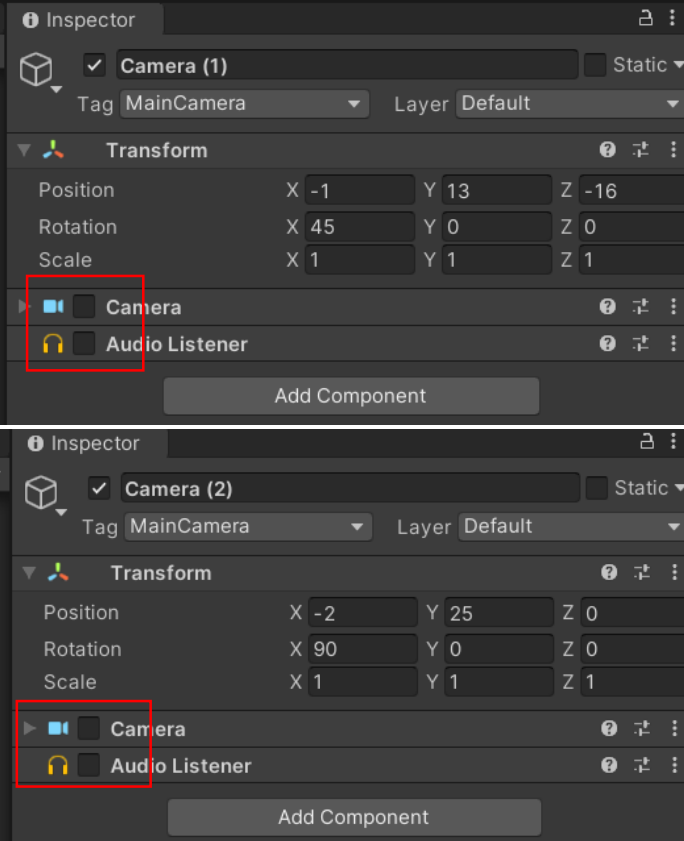
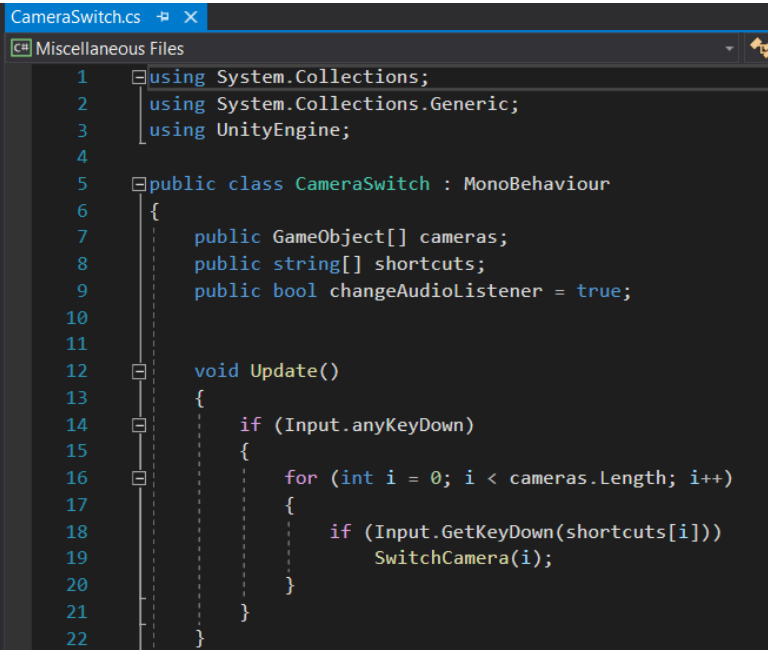
8. Setelah menyimpan script C# tersebut, drag script tersebut menuju Camera yang ada pada Hierarchy

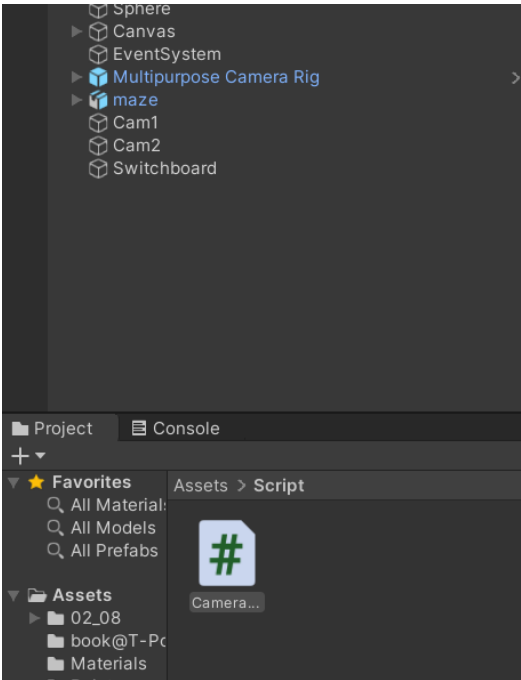
	
9.	<p>Pilih Camera lalu pada Inspector hilangkan centang pada komponen Audio Listener dan rubah beberapa parameter yang ada pada Script PictureInPicture seperti gambar dibawah ini:</p> 
10.	<p>Buat material baru untuk memberi warna pada jalanan nya dengan cara klik kanan pada project > create > material</p>

