



REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202143691, 2 September 2021

Pencipta

Nama : **Aurora Nur Aini, S.Si., M.Sc., Dina Prasetyowati, S.Pd., M.Pd. dkk**

Alamat : Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang, Semarang, JAWA TENGAH, 50272

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Aurora Nur Aini, S.Si., M.Sc., Dina Prasetyowati, S.Pd., M.Pd. dkk**

Alamat : Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang, Semarang, JAWA TENGAH, 50272

Kewarganegaraan : Indonesia

Jenis Ciptaan : **Modul**

Judul Ciptaan : **Pengembangan Aplikasi Augmented Reality Menggunakan Unity (Intermediate)**

Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 30 Agustus 2021, di Semarang

Jangka waktu perlindungan : Berlaku selama hidup Pencipta dan terus berlangsung selama 70 (tujuh puluh) tahun setelah Pencipta meninggal dunia, terhitung mulai tanggal 1 Januari tahun berikutnya.

Nomor pencatatan : 000269562

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL



Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Aurora Nur Aini, S.Si., M.Sc.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang
2	Dina Prasetyowati, S.Pd., M.Pd.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang
3	Dr. Muhammad Prayito, M.Pd.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang
4	Dr. Aryo Andri Nugroho, M.Pd.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang
5	Dr. Ida Dwijayanti, M.Pd.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Aurora Nur Aini, S.Si., M.Sc.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang
2	Dina Prasetyowati, S.Pd., M.Pd.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang
3	Dr. Muhammad Prayito, M.Pd.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang
4	Dr. Aryo Andri Nugroho, M.Pd.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang
5	Dr. Ida Dwijayanti, M.Pd.	Universitas PGRI Semarang Jl. Sidodadi Timur No. 24 - Dr. Cipto Semarang



Deskripsi Produk HaKI

Judul Paten : Pengembangan Aplikasi Augmented Reality Menggunakan Unity
(Intermediate)

Penulis : Aurora Nur Aini, Dina Prasetyowati, Muhammad Prayito,
Aryo Andri Nugroho, Ida Dwijayanti

Nomor Paten : 000269562

Bulan terbit : Agustus 2021

Deskripsi Produk :

Produk ini berupa modul yang berisi tentang pengembangan aplikasi augmented reality menggunakan Software Unity untuk level intermediate. Modul ini merupakan lanjutan dari modul seri pertama yang berjudul Pengembangan Aplikasi Augmented Reality Menggunakan Unity (beginner). Buku ini menjelaskan konsep dasar dari Teknologi Augmented Reality dengan menekankan pada Latihan pembuatan aplikasi berbasis Augmented Reality. Buku ini terdiri dari 7 Bab yang memungkinkan pembaca belajar tentang bagaimana membuat aplikasi Augmented Reality menggunakan Software Unity 3D. Harapannya, pembaca dapat mengembangkan suatu aplikasi Augmented Reality dengan dasar-dasar yang telah dipelajari dari buku ini.



PENGEMBANGAN APLIKASI AUGMENTED REALITY MENGUNAKAN UNITY

INTERMEDIATE



AURORA NUR AINI ▪ DINA PRASETYOWATI ▪
MUHAMMAD PRAYITO ▪ ARYO ANDRI NUGROHO ▪
IDA DWIJAYANTI

PENGEMBANGAN APLIKASI AUGMENTED REALITY MENGUNAKAN UNITY

INTERMEDIATE



AURORA NUR AINI ▪ DINA PRASETYOWATI ▪
MUHAMMAD PRAYITO ▪ ARYO ANDRI NUGROHO ▪
IDA DWIJAYANTI

Kata Pengantar

Puji Syukur penulis panjatkan kepada Allah SWT atas terselesaikannya Buku Pengembangan Aplikasi Augmented Reality Menggunakan Unity (Intermediate). Buku ini merupakan lanjutan dari buku Pengembangan Aplikasi Augmented Reality Menggunakan Unity (Beginner)

Buku ini menjelaskan konsep dasar dari Teknologi Augmented Reality dengan menekankan pada Latihan pembuatan aplikasi berbasis Augmented Reality. Buku ini terdiri dari 7 Bab yang memungkinkan pembaca belajar tentang bagaimana membuat aplikasi AR menggunakan Software Unity 3D. Harapannya, pembaca dapat mengembangkan suatu aplikasi dengan dasar-dasar yang telah dipelajari dari buku ini.

