

BAB 4

MEMBANGUN APLIKASI KE ANDROID

Pertemuan : 7-8

Peralatan & Perlengkapan : Komputer dan Koneksi Internet

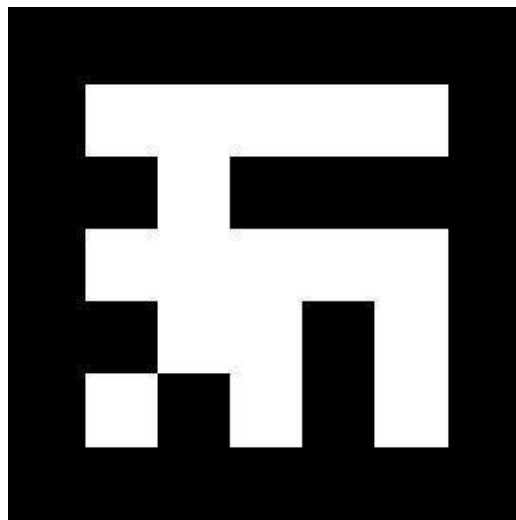
Mahasiswa dapat menggunakan *smartphone* untuk menampilkan objek yang dibuat dalam bentuk APK (Application Package File).

4.1 Konsep/Teori

Pada bahan praktikum kali ini kita akan membuat aplikasi AR dengan sampai membuat file .apk dan di *install* di *smartphone*. Apabila file .apk sudah diinstal di *smartphone* maka kita dapat menjalankan aplikasi yang kita buat di *smartphone*. Pada materi ini kita akan membuat dua aplikasi yaitu: aplikasi yang menggunakan **frame**, dan aplikasi **multiple marker**. Untuk latihan satu kita menggunakan *single marker*, dan untuk latihan kedua kita menerapkan *multiple marker*.

4.1.1 Marker

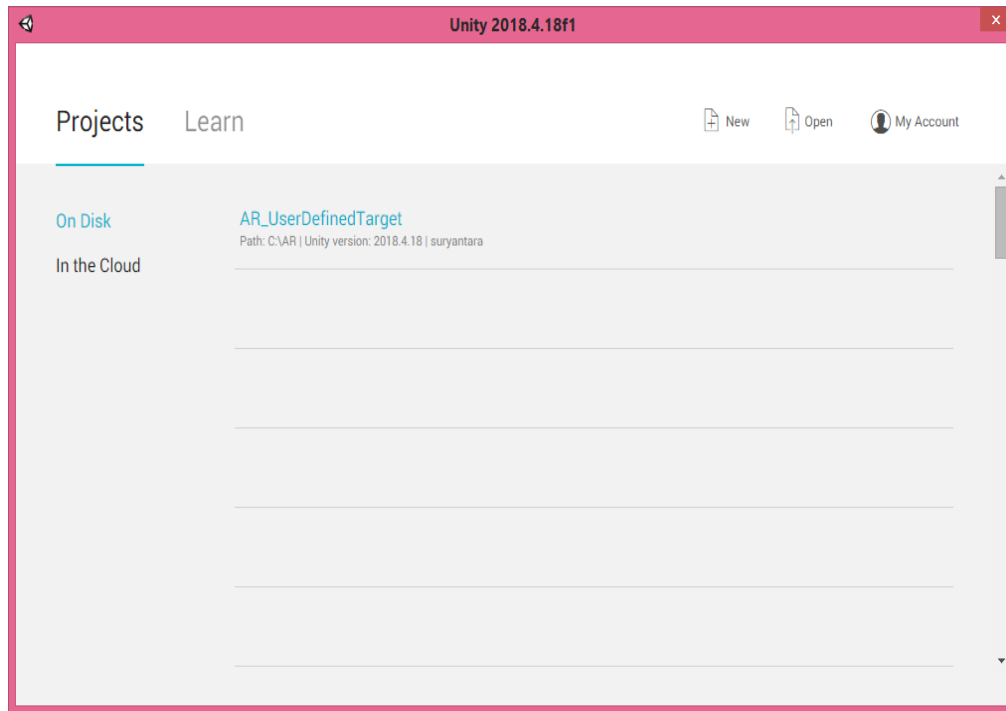
Pada praktikum kali ini kita menggunakan jenis *marker* **single image marker**, pada bahan praktikum minggu 1 kita sudah membuat *marker* dengan **vuforia**, kita gunakan *marker* yang sama seperti pada gambar 4.1.



Gambar 4.1: Ilustrasi *marker*

4.1.2 Menjalankan Unity3D

Sekarang saatnya membuat aplikasi *Augmented Reality* kita. Jalankan Unity3D, seperti pada gambar berikut ini.



Gambar 4.2: Menjalankan Unity3D

4.2 Contoh Soal dan Penyelesaian

Pada latihan kali ini ada dua contoh latihan *Augmented Reality* (AR) yang akan dibahas, dengan menyelesaikan dua latihan ini maka rekan-rekan mahasiswa dapat membuat aplikasi AR lainnya dengan konsep yang hampir sama.

4.2.1 Studi Kasus 1

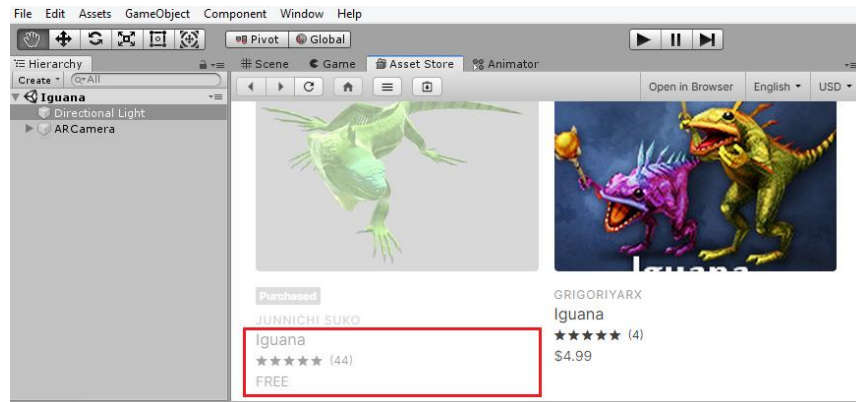
Pada latihan ini kita akan membuat aplikasi AR, pada aplikasi ini adalah dengan men-*scan marker* yang sudah kita buat maka aplikasi akan menampilkan objek 3D secara *augmented*.

Catatan:

Nama latihan kita adalah **Praktek_AR7**. Buatlah aplikasinya, adapun *asset* yang di unduh pada latihan ini adalah **Iguana**, seperti pada tampilan pada Gambar 4.3.

Langkah 1:

Sekarang saatnya kita masukan *asset* (objek) yang akan ditampilkan secara *augmented*. Unity menyediakan *asset* yang dapat diunduh, *asset* yang disediakan oleh Unity ada yang gratis dan berbayar. Kita bisa menggunakan yang gratis (*free*). Kecuali anda mau yang berbayar silahkan.

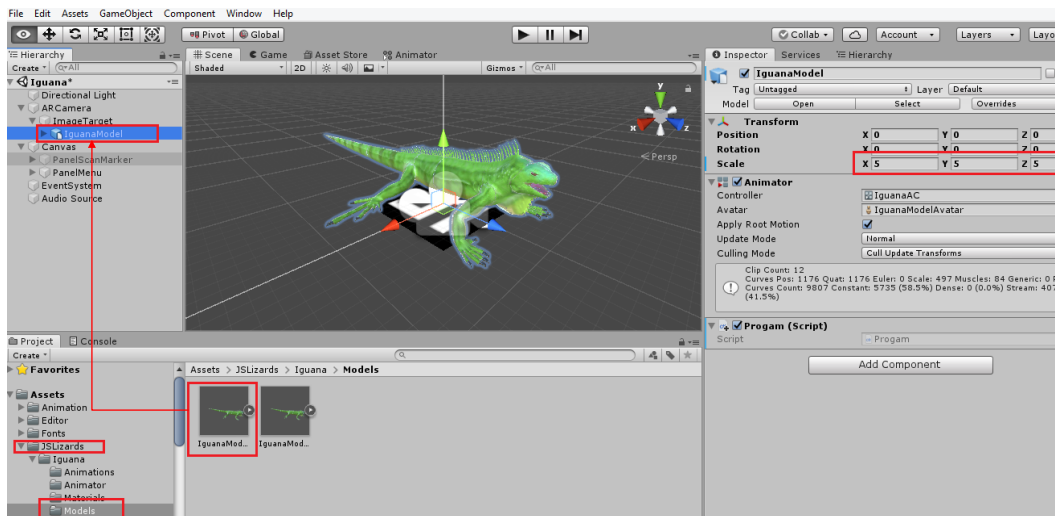


Gambar 4.3: Unduh *Asset Iguana*

Bila sudah selesai mengunduh *asset* selanjutnya klik tombol **import**.

Langkah 2:

Kemudian masukan *asset* yang sudah diunduh ke **Image Target**. **Drag-and-drop** *asset* ke **Image Target**. Ukuran karakter mungkin terlalu kecil, maka dapat diubah ukurannya dengan mengatur Scale menjadi **X=5, Y=5, Z=5** (*sesuaikan dengan kebutuhan*).



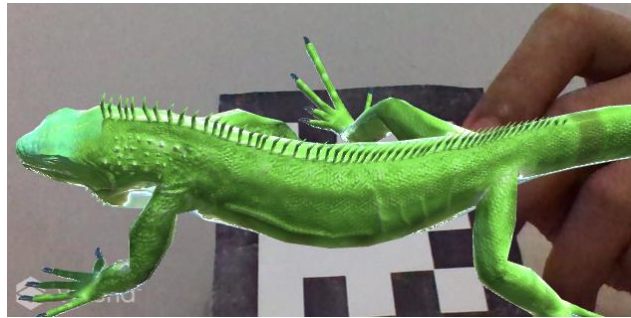
Gambar 4.4: Memasukan *Asset* Ke **Image Target**

Langkah 3:

Lakukan *Build Settings*, bila sudah selesai men-*setting* seperti praktek-praktek biasanya sekarang saatnya memasukkan **License Key**.

Langkah 4:

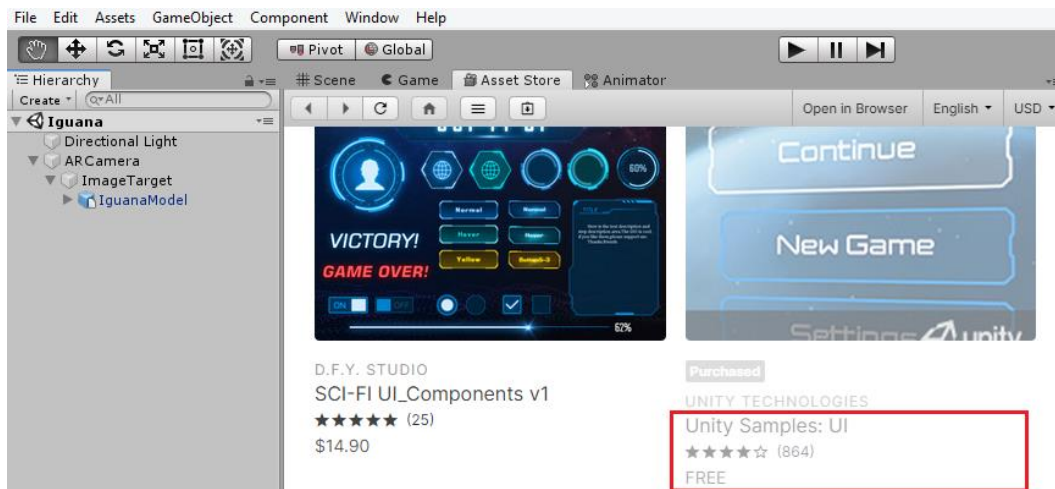
Pada tahap ini anda seharusnya sudah bisa menampilkan objek Iguana secara *augmented* bila marker di-*scan*. Sehingga saat menjalankan aplikasi dan men-*scan marker* maka tampilan aplikasi seperti pada gambar 4.5.



Gambar 4.5: Saat Aplikasi Dijalankan

Langkah 5:

Sekarang kita unduh UI (*user interface*) seperti pada gambar berikut.