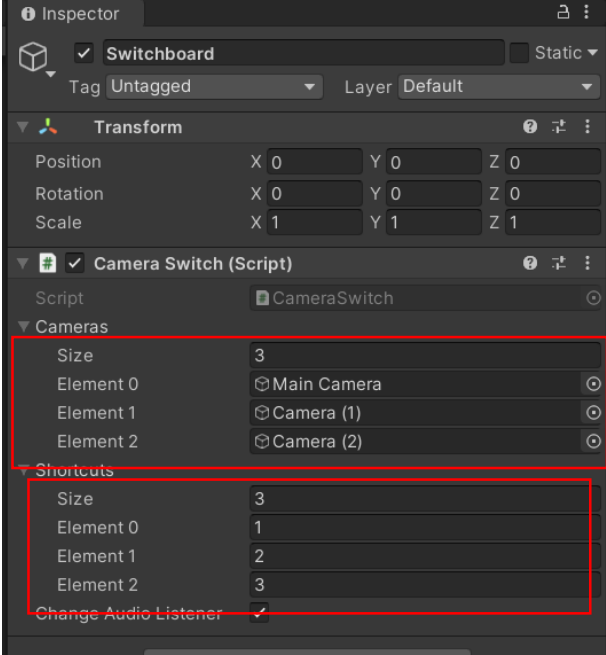
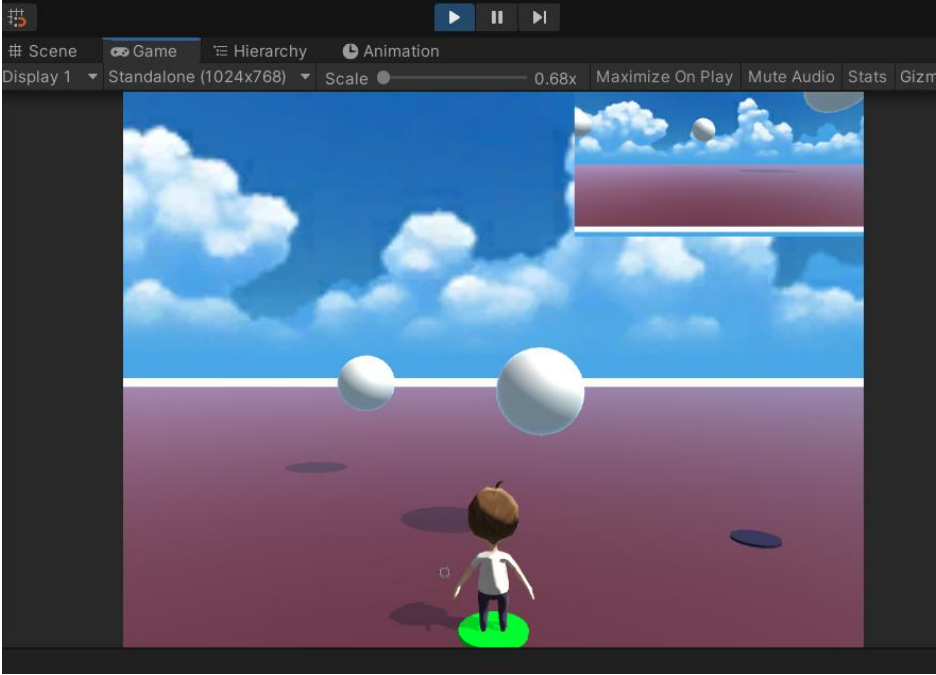
11.	<p>Klik pada object material dan rename menjadi warna-jalanan, lalu rubah warnanya</p>
12.	<p>Masuk ke tab lightning > envirotnmen. Drag gambar Sunny 01A ke dalam skybox material</p>