Akhir kata, semoga buku ini dapat bermanfaat bagi pembaca, khususnya dalam bidang Augmented Reality.

Semarang, Agustus 2021

Tim Penulis

Daftar Isi

Cover	i
Kata Pengantar	ii
Daftar Isi	iii
Bab 1 Pemanfaatan Augmented Reality	1
Apa Perbedaan Augmented Reality Dan Virtual Reality?	1
Augmented Reality	1
Virtual Reality	2
Penerapan Augmented Reality.....	3
Bab 2 Apa yang Dibutuhkan untuk Membuat Augmented Reality	8
Unity 3D.....	8
AR Development.....	9
Bab 3 Membuat Aplikasi AR Book	11
Project 1.	12
Project 2.	20
Bab 4 Membuat Aplikasi Main Menu Menggunakan Unity	29
Bab 5 Menggunakan Animasi Objek Sederhana dengan Menggunakan Animator	47
Project 1	48
Project 2	52
Bab 6 Animasi Objek 3D Menggunakan Tombol UI	62
Bab 7 Aplikasi AR dengan Lebih dari 1 Image Target	80
Project 1.	80
Project 2	85
Project 3	89
Project 4.	95
DAFTAR PUSTAKA.....	111

PEMANFAATAN AUGMENTED REALITY

Apa Perbedaan Augmented Reality Dan Virtual Reality?



Teknologi Augmented Reality (AR) dan Virtual Reality (VR) merupakan sebuah teknologi yang banyak dibicarakan pada saat ini. Banyak perusahaan yang mulai memikirkan pemanfaatannya sebagai salah satu media untuk mempromosikan produknya. Namun, taukah Anda apa perbedaan Augmented Reality dan Virtual Reality?

Augmented Reality

Augmented Reality (AR) atau realitas ditambah adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi

Untuk dapat menggunakan teknologi Augmented Reality pengguna dapat menggunakan *hand held display* seperti Smartphone atau menggunakan *head's up display* seperti Hololens atau Meta 2.



Hololens



Meta 2

Virtual Reality

Virtual Reality (VR) adalah teknologi yang memungkinkan penggunanya berinteraksi dengan lingkungan 3D yang dibuat semirip mungkin dengan dunia nyata atau imajinasi.



VR dapat dinikmati dengan memerlukan perangkat khusus yaitu berupa headset VR, seperti Google Cardboard, Oculus Rift atau Samsung Gear VR.



Google CardBoard



Samsung Gear VR



Oculus Rift

Penerapan Augmented Reality

Berikut contoh penerapan Augmented Reality dalam berbagai bidang.

1. Bangunan dan Desain Interior



Puluhan profesi modern berkisar di seputar bangunan, termasuk desain, konstruksi, penjualan, dekorasi, dan lainnya. Mengingat sebagian besar bangunan ini cukup besar dan tidak mudah dipamerkan dengan gambar atau foto 2D, penerapan Augmented Reality dalam bisnis ini dapat menjadi solusi.

Arsitek dan perencana konstruksi dapat menggunakan teknologi AR untuk mempertahankan skala yang akurat dan melakukan pengukuran cepat. Untuk konstruksi dan revisi aktual, Akhirnya, agen property dan desainer interior dapat menggunakan teknologi tersebut untuk menampilkan properti kepada klien, bahkan menawarkan opsi perubahan dekorasi.

2. Energi



Image credit: MidAmerican Energy Co.