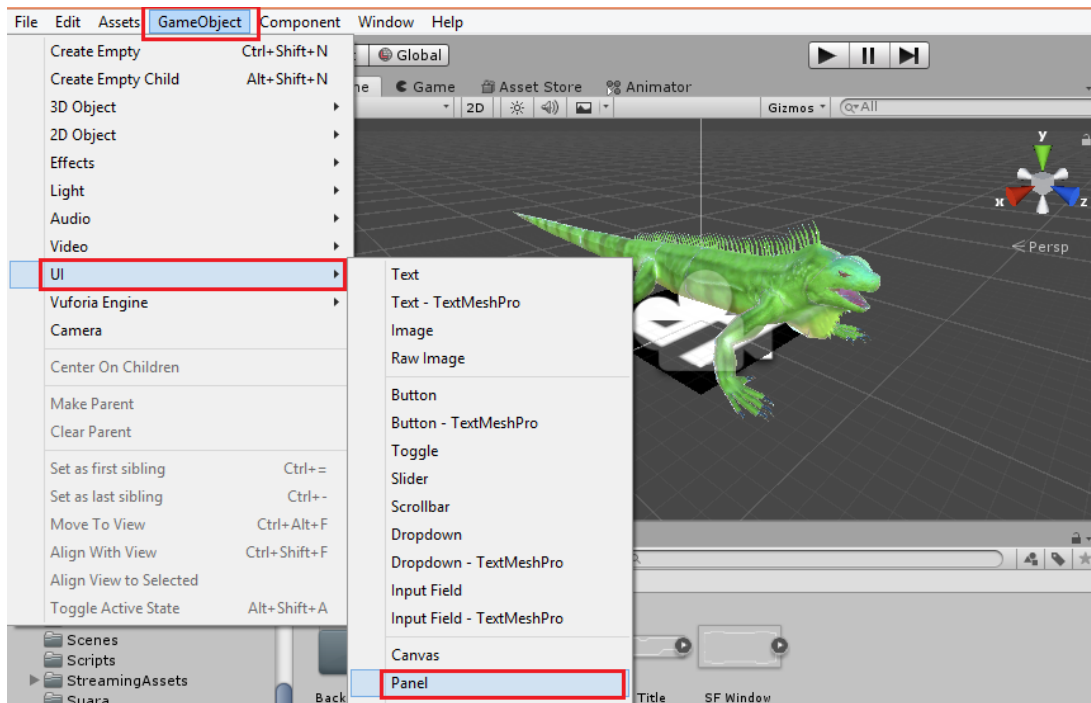


Gambar 4.6: Unduh Asset

Setelah selesai mengunduh *asset*, lanjutkan dengan meng-*import asset*.

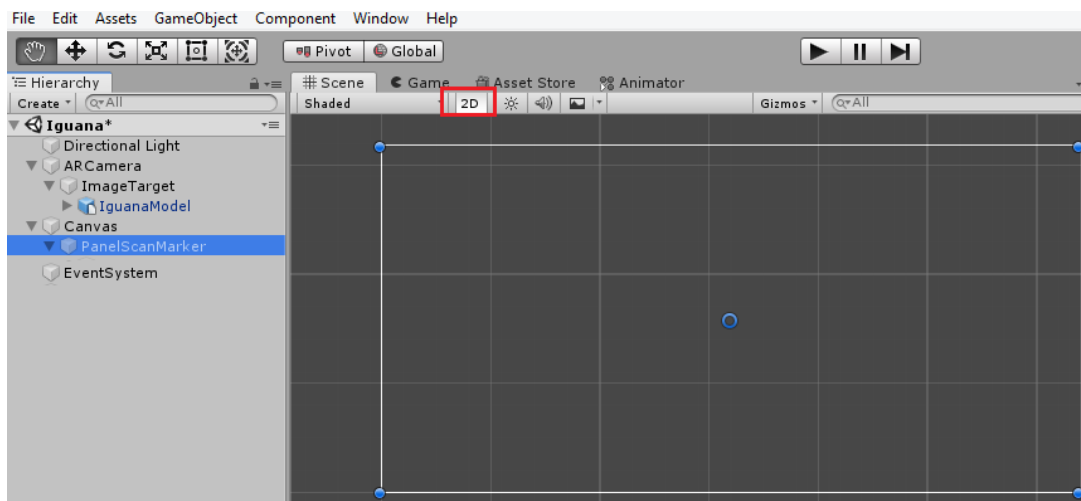
Langkah 6:

Selanjutnya tambahkan **panel**, dengan cara klik menu **GameObject>UI>Panel**.



Gambar 4.7: Tambah Panel

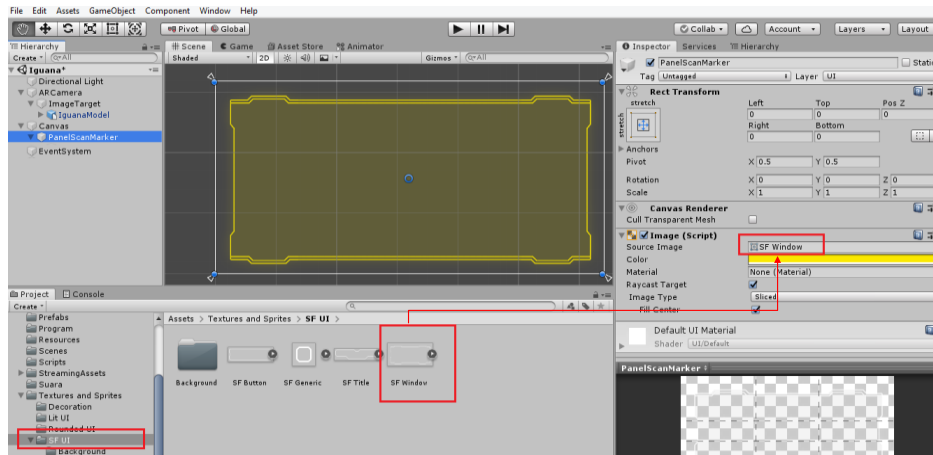
Selanjutnya ganti nama panel dengan nama **PanelScanMarker**, dan ubah layar ke 2D, supaya mudah mendesain.



Gambar 4.8: Tambah Panel

Langkah 7:

Selanjutnya masukan **Frame** ke Panel seperti pada gambar.

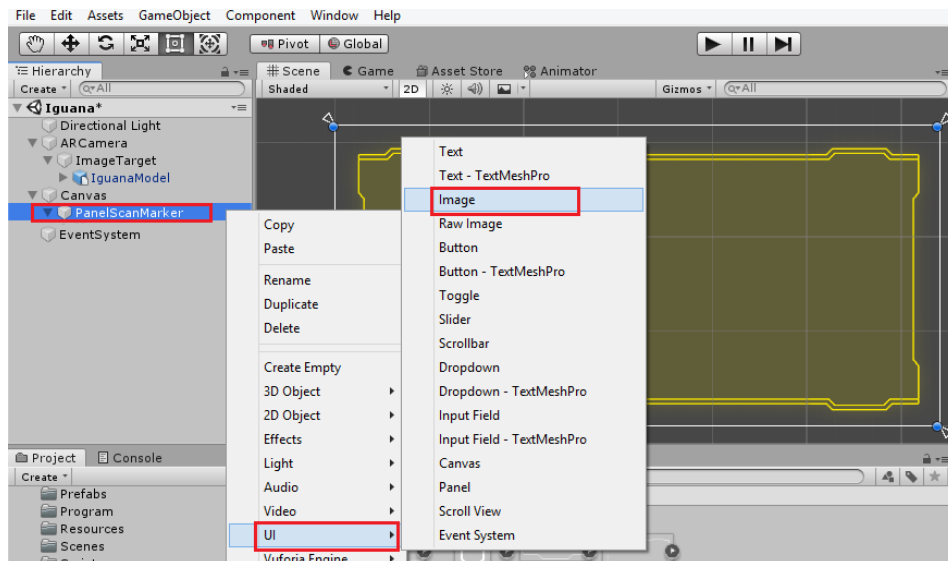


Gambar 4.9: Memasukkan Frame Ke Panel

Ubah warna frame sesuai dengan keinginan anda.

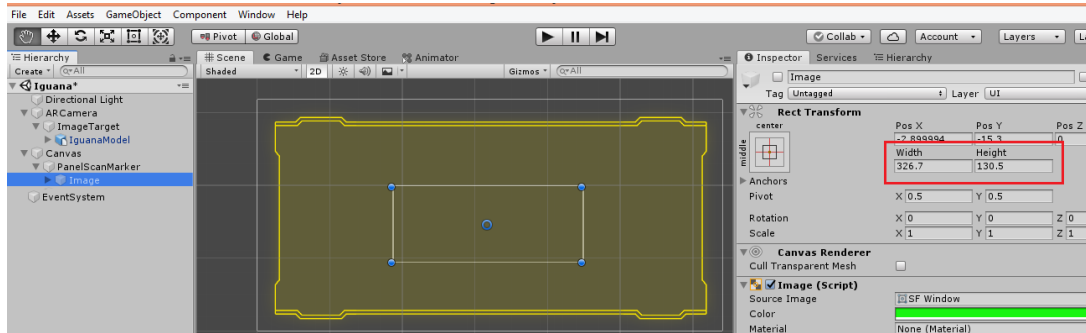
Langkah 8:

Selanjutnya tambahkan sebuah *image* pada panel **PanelScanMarker**, klik kanan pada **PanelScanMarker** seperti pada gambar.



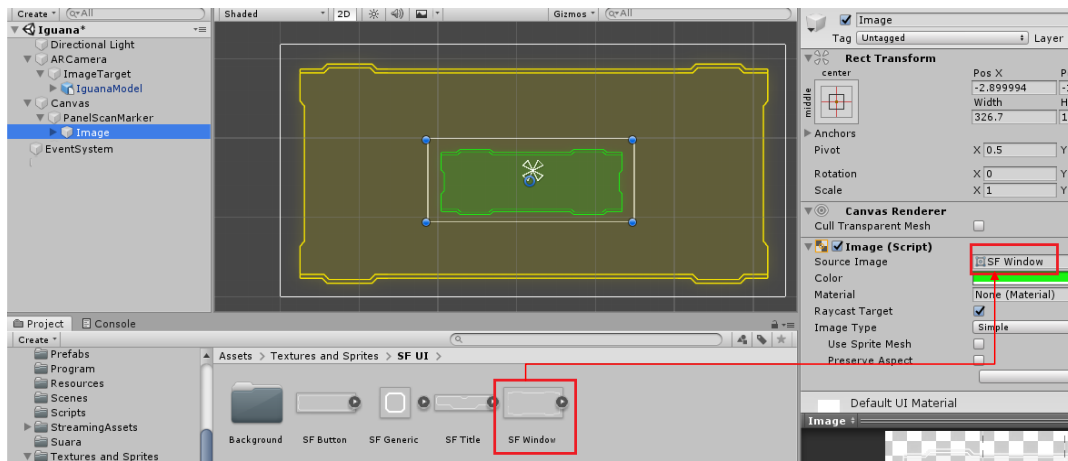
Gambar 4.10: Memasukkan Image Pada Panel PanelScanMarker

Selanjutnya ubah ukuran image yaitu: Lebar = 326.7, dan Tinggi = 130.5.



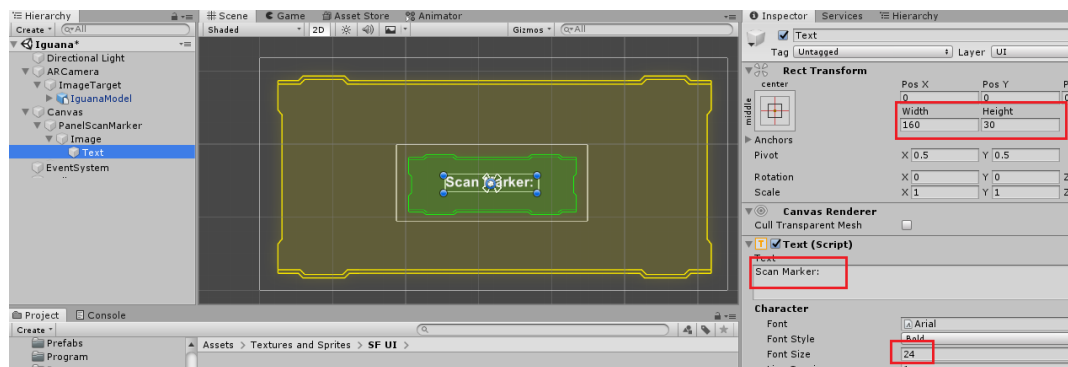
Gambar 4.11: Setting Tinggi dan Lebar Image

Selanjutnya *image* dimasukkan *frame*.



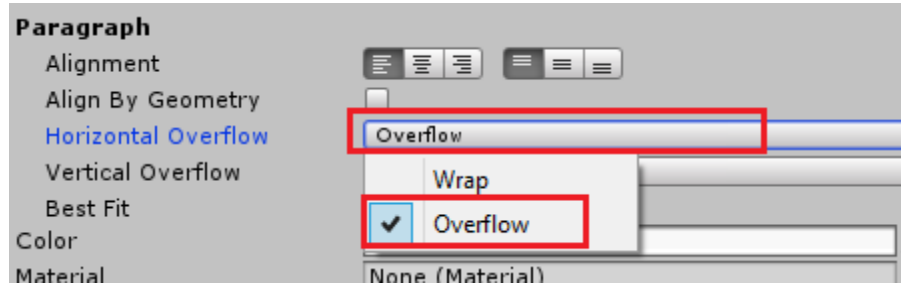
Gambar 4.12: Isi Dengan Frame

Selanjutnya dalam *image* tambahkan *text*;



Gambar 4.12: Tambahkan Text

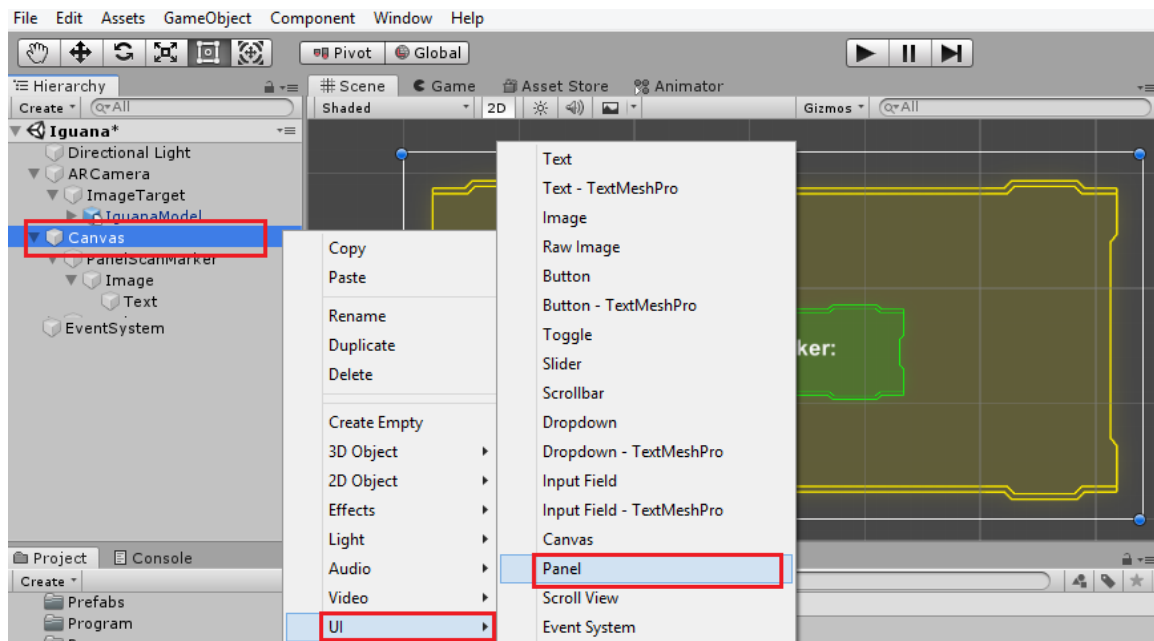
Pada bagian **Horizontal Overflow** ubah ke **Overflow**, seperti pada gambar.