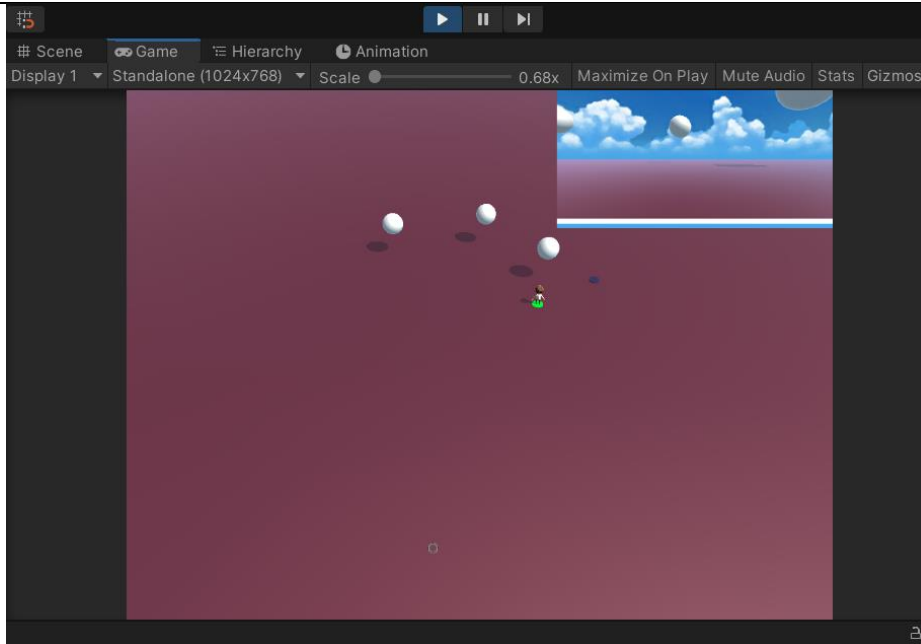
	
13.	<p>Silahkan jalankan program, maka pada layar tampilan akan muncul camera dari arah sudut pandang berbeda yang terletak pada kanan atas layar.</p>  <p>Object male free1 bisa berjalan</p> 
	2. Membuat dan Melakukan Perpindahan Beberapa Sudut Pandang Camera
1.	Import package dari pengerjaan project tugas sebelumnya

2.	<p>Tambahkan 2 new Camera ke dalam scene melalui menu Create yang berada diatas Hierarchy, Pilih Create → Camera. Kemudian beri nama camera1 dan camera2.</p> 
3.	<p>Pada Inspector, rubahlah posisi camera pada camera1 dan camera2 seperti dibawah ini.</p> 
4.	<p>Buatlah GameObject baru menggunakan menu Create pada Hierarchy. Pilih Create → Create Empty. Kemudian ganti nama menjadi Switchboard.</p> 
5.	<p>Dari Inspector pada camera1 dan camera2, hilangkan tanda centang pada komponen Camera dan Audio Listener. Serta atur kolom Tag menjadi MainCamera.</p>

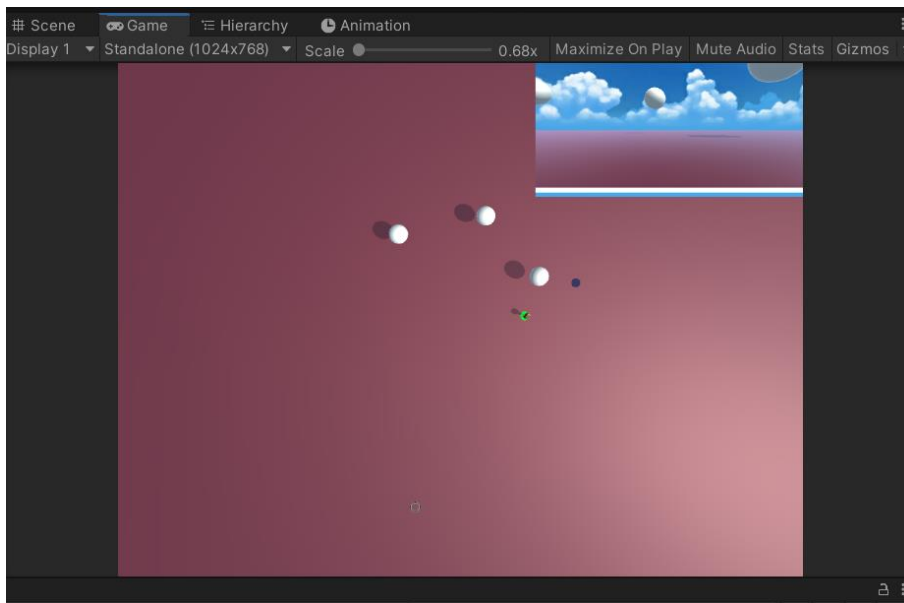
	
6.	<p>Kemudian buatlah script C# dengan code seperti dibawah ini kemudian berilah nama CameraSwitch.</p>  <pre> 1 using System.Collections; 2 using System.Collections.Generic; 3 using UnityEngine; 4 5 public class CameraSwitch : MonoBehaviour 6 { 7 public GameObject[] cameras; 8 public string[] shortcuts; 9 public bool changeAudioListener = true; 10 11 12 void Update() 13 { 14 if (Input.anyKeyDown) 15 { 16 for (int i = 0; i < cameras.Length; i++) 17 { 18 if (Input.GetKeyDown(shortcuts[i])) 19 SwitchCamera(i); 20 } 21 } 22 } </pre>

	<pre> 25 void SwitchCamera(int index) 26 { 27 for (int i = 0; i < cameras.Length; i++) 28 { 29 if (i != index) 30 { 31 cameras[i].GetComponent<Camera>().enabled = false; 32 if (changeAudioListener) 33 cameras[i].GetComponent<AudioListener>().enabled = false; 34 } 35 else 36 { 37 cameras[i].GetComponent<Camera>().enabled = true; 38 if (changeAudioListener) 39 cameras[i].GetComponent<AudioListener>().enabled = true; 40 } 41 } 42 } 43 44 </pre>	
7.	<p>Setelah itu drag script C# tersebut menuju GameObject Switchboard.</p> 	
8.	<p>Pada Inspector, atur ukuran Cameras dan Shortcuts dengan “sizes = 3”. Kemudian, drag dan arahkan secara urut mulai dari Main Camera (berada di Multipurpose Camera Rig → Pivot), cam1, dan cam2 menuju ke setiap slots yang ada pada Cameras. Kemudian pada Shortcuts ketikkan angka 1, 2, dan 3 secara urut pada slots yang tersedia. Seperti gambar di bawah ini:</p>	