Industri energi mencakup semua bisnis dan fasilitas pemerintah yang terkait dengan penyediaan energi bagi masyarakat, mulai dari rig minyak hingga jaringan listrik, perkebunan, dan lainnya. Di antara berbagai fasilitas, AR menangani beberapa kebutuhan umum termasuk pendidikan dan pelatihan yang aman, komunikasi yang lebih baik, dan bantuan jarak jauh. Selain itu, pekerja dapat menggunakan fitur kamera / pindai pada perangkat keras mereka untuk mengidentifikasi dan mendapatkan informasi mendetail tentang komponen dan elemen infrastruktur.

3. Manufaktur



Dalam industri manufaktur, AR memiliki berbagai macam manfaat. Pertama, ini dapat membantu pekerja yang terlibat dalam prototipe R&D dan menguji desain dalam pengaturan spasial yang realistis. Sedangkan untuk pekerja di bidang

perakitan, AR dapat membantu dalam pelatihan yang lebih aman, hingga membantu dalam bekerja lebih efisien. Misalnya, mereka dapat menggunakan perangkat lunak untuk memindai produk manufaktur dengan cepat dan menarik informasi / skema yang diperlukan atau mengirimkan data.

4. Otomotif



Image credit: BMW Group

Ada banyak kemiripan antara industri Otomotif dan Manufaktur. Bedanya adalah otomotif tidak hanya memproduksi mobil, tetapi juga berfokus pada penjualan, upgrade, dan penyediaan layanan tambahan seperti peningkatan perangkat lunak dan logistik. Selain penggunaan produksi standar seperti pelatihan jarak jauh dan pembuatan prototipe, AR juga memudahkan pekerja untuk memamerkan model, meningkatkan pengalaman berkendara melalui panduan visual, dan memvisualisasikan data mengemudi seperti cuaca dan kondisi jalan.

BMW Group menggunakan kacamata AR dan aplikasi khusus untuk mempercepat pembuatan prototipe kendaraan dan suku cadang mobil. Aplikasi ini terutama difokuskan untuk menentukan desain mana yang cocok dengan rangka mobil dalam berbagai sudut dan metode. Ini juga mengukur waktu perakitan teoritis untuk lebih mudah menemukan desain yang paling layak. Perusahaan mengklaim bahwa pendekatan imersif ini mempercepat validasi modul kendaraan hingga satu tahun.

5. Pendidikan



Image credit: Ryerson University

Salah satu keluhan terbesar tentang sistem pendidikan di seluruh dunia berkaitan dengan materi dan pendekatan pengajaran yang kurang menarik dan sudah ketinggalan zaman. Dalam hal ini, AR berfungsi sebagai instrumen yang sempurna dalam menyampaikan informasi dan membantu siswa memperoleh keterampilan baru. Lebih khusus lagi, ini mampu menjadikan pembelajaran dapat diakses di mana saja, tanpa memerlukan buku teks atau perangkat keras khusus. Siswa hanya menggunakan aplikasi dengan semua informasi yang disampaikan melalui visual 3D yang menarik.

6. Kesehatan



Image credit: Hospital for Special Surgery

Dalam perawatan kesehatan, AR terutama berperan sebagai penghubung dokter dan pasien, serta hal-hal standar seperti pelatihan dan pendidikan profesional. Dalam beberapa kasus, ini dapat membantu untuk operasi kehidupan nyata yang membutuhkan ketelitian, seperti pembedahan dan kedokteran gigi. Selain itu, ini dapat membuat informasi medis lebih dapat diakses oleh tunanetra, dan memvisualisasikan pemindaian kesehatan (MRI, X-ray) dengan cara yang lebih mudah dipahami.

7. Penjualan dan Pemasaran



Kategori ini berlaku untuk semua industri yang terdaftar sebelumnya, karena kebanyakan dari industri fokus pada promosi produk atau jasa. Selain itu, sebagian besar perusahaan di industri memiliki departemen pemasaran yang dapat memanfaatkan integrasi AR. Diantaranya menampilkan produk di konfigurator / showroom 3D atau solusi yang lebih canggih seperti panduan pengguna AR dengan dukungan / pemecahan masalah jarak jauh. Terdapat banyak penerapan di bidang ini untuk solusi lain.

APA YANG DIBUTUHKAN UNTUK MEMBUAT AUGMENTED REALITY ?