Gambar 4.13: Setting ke Overflow

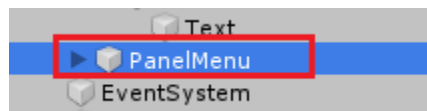
Langkah 9:

Selanjutnya tambahkan *panel* baru di **Canvas**, klik kanan di **Canvas>UI>Panel** seperti ditunjukkan pada gambar.



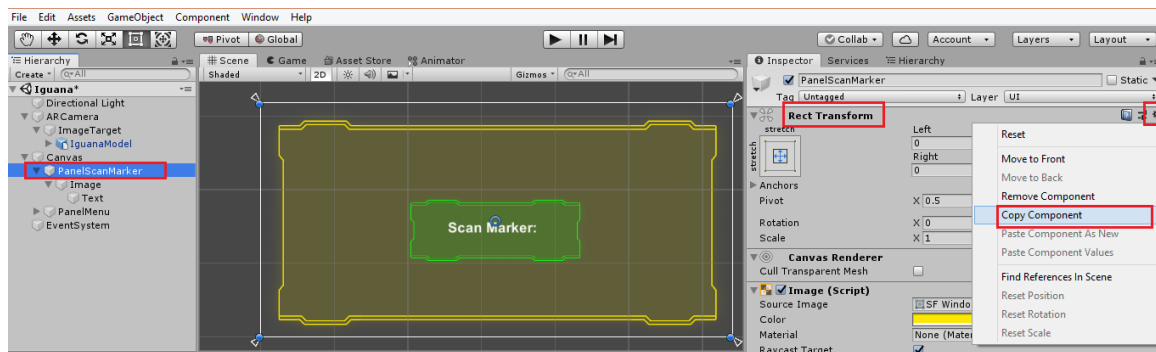
Gambar 4.14: Tambah Panel

Bila sudah ditambah *panel* baru, maka ganti nama *panel* menjadi **PanelMenu**.



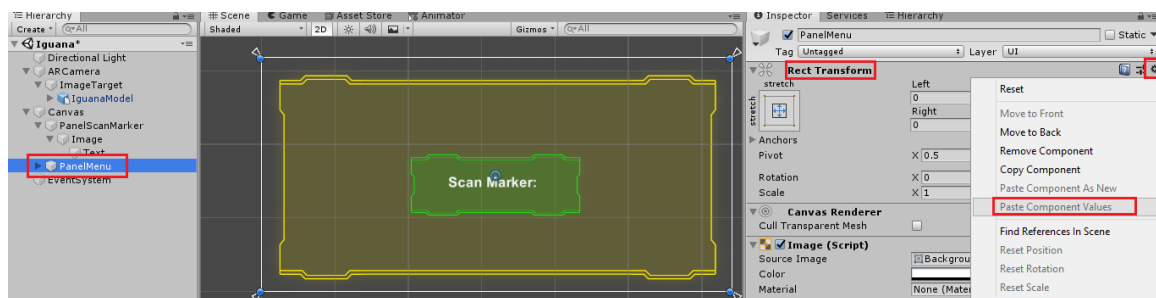
Gambar 4.15: Ganti Nama Panel Menjadi PanelMenu

Selanjutnya klik panel **PanelScanMarker** kemudian pergi ke menu **Inspector** pada bagian **Rect Transform** dan klik yang diberi tanda kotak merah dan klik **Copy Component**.



Gambar 4.16: Copy Component

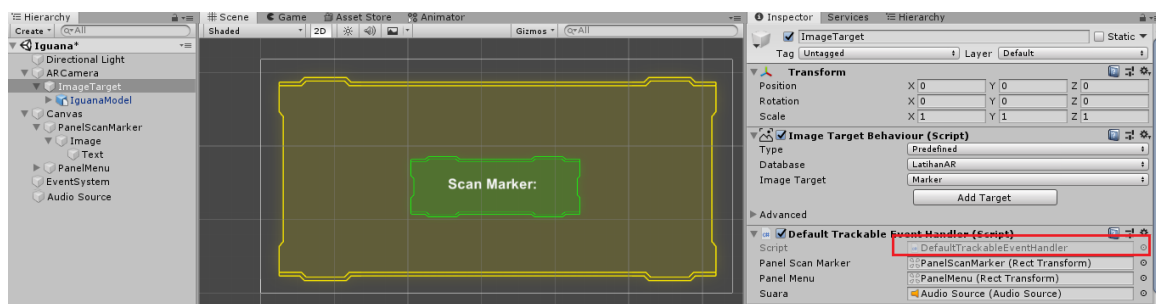
Selanjutnya pergi ke *panel* **PanelMenu** dan lakukan **Paste Component Value**, seperti ditunjukkan pada gambar.



Gambar 4.17: Paste Component

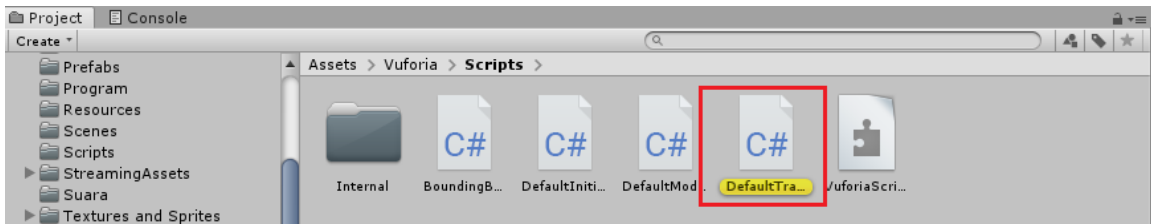
Langkah 10:

Selanjutnya membuat *script*, klik **Imagetarget** kemudian *double click* pada bagian yang diberi tanda kotak merah (di bagian **Script**).



Gambar 4.18: Membuat Script

Maka akan menampilkan kotak dialog seperti pada gambar 4.19, *double click* pada *script DefaultTrackableEventHandler* maka akan menampilkan bagian C#.

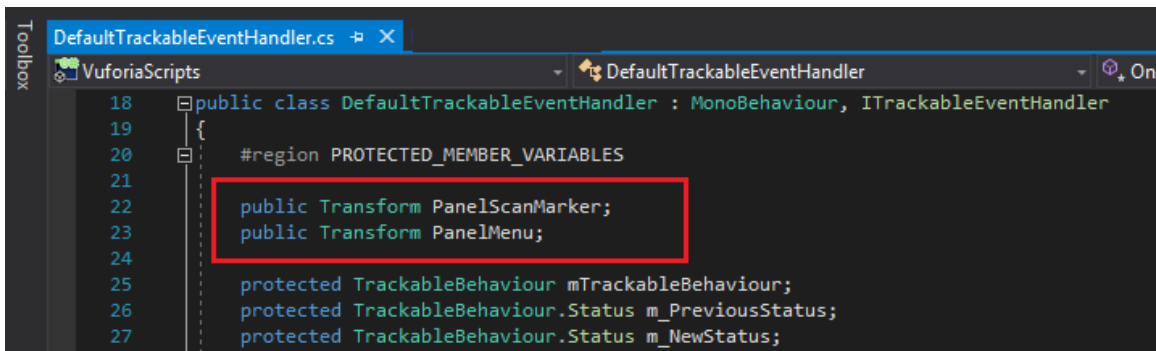


Gambar 4.19: Bagian Script Dengan C#

Langkah 11:

Selanjutnya tambahkan script berikut di C#, pada gambar 4.29 yang diberi tanda kotak.

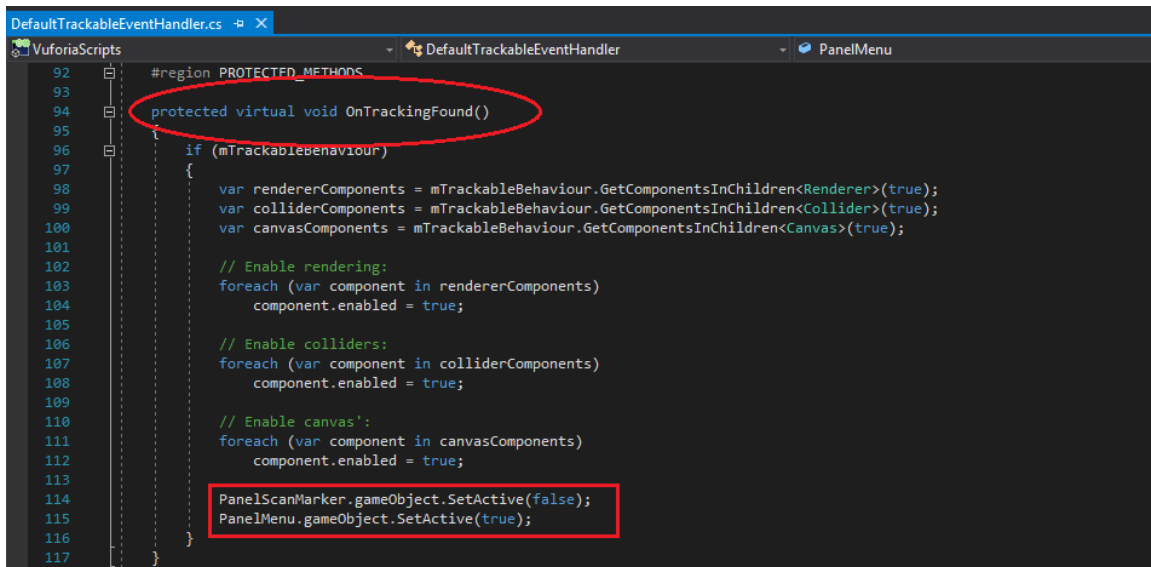
```
public Transform PanelScanMarker;  
public Transform PanelMenu;
```



Gambar 4.29: Menambahkan Script

Tambahkan *script* berikut di bagian `protected virtual void OnTrackingFound();`, yang diberi tanda kotak merah pada gambar.

```
PanelScanMarker.gameObject.SetActive(false);  
PanelMenu.gameObject.SetActive(true);
```



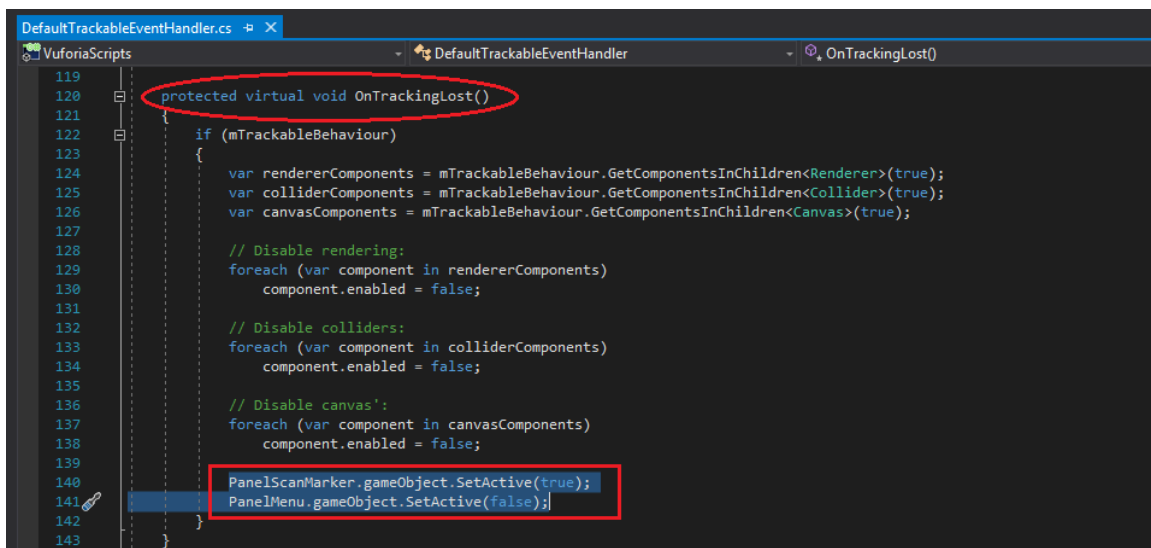
Gambar 4.21: Menambahkan Script

Tambahkan *script* berikut di bagian `protected virtual void OnTrackingLost()`, seperti yang diberi tanda kotak merah pada gambar.

```

PanelScanMarker.gameObject.SetActive(true);
PanelMenu.gameObject.SetActive(false);

```

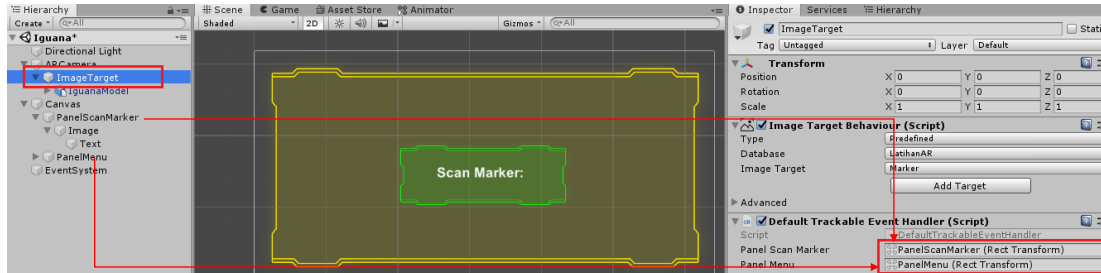


Gambar 4.22: Menambahkan Script

Bila sudah selesai menambahkan *script* pada bagian yang perlu ditambahkan, selanjutnya kembali ke layar **worksheet unity**.