		
9.	<p>Jalankan program dan coba tekan angka 1, 2, dan 3 pada keyboard, maka akan terjadi perubahan pada sudut pandang camera utama.</p> <p>Jika dipencet angka 1</p>  <p>Jika dipencet angka 2</p>	

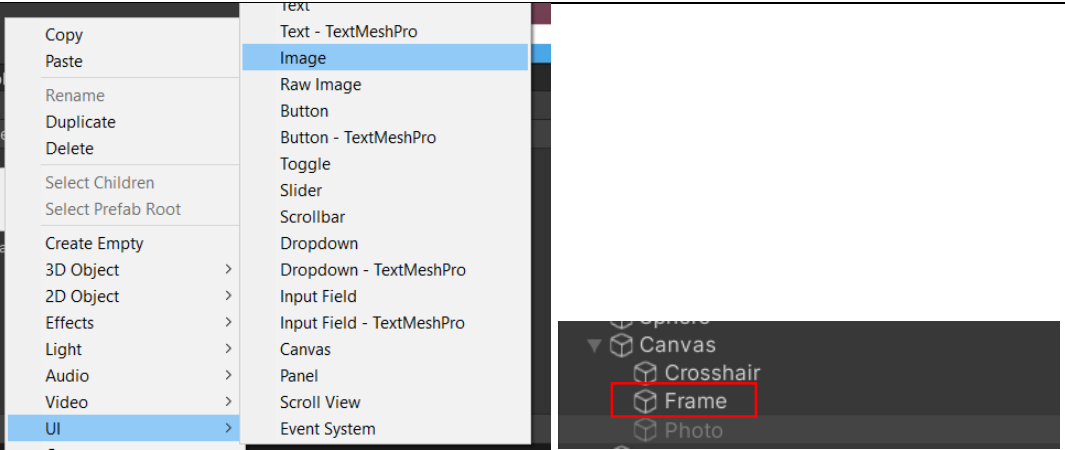
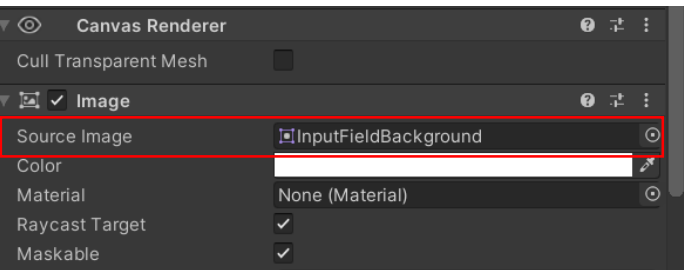
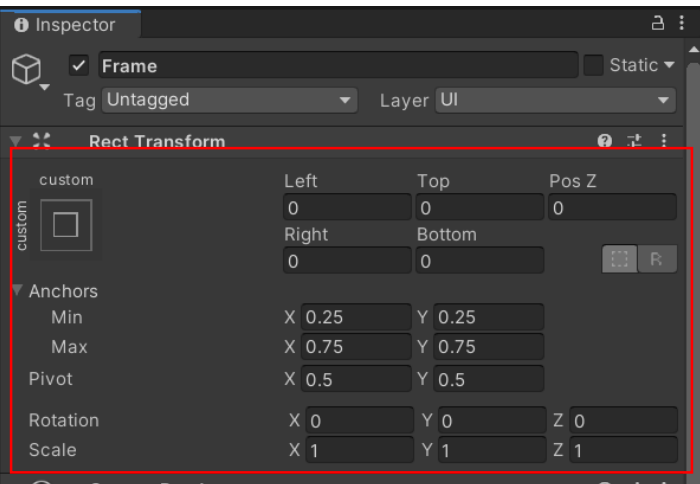
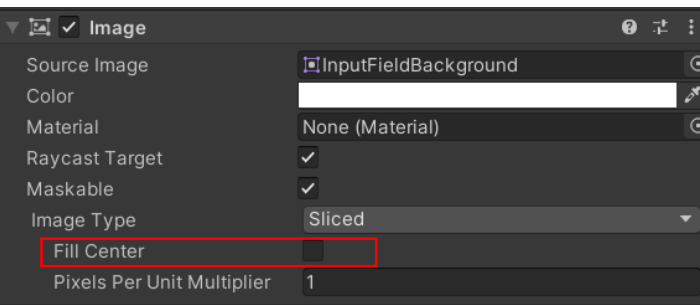