Membuat aplikasi mobile yang berhubungan dengan AR merupakan salah satu ide bisnis yang cemerlang namun harus dibuat dengan baik. Ada beberapa software yang dapat digunakan untuk membangun aplikasi Augmented Reality diantaranya Assemblr, Geogebra, Lens Studio, Adobe Aero, Unity 3D dan lain-lain.

Pada pembahasan berikutnya kita akan menggunakan Unity 3D untuk membuat aplikasi Augmented Reality.

Unity 3D



Unity 3D adalah salah satu game engine terbaik yang dikembangkan oleh Unity Technologies dan bersifat cross-platform, artinya anda dapat membuat serta merilis game kita ke berbagai platform terkenal, seperti Windows, Linux, Mac OS, Android, iOS, PS3, PS4, Xbox One, dan lain-lain. Dengan Unity, anda dapat membuat game sesuai keinginan, misalnya 2D dan 3D. Selain untuk membuat game, Unity 3D juga dapat digunakan untuk membangun aplikasi Augmented Reality maupun Virtual Reality. Setiap software tentu memiliki kelebihan dan kekurangan. Berikut kelebihan dan kekurangan Unity 3D

Kelebihan :

1. Gratis, tidak perlu mengeluarkan biaya. Cukup dengan Unity versi *Free (Personal Edition)*, maka kita sudah bisa membuat karya *game*.
2. *Cross-platform*, artinya *project game* dapat dirilis ke berbagai *platform* terkenal.
3. UI yang ditawarkan cukup *user-friendly*, sehingga cocok untuk pemula.
4. Banyak fitur yang ditawarkan.
5. Dokumentasi atau tutorial yang banyak dan gratis untuk dipelajari resmi dari Unity. Silakan Anda dapat mengunjungi situs dokumentasi resminya.
6. Memiliki *assets store*, di mana kita dapat mencari *assets 2D / 3D* gratis maupun berbayar.
7. Mempunyai *text editor* bernama Mono Develop. Mirip seperti visual code, Mono Develop ini dapat digunakan untuk *coding game* dan sudah terintegrasi langsung ke Unity Engine.
8. Ringan, yakni bisa dijalankan di PC yang tidak terlalu *High-Spec*.
9. Proses *desain level game* menjadi lebih mudah. Unity menawarkan banyak alat untuk desain *level game*, seperti *Terrain Editor*, *Responsive UI Editor*, dan masih banyak lagi.

Kekurangan :

1. Butuh penyesuaian yang agak kompleks pada saat ingin mengembangkan *game 2D* tanpa bantuan *plugin*.
2. Perlu penyesuaian mengikuti gaya *component based*.
3. *Cache memory* yang digunakan lumayan banyak.

AR Development

Untuk membuat Augmented Reality dengan menggunakan Unity 3D diperlukan sebuah perangkat alat AR Development. Ada banyak AR Development yang dapat digunakan, misalnya Vuforia, Google ARCore, Apple ARKit, MaxST dan

Wikitide. Pembahasan kita di bab-bab berikutnya akan memanfaatkan Vuforia yang dapat diakses melalui <https://developer.vuforia.com/>



Vuforia adalah salah satu perangkat AR development yang paling terkenal dengan banyak fitur yang menarik. Vuforia menyediakan versi beta yang dapat digunakan gratis dengan fitur lebih sedikit dan memiliki watermark. Disisi lain, versi premiumnya memungkinkan anda untuk terhubung dengan banyak fitur dan tanpa watermark. Aplikasi ini kompatibel dengan Android, iOS, Windows dan Unity.

MEMBUAT APLIKASI AR BOOK



Pernahkah Anda melihat sebuah buku Augmented Reality?