Langkah 12:

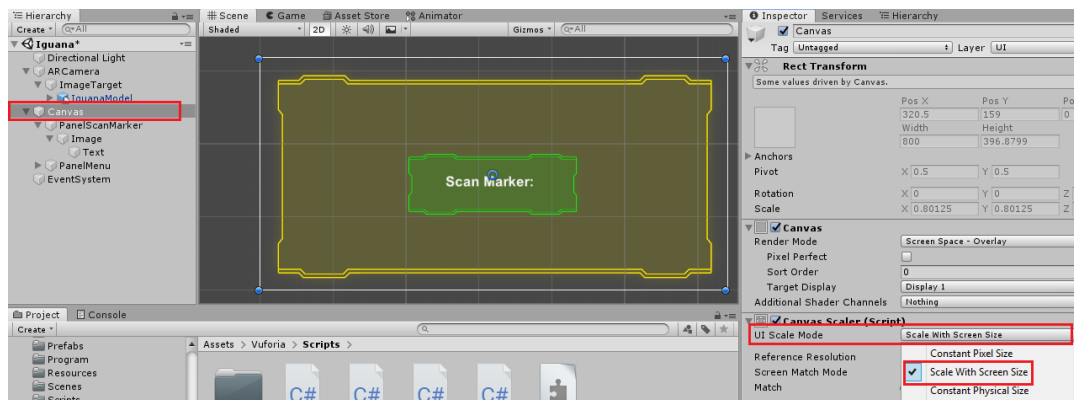
Selanjutnya *drag* **PanelScanMarker** ke posisi yang ditunjuk oleh ujung panah pada gambar, dan drag **PanelMenu** ke posisi ujung panah yang dituju.



Gambar 4.23: Memasukkan Script

Langkah 13:

Selanjutnya klik **Canvas**, dan ubah ke **Scale with Screen Size** seperti pada gambar.



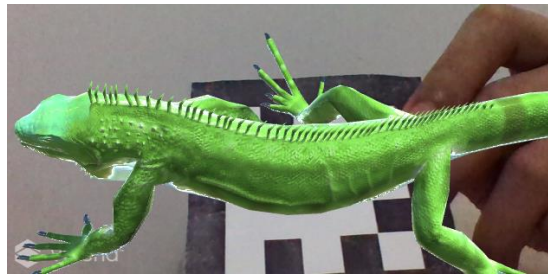
Gambar 4.24: Ubah Setting Canvas

Bila sudah selesai, maka jalankan aplikasinya. Pada saat dijalankan maka akan tampil seperti pada gambar 4.25.



Gambar 4.25: Aplikasi Saat Dijalankan

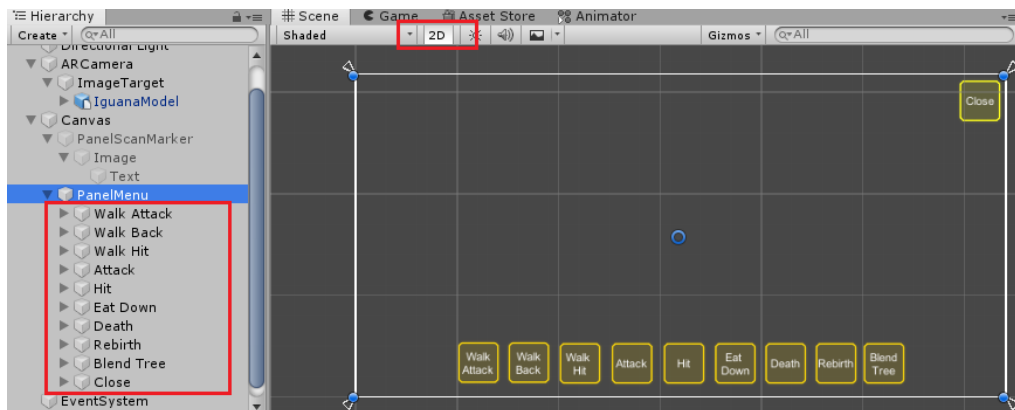
Saat dijalankan aplikasi maka akan meminta pengguna untuk men-*scan marker*, setelah *marker* di-*scan* baru akan menampilkan objek.



Gambar 4.26: Objek Tampil Secara Augmented

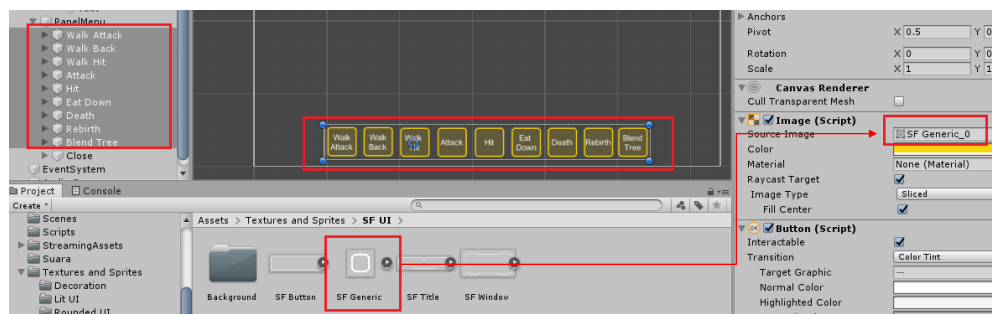
Langkah 14:

Selanjutnya pada panel **PanelMenu** tambahkan tombol (*button*) atur posisinya seperti pada gambar, ukuran tombol adalah: lebar = 60 dan tinggi = 60.



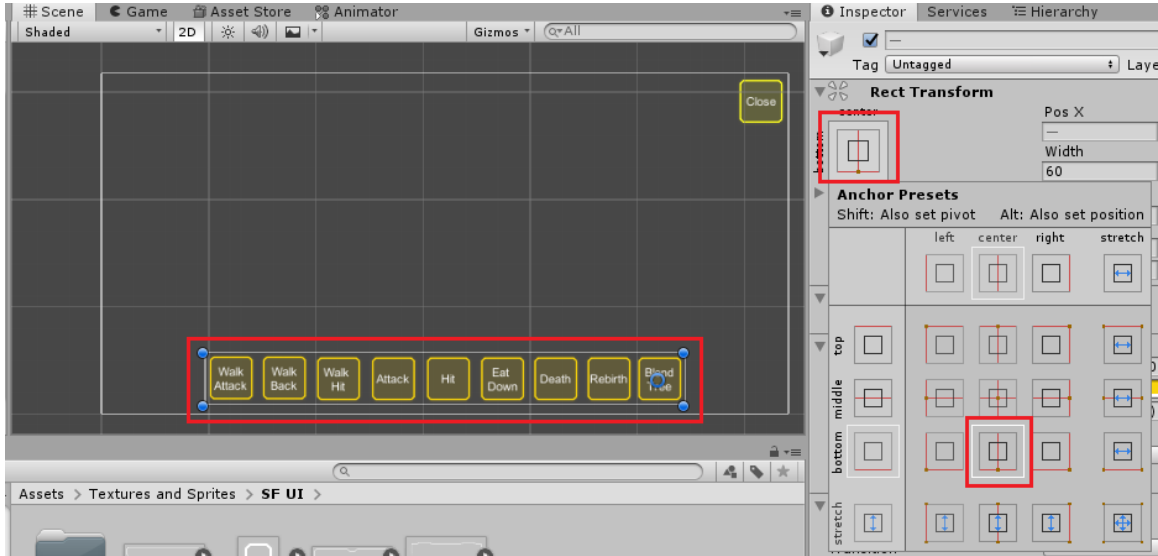
Gambar 4.27: Menambah Tombol

Kotak diisi dengan *Frame* seperti pada gambar, atur posisinya di Tengah Bawah.



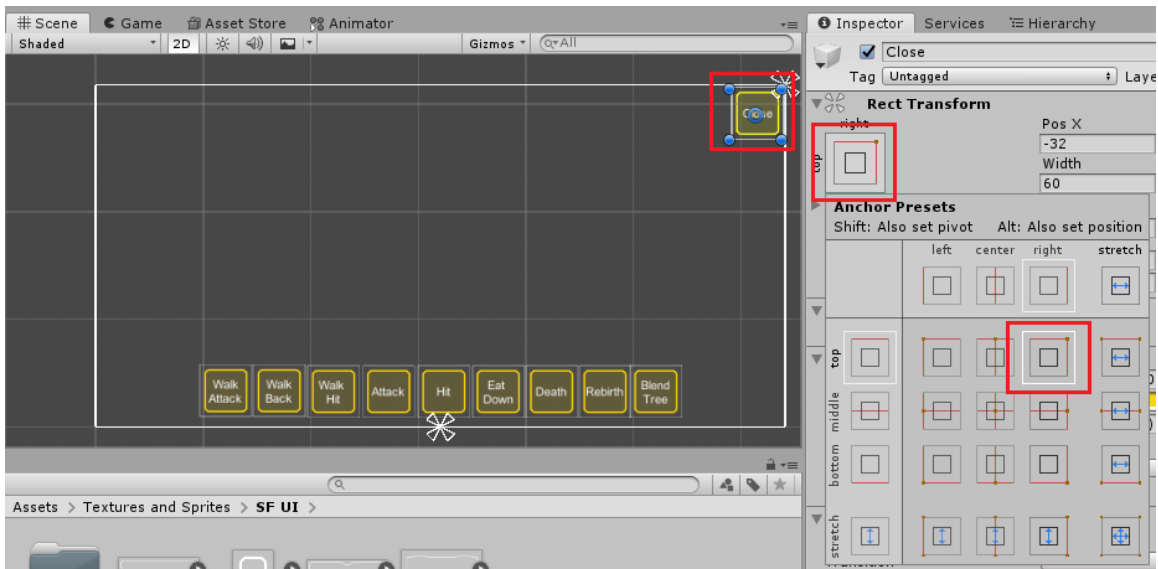
Gambar 4.28: Isi Tombol Dengan Frame

Atur posisi tombol di tengah bawah.



Gambar 4.29: Atur Posisi Tombol

Atur posisi tombol **Close** di posisi pojok kanan atas.