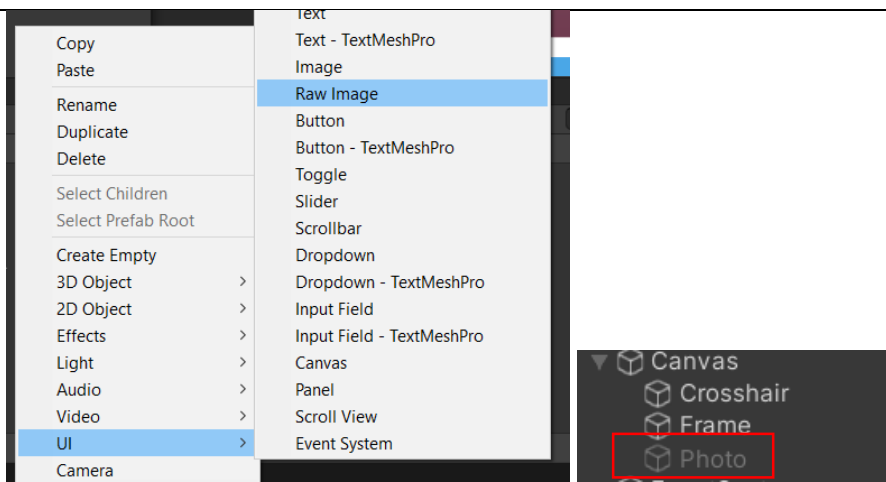
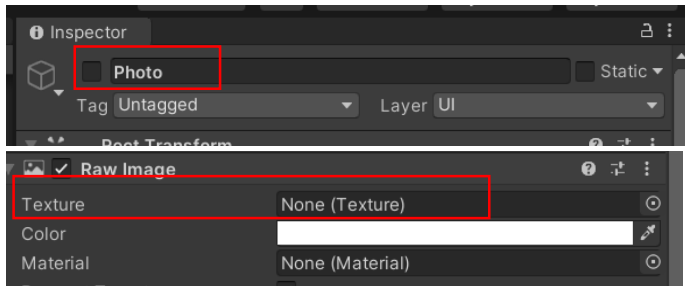
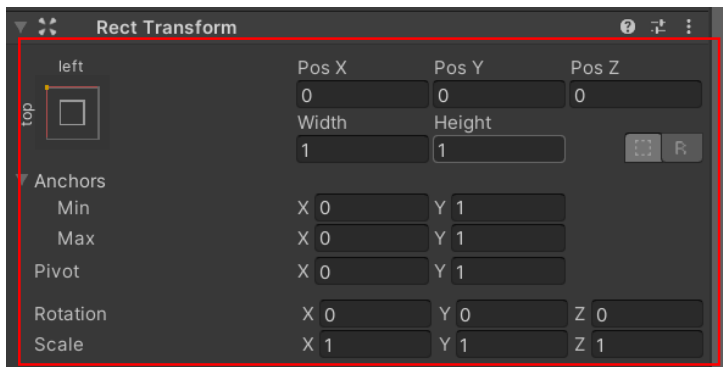
Jika dipencet angka 3



3. Membuat hasil tangkapan (Capture) layar yang berada di dalam frame

1. Buat project baru dan import package dari project pengerjaan tugas sebelumnya
2. Buatlah UI Image dari menu Create yang ada pada Hierarchy, Pilih Create → UI → Image. Setelah itu akan muncul Image yang berada pada Canvas kemudian ganti nama tersebut menjadi frame.

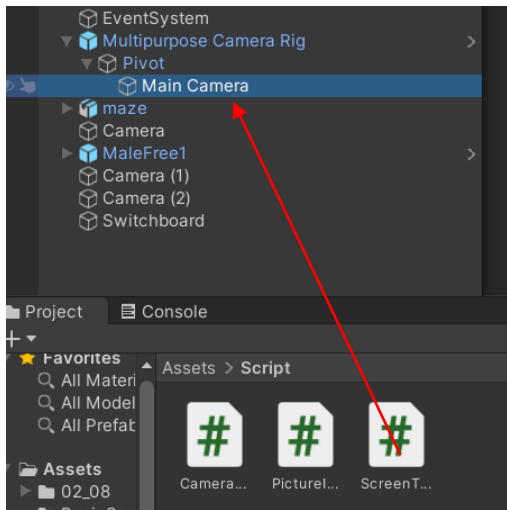
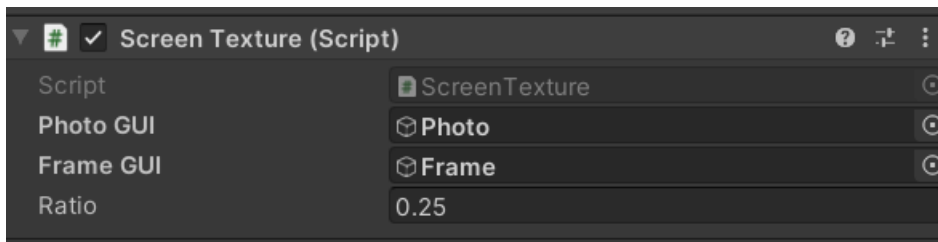
	