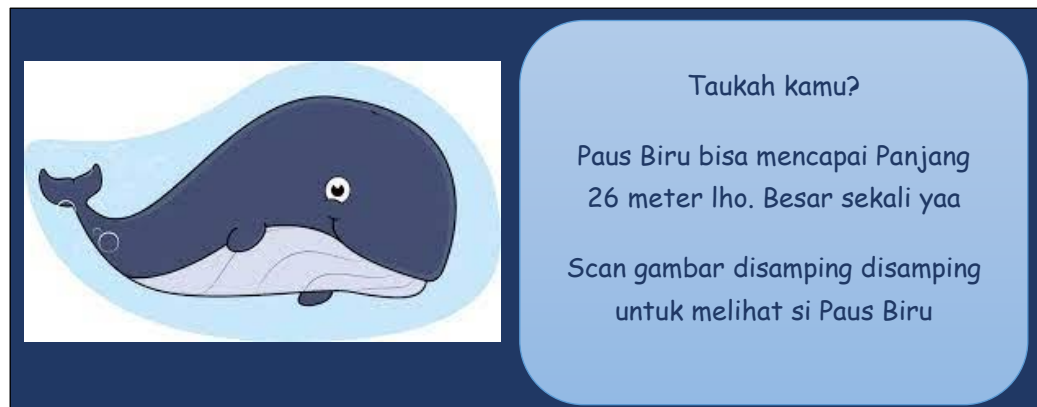
Untuk dapat menggunakan buku Augmented Reality beserta vitur 3D-nya pembaca harus menginstall

suatu aplikasi yang ditentukan kemudian menyorot bagian-bagian tertentu (disebut marker) pada buku tersebut untuk memunculkan bentuk-bentuk 3D-nya. Tentu saja tidak mungkin jika pembaca harus menginstall satu aplikasi untuk setiap gambar/marker.

Pada bagian ini kita akan membuat aplikasi yang dapat membaca beberapa marker/image target seperti pada sebuah buku Augmented Reality. Aplikasi ini kita beri nama **AR Book**. Untuk membuat AR Book tentu kita memerlukan beberapa gambar yang nantinya akan kita pakai sebagai marker/image target. Gambar-gambar yang bisa digunakan haruslah berekstensi .jpg atau .png RGB atau Grayscale dengan ukuran maksimal 2 MB dan lebar minimal 320 pixel. Contohnya,



Jika buku Augmented Reality ini akan dibuat dalam versi cetak, maka gambar yang menjadi target marker tersebut harus disertakan agar dapat di scan oleh pembaca yang telah menginstall aplikasi yang kita buat.

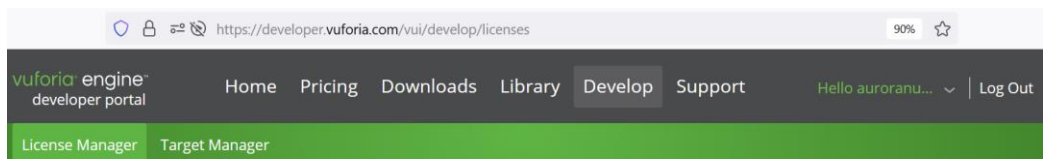


Taukah kamu?
Paus Biru bisa mencapai Panjang 26 meter lho. Besar sekali yaa
Scan gambar disamping disamping untuk melihat si Paus Biru

Project 1.

Setelah Anda memiliki gambar-gambar yang akan dijadikan image target, berikutnya adalah mengunggahnya ke Vuforia Developer.

1. Buka laman Vuforia Developer – Log in. Setelah Log in kemudian masuk pada tab Develop. Pada bagian ini ada Lisence Manager dan Target Manager. Klik pada Lisence Manager kemudian Get Development Key.



License Manager

Get Development Key

Buy Deployment Key

Create a license key for your application.

Ketikkan nama lisensi (Anda dapat mengubah nama lisensi nanti). Kemudian centang pada bagian “By checking this box, I acknowledge that this licence key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).”. Selanjutnya klik Confirm.