Gambar 4.30: Atur Posisi Tombol Close

Langkah 15:

Selanjutnya membuat *script* untuk menggerakkan objek 3D. Kita buat sebuah *folder* dengan nama **program**. Klik kanan pada **Asset>Create>Folder**. Selanjutnya *double click* folder

tersebut dan buat *script*. Klik kanan dalam *folder Program*, kemudian pilih **Create>C# Script** beri nama *script*-nya dengan nama **Program**. Kemudian *double click* pada *script* dan buat *script*-nya seperti berikut.

```
Nama Script: Program

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

public class Program : MonoBehaviour
{
    private Animator animasi;
    // Start is called before the first frame update
    void Start()
    {
        animasi = GetComponent<Animator>();
    }

    public void animWalkAttack()
    {
        animasi.Play("WalkAttack", -1, 0f);
    }

    public void animWalkBack()
    {
        animasi.Play("WalkBack", -1, 0f);
    }

    public void animWalkHit()
    {
        animasi.Play("WalkHit", -1, 0f);
    }

    public void animAttack()
    {
        animasi.Play("Attack", -1, 0f);
    }

    public void animHit()
    {
        animasi.Play("Hit", -1, 0f);
    }

    public void animEatDown()
    {
        animasi.Play("EatDown", -1, 0f);
    }

    public void animDeath()
    {
        animasi.Play("Death", -1, 0f);
    }

    public void animRebirth()
```

```

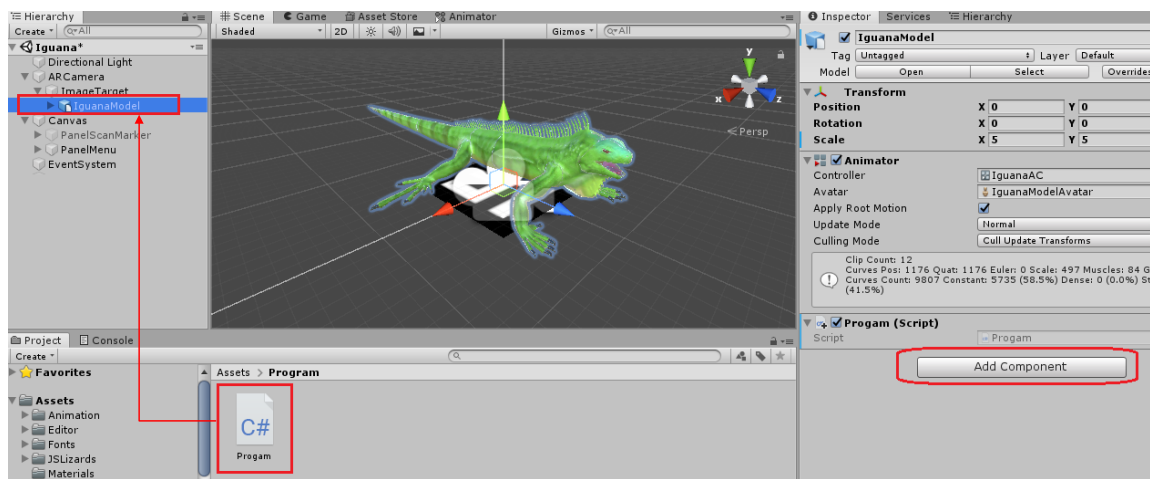
{
    animasi.Play("Rebirth", -1, 0f);
}

public void animBlendTree()
{
    animasi.Play("Blend Tree", -1, 0f);
}

public void QuitAR()
{
    Application.Quit();
    Debug.Log("Keluar!");
}
}

```

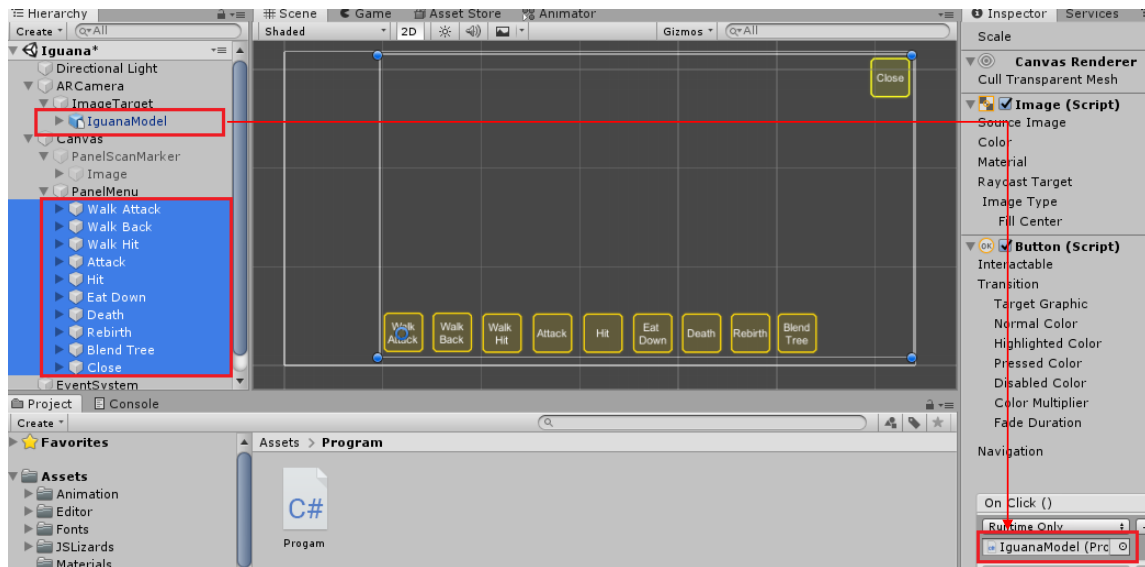
Bila sudah selesai membuat *script*, maka selanjutnya memasukkan *script* ke objek **Iguana** dengan cara mendrag *script Program* ke objek **Iguana** seperti ditunjukkan pada gambar. Atau bisa juga *script* di *drag* ke tombol **Add Component**.



Gambar 4.31: Memasukkan Script

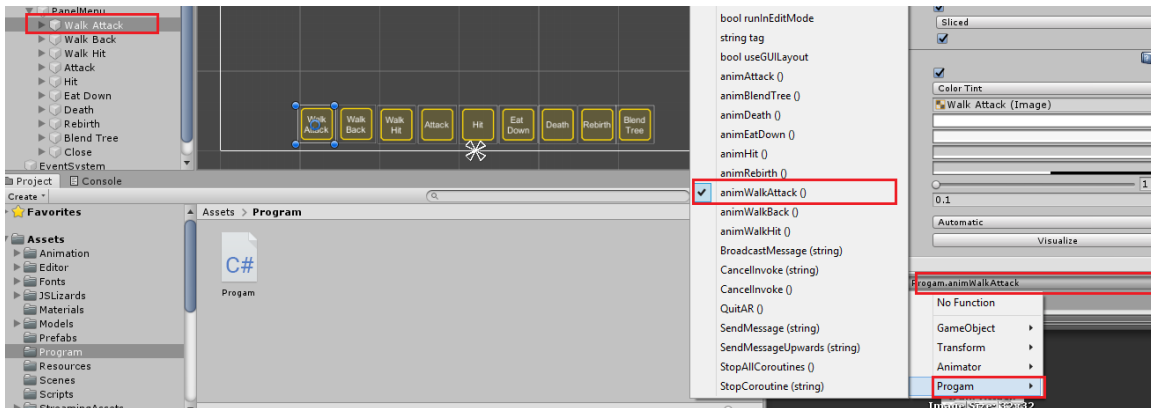
Langkah 16:

Selanjutnya memberi perintah ke masing-masing tombol, blok seluruh tombol kemudian klik tanda (+) pada **On Click** supaya kita bisa memasukkan *script*. Kemudian *drag* objek Iguana ke **On Click** seperti ditunjukkan pada gambar.



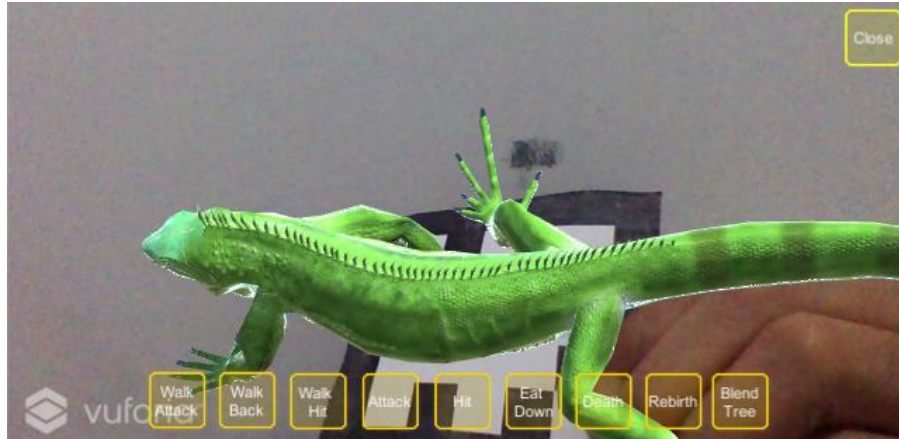
Gambar 4.32: Mendrag Objek Ke On Click

Selanjutnya klik satu-satu tombol untuk memberikan perintah, misalkan klik tombol **Walk Attack** kemudian lakukan seperti pada gambar.



Gambar 4.33: Memberi Perintah Pada Tombol

Lakukan pada semua tombol sesuai dengan perintah yang sudah dibuat di script C#, sampai semua tombol sudah diberi perintah. Bila sudah jalankan aplikasinya. Sekarang kita dapat mengontrol objek dengan menekan tombol. Selamat mencoba.



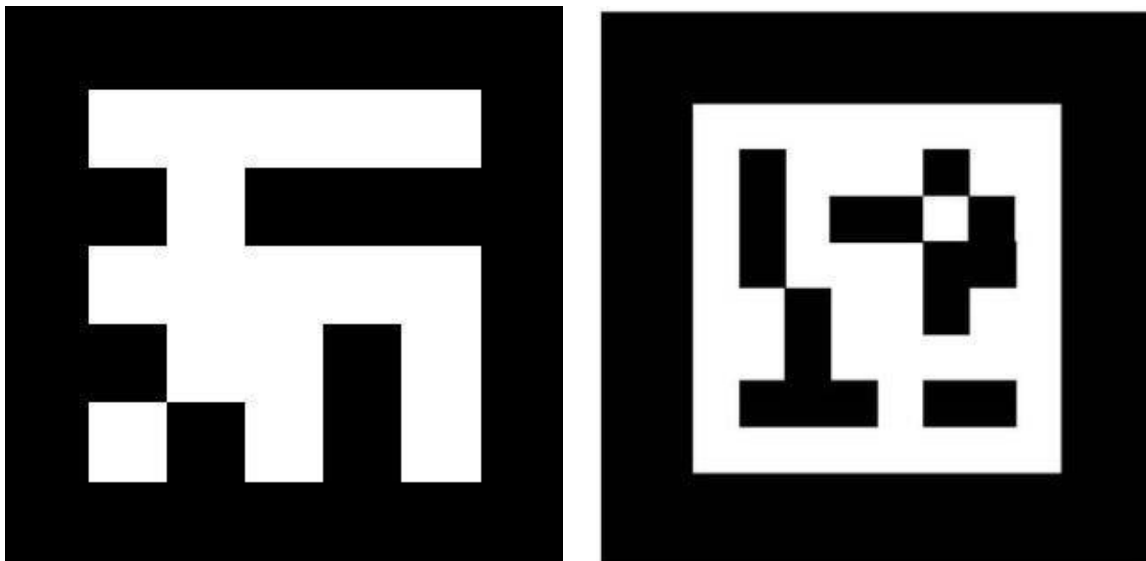
Gambar 4.34: Aplikasi Dijalankan

Catatan:

Unduhlah sound (suara) untuk game, sehingga saat menekan tombol ada efek suara yang dimunculkan. Untuk menambah suara ke tombol dapat dilihat pada pelajaran di modul-modul sebelumnya. Caranya sama, selamat mencoba dan berkreasi. Anda juga dapat menambahkan efek suara saat mau menscan marker, dengan cara rekaman suara anda sebagai suara saat mengarahkan men scan marker.

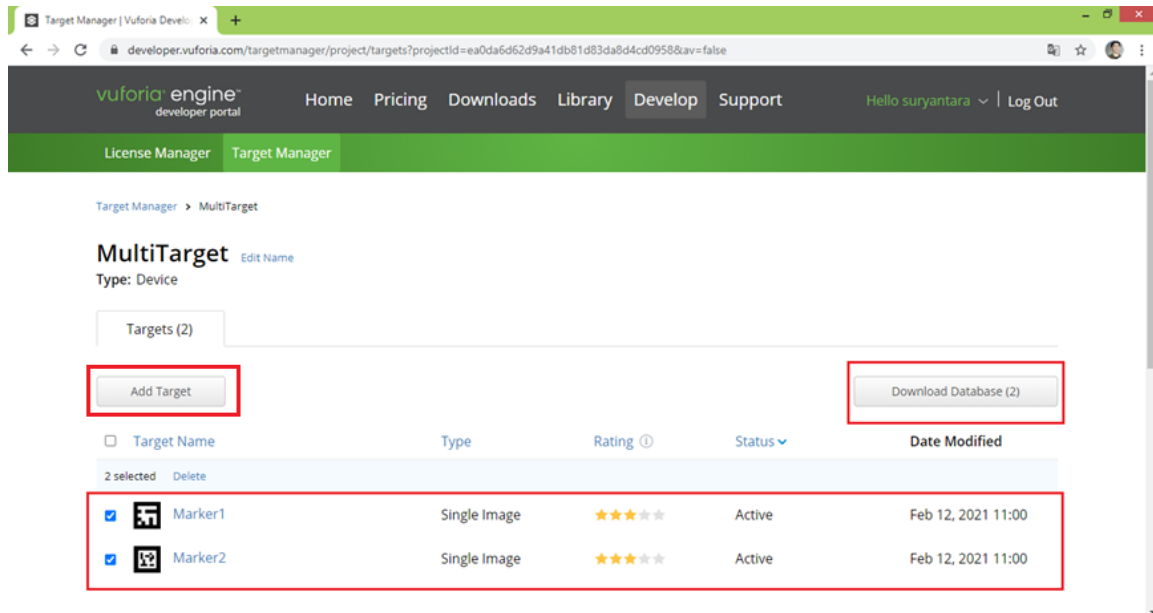
4.2.2 Studi Kasus 2

Pada latihan 2 kita menerapkan **multiple marker**, marker yang digunakan seperti gambar 4.35.