3.	<p>Pilih frame pada Hierarchy, kemudian pada komponen Image (Script) yang berada di Inspector atur kolom Source Image menjadi InputFieldBackground. Seperti pada gambar di bawah ini:</p> 
4.	<p>Pada Inspector milik Frame, Rubahlah Rect Transform sesuai dengan gambar dibawah ini:</p> 
5.	<p>Pada komponen Image (Script), hilangkan centang pada Fill Center.</p> 
6.	<p>Buatlah UI Raw Image dari menu Create yang terdapat pada Hierarchy, Pilih Create → UI → RawImage. Setelah itu akan muncul pada Canvas GameObject kemudian ganti nama menjadi Photo.</p>

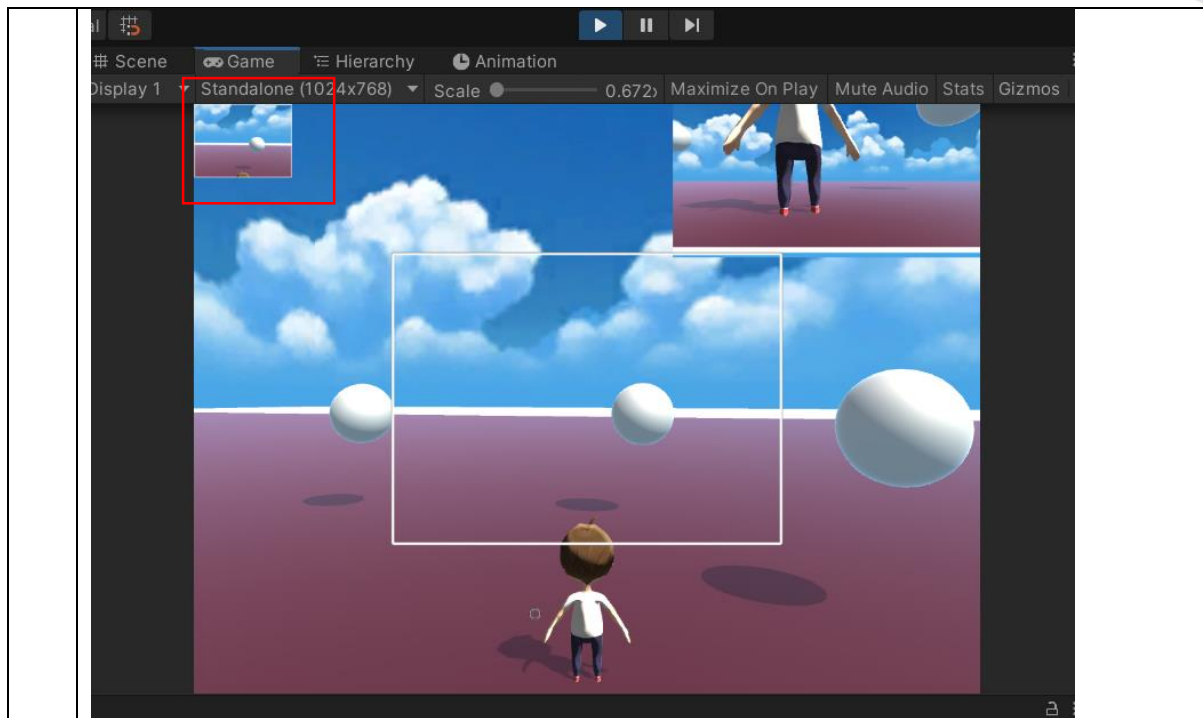
	
7.	<p>Pada Hierarchy pilih Photo, kemudian pada Inspector carilah komponen Raw Image (Script) dan atur kolom Texture menjadi None (Texture). Setelah itu hilangkan centang pada kolom Photo yang berada di atas</p> 
8.	<p>Langkah selanjutnya atur Rect Transform pada Photo sesuai gambar dibawah ini:</p> 
9.	<p>Kemudian buatlah script C# dengan code seperti dibawah ini kemudian berilah nama ScreenTexture.</p>

```

ScreenTexture.cs x
Miscellaneous Files ScreenTexture
1 using UnityEngine;
2 using UnityEngine.UI;
3 using System.Collections;
4
5 /* -----
6  * class to demonstrate how to take snapshots
7  * of the screen and use it as a GUI texture
8  */
9
10 public class ScreenTexture : MonoBehaviour
11 {
12     // GameObject variable for the GUI object where to display the texture
13     public GameObject photoGUI;
14     // GameObject variable for the GUI object to be used as frame
15     public GameObject frameGUI;
16     // Float variable for the ratio between size of the snapshot and displayed texture
17     public float ratio = 0.25f;
18
19     /* -----
20     * During Update, detect if the left mouse button was pressed,
21     * starting the CaptureScreen() coroutine, if so.
22     */
23     void Update()
24     {
25         if (Input.GetKeyUp(KeyCode.Mouse0))
26             // IF the left mouse button was pressed, THEN start the CaptureScreen coroutine
27             StartCoroutine(CaptureScreen());
28     }
29
30     /* -----
31     * A function to calculate the dimension and location of the snapshot,
32     * capture it and apply it to its respective GUI element
33     */
34     IEnumerator CaptureScreen()
35     {
36         // Disable GUI element for the last snapshot taken (otherwise it will be superposed to the next snapshot)
37         photoGUI.SetActive(false);
38
39         // A shorthand for the screen's width
40         int sw = Screen.width;
41
42         // A shorthand for the screen's height
43         int sh = Screen.height;
44
45         // A shorthand for the Rect Transform settings of the GUI element for the framing
46         RectTransform frameTransform = frameGUI.GetComponent<RectTransform>();
47
48         // Rect for the snapshot area, initially based on the GUI frame's the Rect Transform
49         Rect framing = frameTransform.rect;
50
51         // A shorthand for the coordinates of the GUI frame's pivot
52         Vector2 pivot = frameTransform.pivot;
53
54         // A 2D vector for the Anchor Min (defines horizontal and vertical origin of the frame)