[Back To License Manager](#)

Add a free Development License Key

License Name*
Multi Target AR

You can change this later

License Key

Develop

Price: No Charge

Reco Usage: 1,000 per month

Cloud Targets: 1,000

VuMark Templates: 1 Active

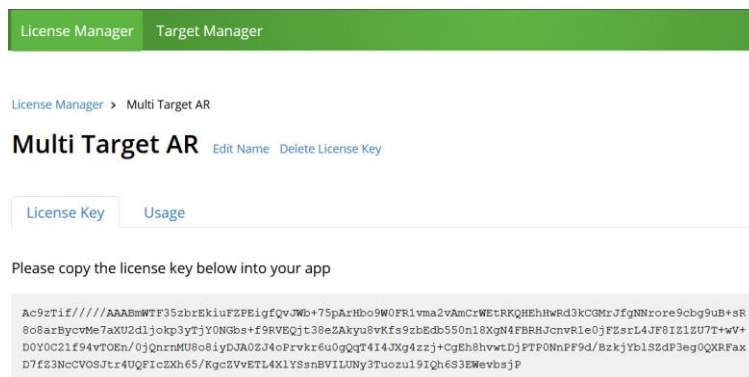
VuMarks: 100

By checking this box, I acknowledge that this license key is subject to the terms and conditions of the [Vuforia Developer Agreement](#).

Cancel

Confirm

Kemudian buka Lisence Name yang baru saja Anda buat.



```
Ac9zTif/////AAABmWTF35zbrEkiuFZFEIgfQvJWb+75pArHbo9W0FR1vma2vAmCrWELRQHEhHwRd3kCGMrJfgNNrore9cbg9uB+sR
8o8arBycvMe7axU2d1jokp3yTjYONGbs+f9RVEQjt38eZAkya8vKfs9zbEdb550n18XgN4FBRHJcnvRle0jFZsrL4Jf81E12U7T+vw+
D0Y0c21f94wTOEn/0jQnrmU8o8iYdJA02J4oPrvkr6u0gQqT414JXg4zrzj+CgEh8hvtDjPTF0NnPF9d/BzkjYb1SzdP3eg0QXRfAx
D7fE3NcCVOSJtr4UQFic2Xh65/Rgc2VvETL4K1YSsnBVILUNy3Tuozu19Iqh6S3EwvbsjP
```

Klik sekali pada lisence key untuk meng-copy-nya.

2. Berikutnya kita unggah gambar-gambar untuk dijadikan target marker.

Klik pada bagian **Target Manager** – Klik **Add Database** – Ketikkan **Database Name** – Pada bagian **Type** pilih **Device** – Klik **Create**

Create Database

Database Name *
Multi_Target_AR

Type:

Device
 Cloud
 VuMark

3. Klik Database yang baru Anda buat, kemudian Add Target sesuai dengan jumlah target marker yang akan digunakan




License Manager Target Manager

Target Manager > Multi_Target_AR

Multi_Target_AR [Edit Name](#)

Type: Device

Targets (3)

<input type="checkbox"/> Target Name	Type	Rating ⓘ	Status ▾	Date Modified
<input type="checkbox"/>  Forest	Single Image	★★★★★	Active	Jul 24, 2021 22:57
<input type="checkbox"/>  Car-Cartoon	Single Image	★★★★★	Active	Jul 24, 2021 22:57
<input type="checkbox"/>  air-plane	Single Image	★★★★★	Active	Jul 24, 2021 22:56