Gambar 4.35: Marker

Buatlah *marker* baru dengan nama **MultiTarget**, masukan dua buah gambar sebagai marker. Pada saat selesai memasukan gambar ke-1, selanjutnya klik tombol **add target** memasukan gambar ke-2, sehingga kita memiliki 2 buah gambar sebagai marker. Seperti pada tampilan gambar 4.36.



Gambar 4.36: Marker Yang Digunakan

Bila sudah selesai membuat *marker*, klik tombol **Download Database (all)**.

Download Database

2 of 2 active targets will be downloaded

Name:
MultiTarget

Select a development platform:

☐ Android Studio, Xcode or Visual Studio

☒ Unity Editor

Cancel

Download

Gambar 4.37: Marker Yang Digunakan

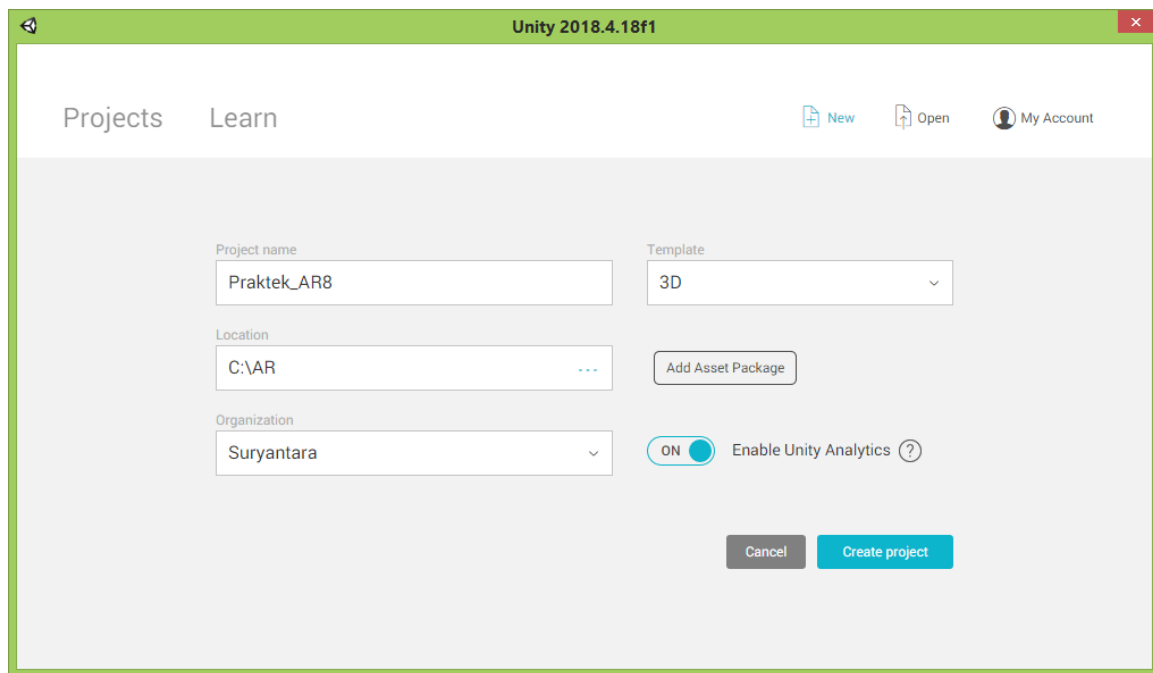
Dari tampilan gambar 4.37 klik *radio button* **Unity Editor**, dan selanjutnya klik tombol **Download**. Setelah selesai mengunduh silahkan simpan dengan nama filenya **MultiTarget**.

Catatan:

Gambar yang dijadikan marker bebas, namun pada latihan ini kita menggunakan marker seperti pada gambar 4.35. Dalam membuat multiple marker kita bisa menggunakan banyak gambar sebagai marker. Sesuaikan dengan kebutuhan, dalam latihan ini kita menggunakan dua gambar sebagai marker. Kegunaanya adalah satu marker untuk satu karakter (objek), dalam latihan ini kita akan menampilkan dua buah karakter, masing-masing karakter diwakilkan dengan satu marker.

Langkah 1:

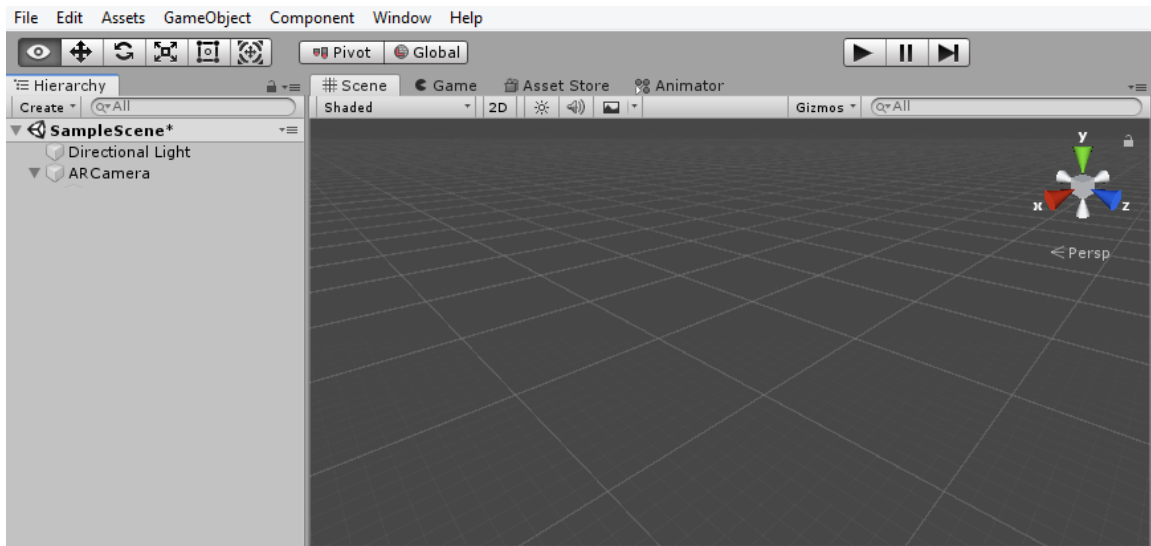
Sebagai latihan mandiri, buatlah sebuah proyek dengan nama **Praktek_AR8**, proses tahapan-tahapannya kurang lebih sama seperti pada studi kasus sebelum-sebelumnya.



Gambar 4.38: Membuat Proyek Baru

Langkah 2:

Setelah memasukkan nama proyek, dari gambar 4.38 klik tombol **Create Project**, maka akan tampil *worksheet* seperti pada gambar 4.39.

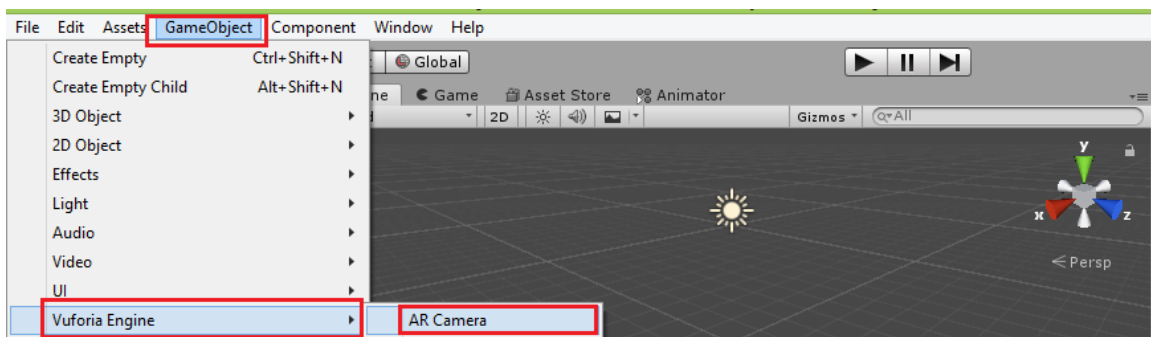


Gambar 4.39: Worksheet

Dari worksheet, hapus **MainCamera**, kemudian tambahkan **ARCamera**. Selanjutnya jalankan database marker **MultiTarget** yang sudah kita unduh dari Vuforia, kemudian klik tombol **Import**. Jangan lupa kopi *license key* multi target dari vuforia.

Langkah 3:

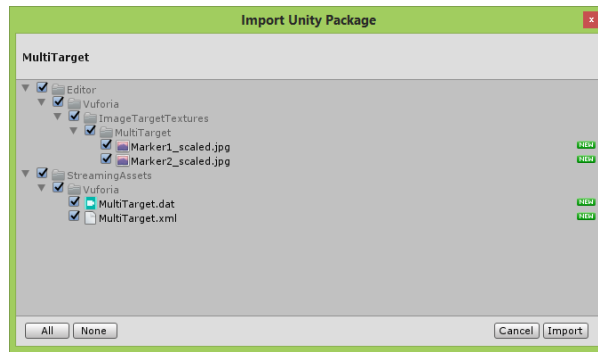
Lakukan **build setting**, langkah ini seperti pada langkah-langkah yang sering kita lakukan pada membuat latihan AR.



Gambar 4.40: Masukkan AR Camera

Langkah 4:

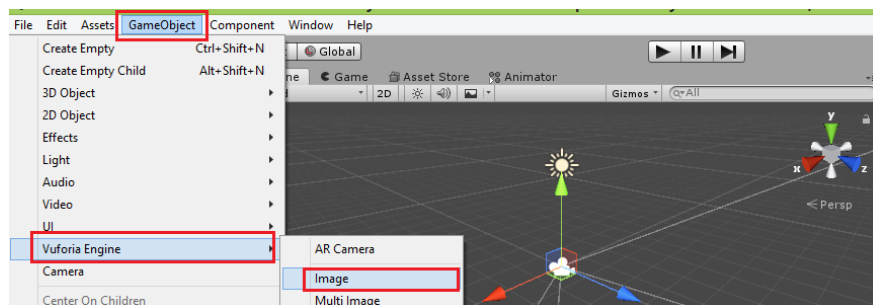
Double klik database **MultiTarget** yang sudah diunduh, kemudian klik **Import**.



Gambar 4.41: Mengimport Database Marker

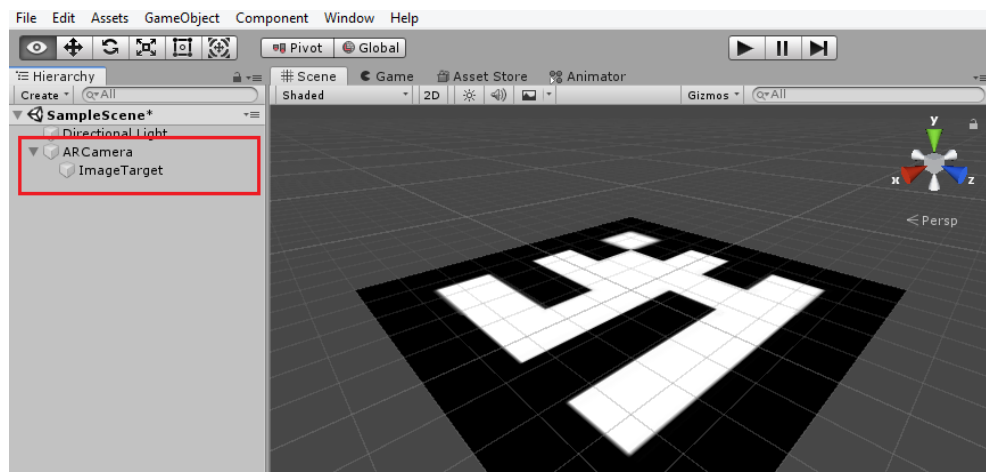
Langkah 5:

Tambahkan **image target**, kita membutuhkan dua buah *image target* karena *marker* yang kita gunakan ada dua.



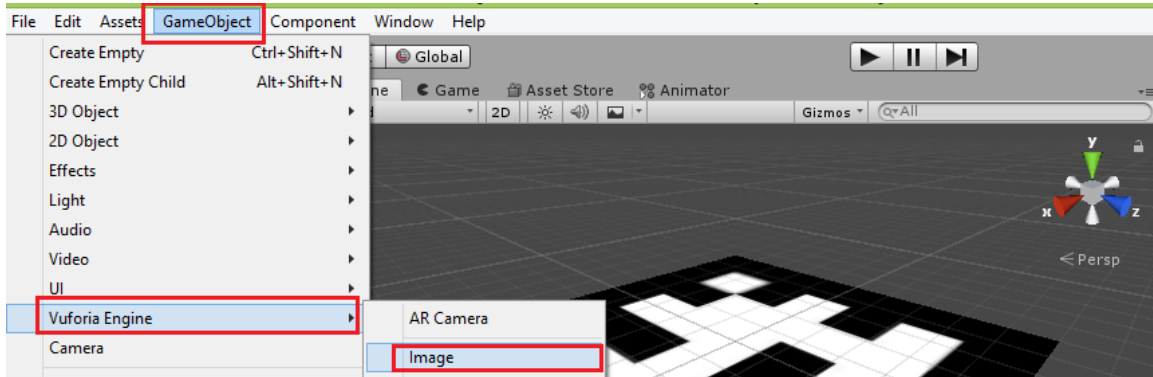
Gambar 4.42: Menambahkan Image Untuk Marker

Sehingga tampil seperti pada gambar 4.43.