55         Vector2 origin = frameTransform.anchorMin;
56
57         // Convert X coordinate of origin point to pixels by multiplying it by screen's width
58         origin.x *= sw;
59
60         // Convert Y coordinate of origin point to pixels by multiplying it by screen's height
61         origin.y *= sh;
62
63         // float var for horizontal offset of the frame, obtained by multiplying horizontal pivot point by frame width
64         float xOffset = pivot.x * framing.width;
65
66         // Add horizontal offset to frame horizontal origin
67         origin.x += xOffset;
68
69         // float var for vertical offset of the frame, obtained by multiplying vertical pivot point by frame height
70         float yOffset = pivot.y * framing.height;
71
72         // Add vertical offset to frame vertical origin
73         origin.y += yOffset;
74
75         // Offset framing horizontal location
76         framing.x += origin.x;
77
78         // Offset framing vertical location
79         framing.y += origin.y;
80
81         // Int variable for texture width, based on framing width
82         int texWidth = (int)framing.width;
83
84         // Int variable for texture height, based on framing height
85         int texHeight = (int)framing.height;
86
87         // Create a new Texture measuring texWidth x texHeight
88         Texture2D texture = new Texture2D(texWidth, texHeight);
89
90         // Wait for the EndOfFrame before capturing snapshot
91         yield return new WaitForEndOfFrame();
92     }
93 }

```


	<pre> 92 //Read Pixels from screen 93 texture.ReadPixels(framing, 0, 0); 94 95 // Apply captured pixels onto texture 96 texture.Apply(); 97 98 // Re-activate GUI element for displaying snapshot 99 photoGUI.SetActive(true); 100 101 // 3D Vector for the new snapshot dimension (based on framing dimension multiplied by selected ratio) 102 Vector3 photoScale = new Vector3(framing.width * ratio, framing.height * ratio, 1); 103 104 // Resize GUI texture display to specified dimensions 105 photoGUI.GetComponent<RectTransform>().localScale = photoScale; 106 107 // Set captured texture as GUI display's texture 108 photoGUI.GetComponent<RawImage>().texture = texture; 109 110 }</pre>
10.	<p>Setelah itu drag script C# tersebut menuju Main camera yang berada pada multipurpose Camera Rig -> pivot</p> 
11.	<p>Pada Inspector milik Main Camera, carilah komponen Screen Texture (Script) kemudian atur kolom Photo GUI dengan memilih Photo. Sama halnya dengan kolom Frame GUI dengan memilih Frame.</p> 
12.	<p>Jalankan program, maka anda akan bisa meng-capture layar yang berada dalam frame dengan cara menekan klik kiri pada mouse. Hasil capture akan tampil pada pojok kiri atas layar.</p>



KODE PEMROGRAMAN

1. Camera Switch

```
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;
                }
            }
        }
    }
}
```

	<pre> { cameras[i].GetComponent<Camera>().enabled = true; if (changeAudioListener) cameras[i].GetComponent<AudioListener>().enabled = true; } } } </pre>
2.	<p>PictureInPicture</p> <pre> 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; </pre>

	<pre> 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); } </pre>
3.	<pre> 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()); } } </pre>



```

    }

    /* -----
    * 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;

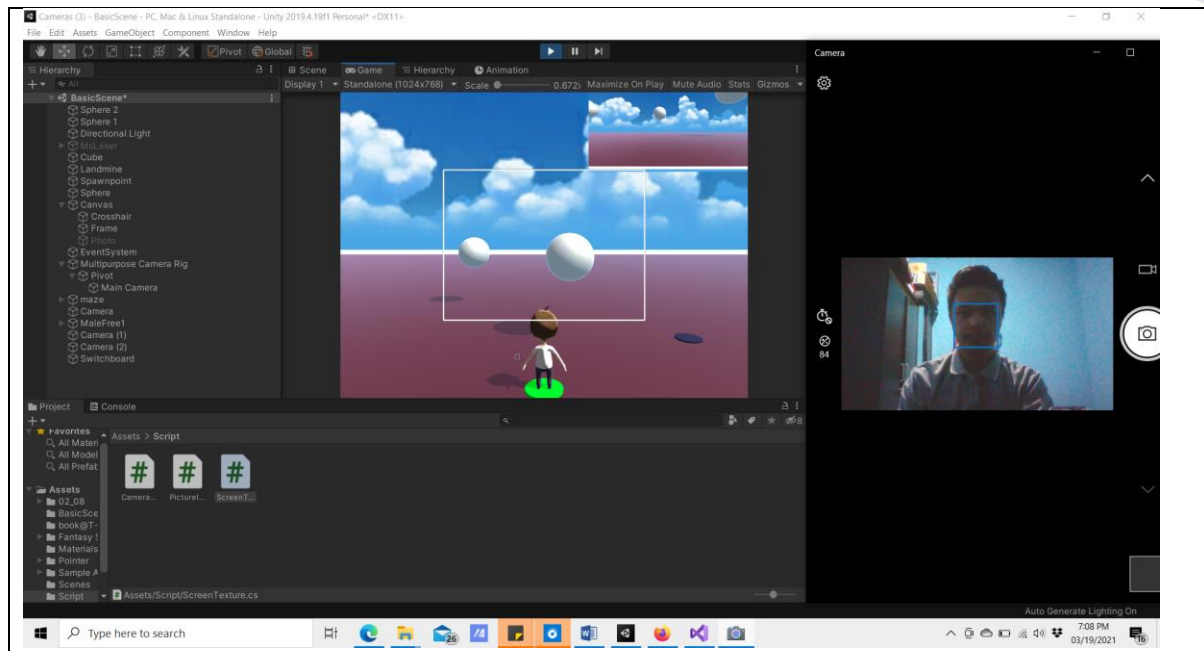
        // Add vertical offset to frame vertical origin
        origin.y += yOffset;

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

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

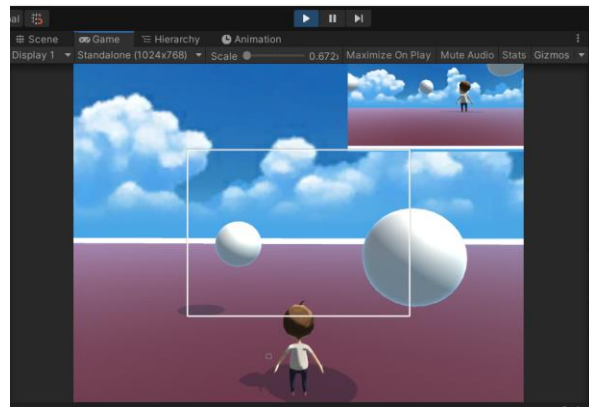
```

	<pre> // 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 multiplied 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; } </pre>
KESIMPULAN	
<p>Saya sudah berhasil membuat game, yang dimana object atau karakter tersebut bisa bergerak atau berjalan. Dan di dalam game tersebut saya sudah bisa meng capture, mengatur posisi sudut (View) dari kamera dan berhasil menampilkan view camera dari arah sudut pandang berbeda yang terletak pada kanan atas layar</p>	
SWAFOTO MAHASISWA+PROJECT	

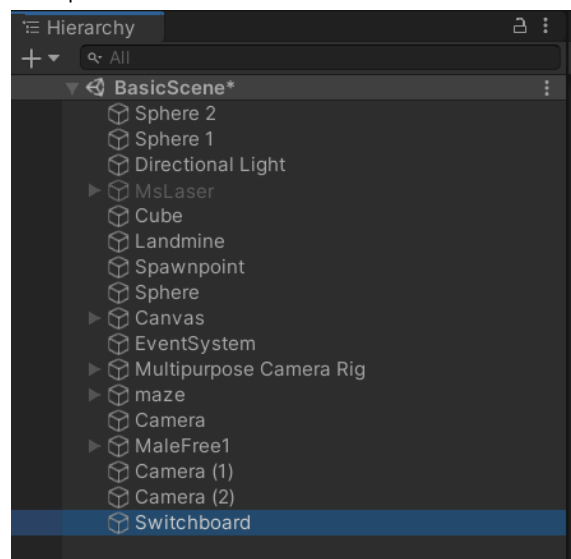


Format laporan:

NIM : 1841720167
 Nama : Raihan Rachmadani
 Kelas : TI – 3H
 Swafoto :
 Deskripsi Game : Sudah, ada di atas
 a. Nama Game : Game berjalan
 b. Alur Game (print screen dan penjelasan) : Untuk alurnya disini hanya berjalan mengelilingi bola- bola putih



c. Komponen materi yang dipakai : Disini menggunakan banyak komponen berjumlah 17 komponen (disertakan screenshoot)



- d. Asset yang dipakai (sertakan : Disini saya selain menggunakan asset yang diberikan oleh dosen pada praktikum, saya juga mendownload asset dari internet bernama Fantasi Skybox Free dan Supercyan Character Pack Free
screenshot, jika terdapat asset yang digunakan berasal dari internet atau sumber lain, cantumkan link)
Link Download assets ([Tutorial Livander Gamedev](https://www.tutorial-livander.com/) ([mediafire.com](https://www.mediafire.com/)))