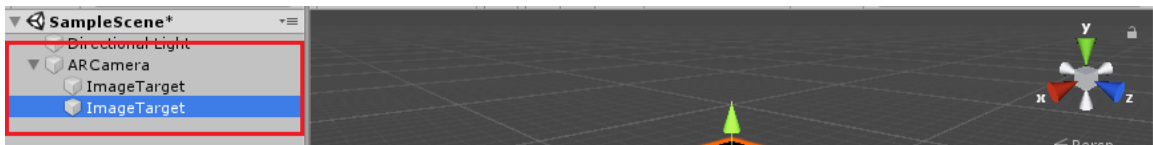
Gambar 4.43: Memasukkan Marker

Tambahkan *image target* lagi untuk *image target market* ke-2.



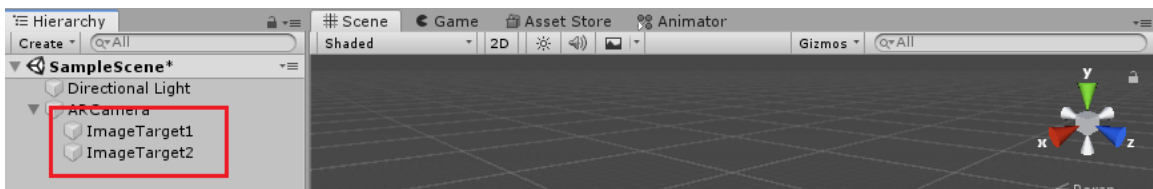
Gambar 4.44: Menambahkan Image

Sehingga hasilnya seperti pada gambar 4.45.



Gambar 4.45: Duah ImageTarget Untuk Marker

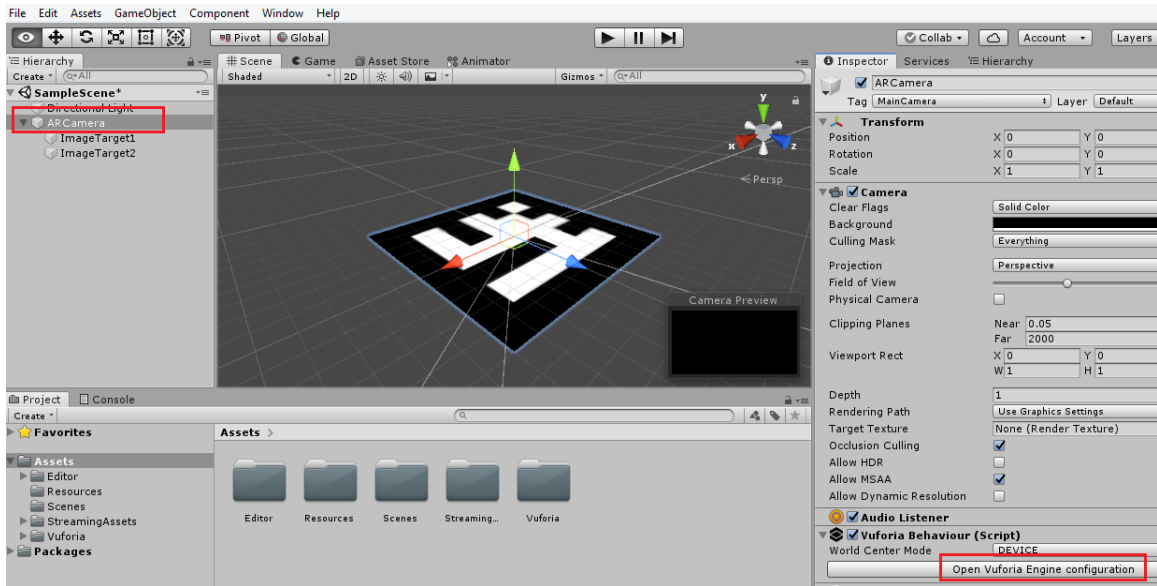
Selanjutnya ganti nama ImageTarget menjadi **ImageTarget1** dan **ImageTarget2**, seperti pada gambar 4.46.



Gambar 4.46: Mengganti Nama ImageTarget

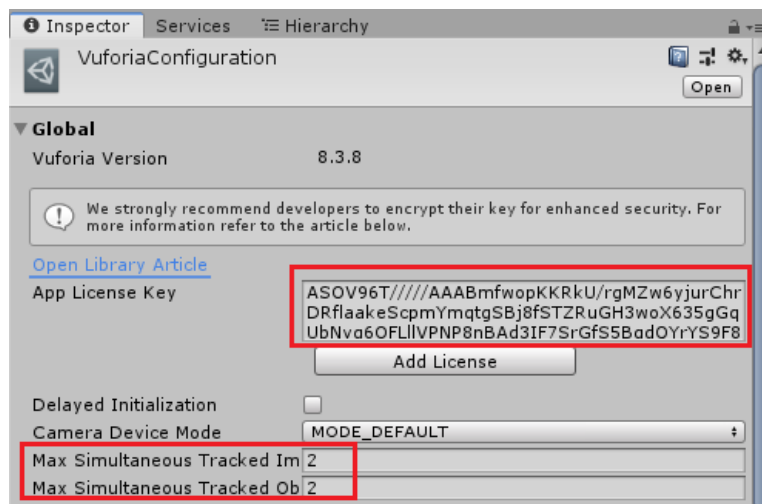
Langkah 6:

Selanjutnya masukkan License Key, ke bagian kotak isian License key di jendela Inspector.



Gambar 4.47: Masukan License Key

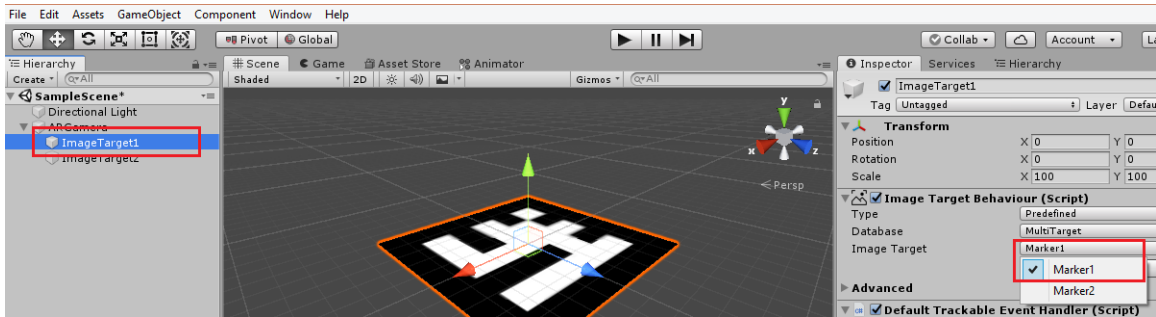
Masukkan *Lcense Key*.



Gambar 4.48: Masukan License Key

Langkah 7:

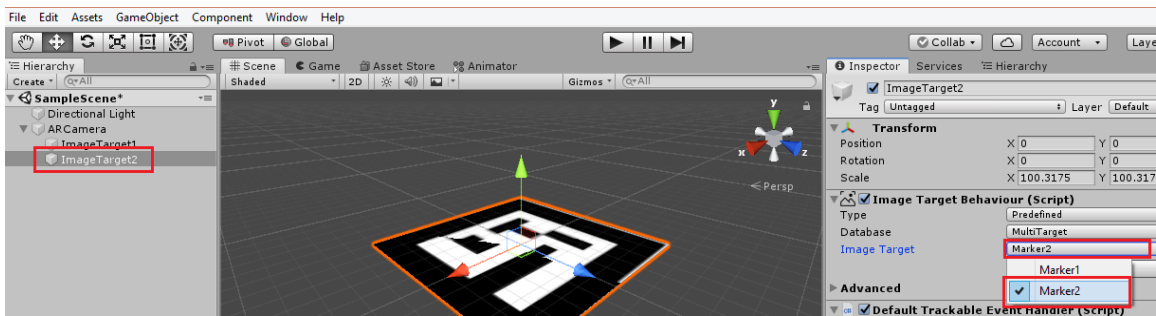
Isi **ImageTarget1** dengan gambar **Marker1**, seperti pada gambar 4.49.



Gambar 4.49: Isi ImageTarget1 Dengan Marker1

Langkah 8:

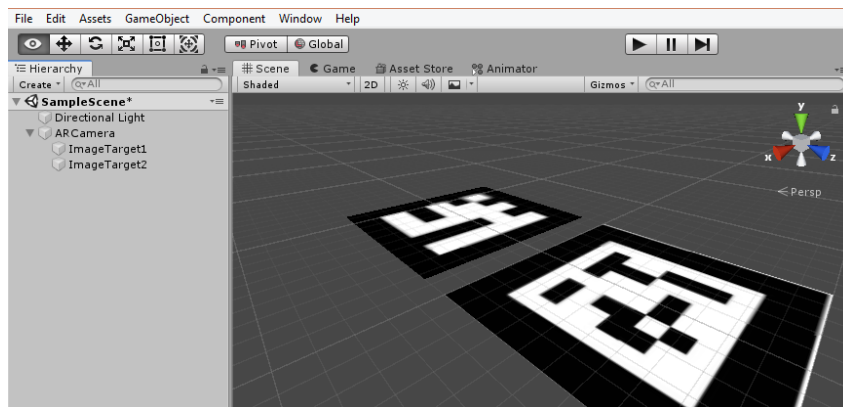
Isi **ImageTarget2** dengan gambar **Marker2**, seperti pada gambar 4.50.



Gambar 4.50: ImageTarget2 Dengan Marker2

Langkah 9:

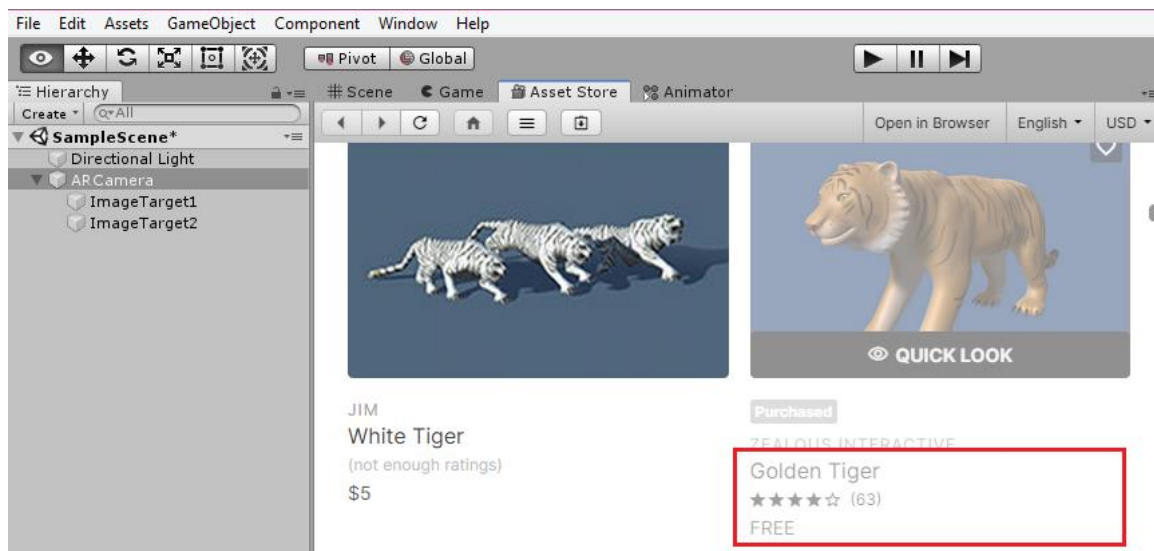
Atur posisi (jarak) ImageTarget1 dan ImageTarget2 dengan cara pindahkan, seperti pada gambar 4.51.



Gambar 4.51: Atur Jarak ImageTarget1 dan ImageTarget2

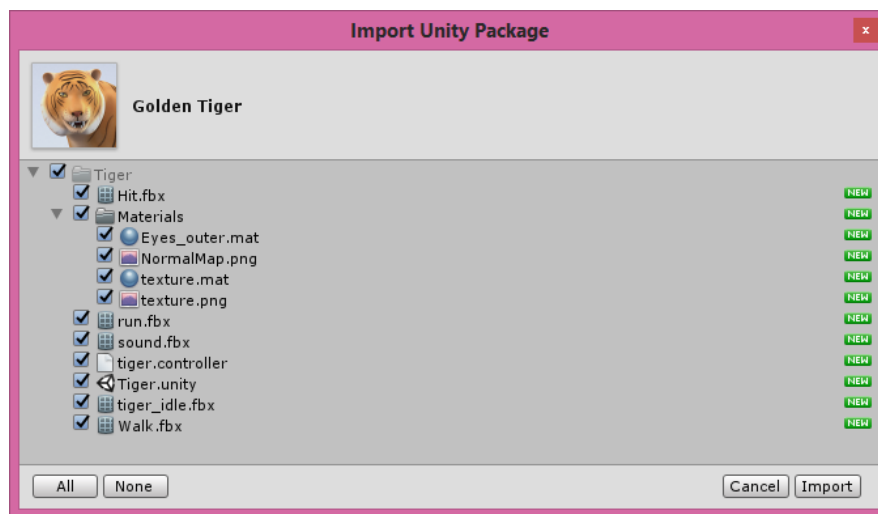
Langkah 10:

Unduh *asset* dari *Asset Store*, *asset* yang diunduh adalah *asset* Tiger.



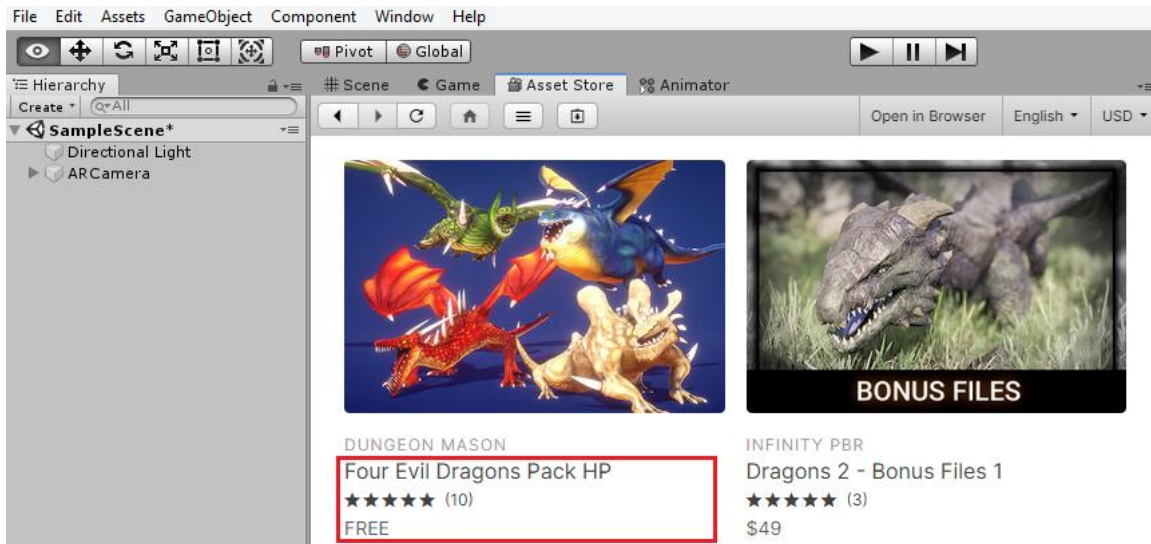
Gambar 4.52: Unduh Asset Tiger

Bila sudah selesai mengunduh *asset*, maka selanjutnya tekan tombol **Import**.



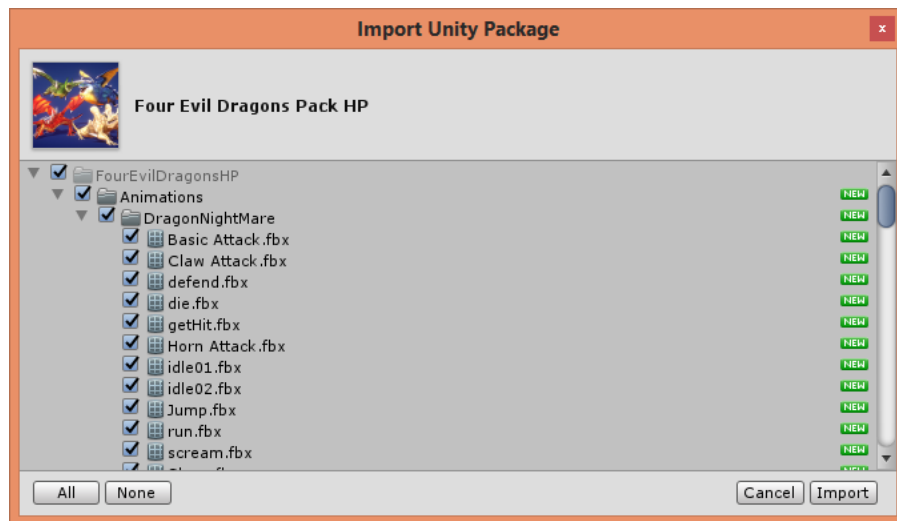
Gambar 4.53: Import Asset Tiger

Unduh *asset* dari *Asset Store*, *asset* yang diunduh adalah *asset* Dragon.



Gambar 4.54: Unduh Asset Dragon

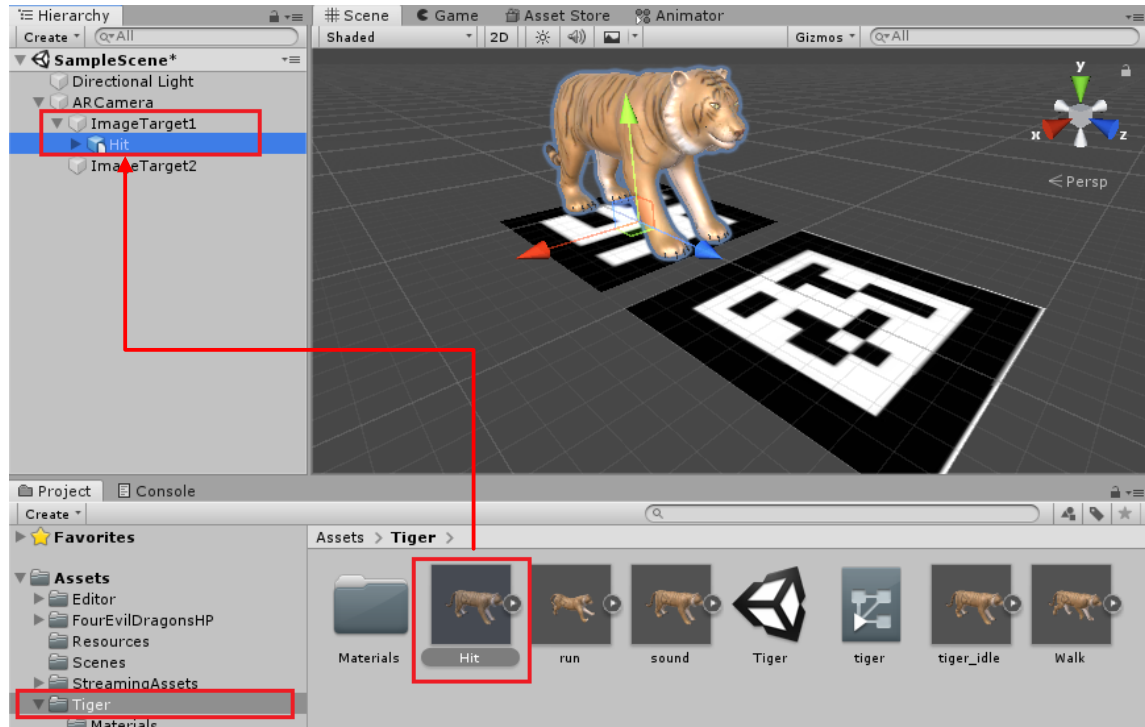
Bila sudah selesai mengunduh *asset*, maka selanjutnya tekan tombol **Import**.



Gambar 4.55: Import Asset Dragon

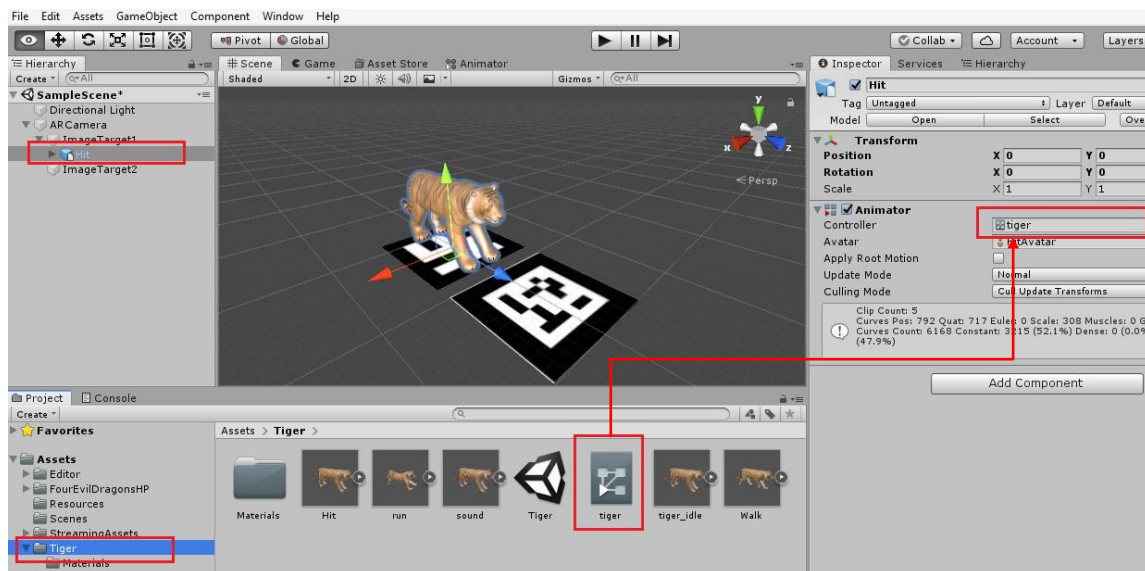
Langkah 11:

Selanjutnya masukkan *asset* **Tiger** ke **ImageTarget1**, dan **Dragon** ke **Imagetarget2**. Skala *asset* bisa disesuaikan dengan keinginan. Sepertinya ukuran Tiger sudah pas skalanya yaitu X=1, Y=1, Z=1. Namun bila mau diubah ukurannya silahkan.



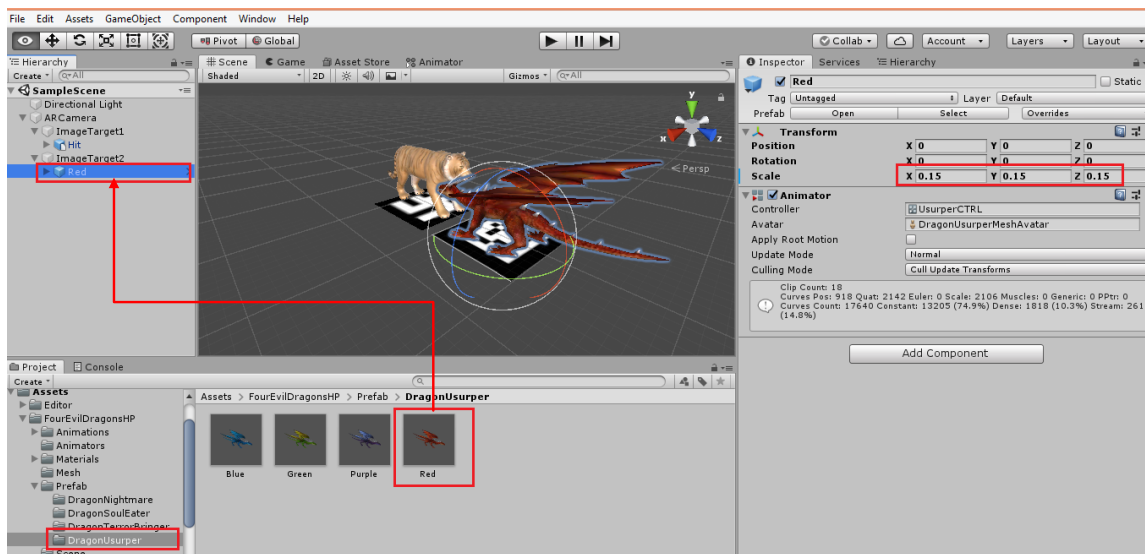
Gambar 4.56: Drag Asset Tiger ke ImageTarget1

Selanjutnya drag animasi **Tiger** ke **Controller**, seperti pada gambar 4.57.



Gambar 4.57: Drag Animasi Tiger ke Controller

Selanjutnya masukan *asset* Dragon ke **ImageTarget2**, ubah skala objeknya menjadi X=0.15, Y=0.15, Z=0.15.



Gambar 4.58: Drag Asset Dragon ke ImageTarget2

Langkah 12:

Selanjutnya coba jalankan aplikasi, maka kita dapat menampilkan dua buah objek dengan menggunakan dua buah marker. Maka seolah-olah macan dan naga sedang bertarung.



Gambar 4.59: Aplikasi Dijalankan

Supaya lebih realistis carikan suara dinosaurus dan suara tiger atau dengan cara merekam suara dinosaurus dan tiger dari video di youtube, rekam suaranya dalam beberapa detik, kemudian simpan dengan format WAV atau MP3. Lakukan langkah yang sama seperti pada memasukkan suara pada latihan helicopter. Selamat mencoba.

4.3 Soal-soal

Pada latihan ini rekan-rekan mahasiswa mengerjakan latihan secara mandiri.

1. Buatlah sebuah aplikasi *Augmented Reality* dengan *asetnya* bebas (bisa di unduh di unity *asset*), menggunakan konsep multiple (multi) target, coba buat file .APK dan jalankan di *smartphone* anda.

Profile



Modul bahan ajar *Augmented Reality* ini didesain sesuai dengan rancangan RPS oleh I Gusti Ngurah Suryantara, S.Kom., M.Kom untuk bahan ajar *Augmented Reality* di Universitas Bunda Mulia di program studi Informatika. Materi ini digunakan untuk kalangan sendiri sebagai bahan ajar Augmented Reality, dilarang: **mengutip, menyebarkan, meng-copy (memperbanyak) dan mengkomersialkan** modul ini tanpa sepengetahuan/izin penulis.

©12022021

Asset, Marker, dan **Icon** dapat di unduh pada google drive saya dengan link:

<https://drive.google.com/drive/folders/1a95qnIhKyVkBkxSH6c5QUa5aYJZIY5ZS?usp=sharing>