4. Klik All Database (All) untuk mengunduh semua target marker.

Download Database

3 of 3 active targets will be downloaded

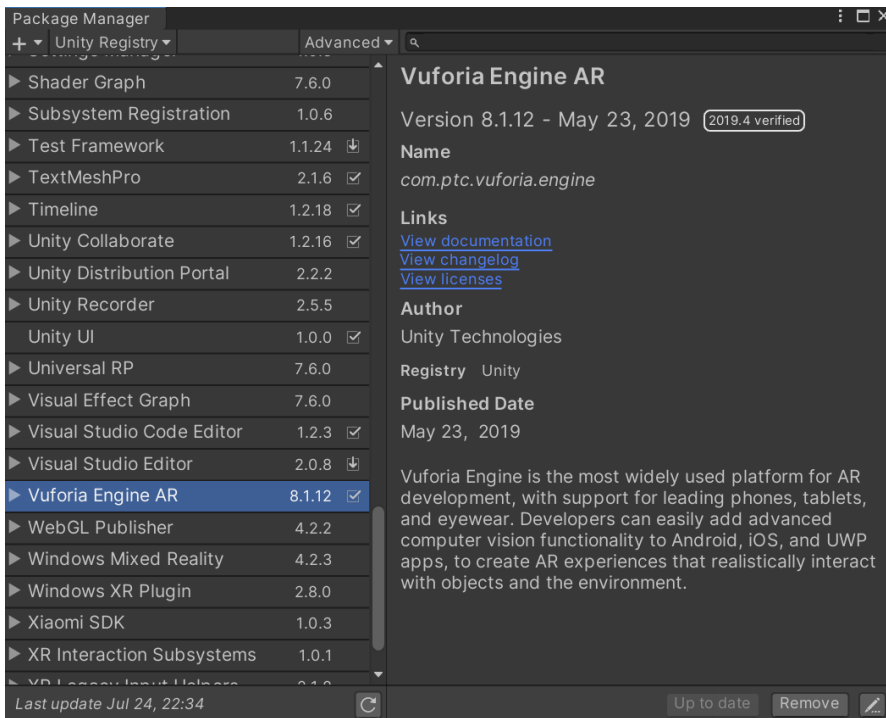
Name:
Multi_Target_AR

Select a development platform:

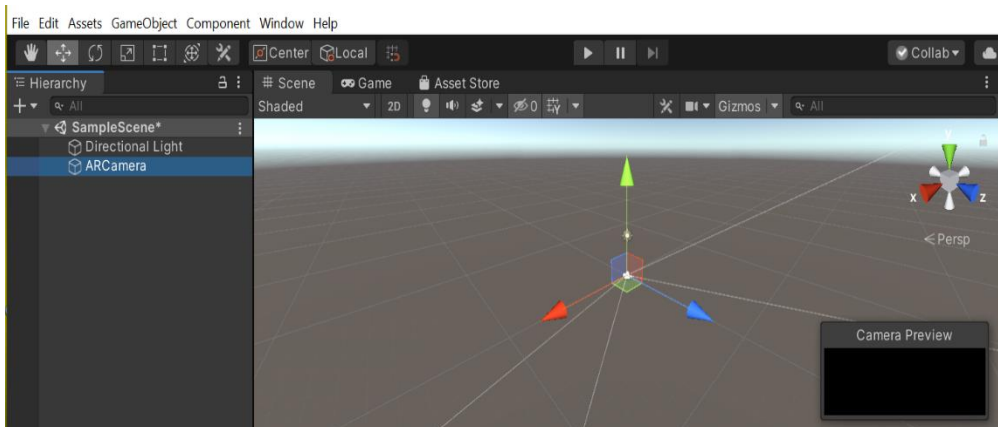
- Android Studio, Xcode or Visual Studio
 Unity Editor

5. Pilih Unity Editor kemudian klik Download

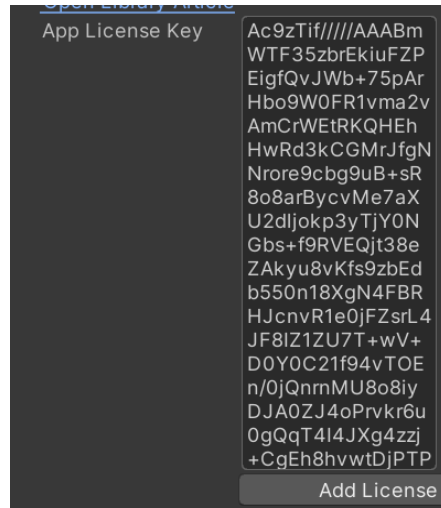
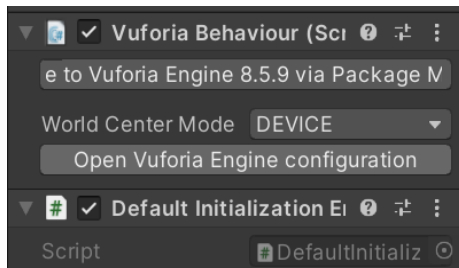
6. Berikutnya, buka project baru di Unity. Install Vuforia Engine AR melalui Windows – Package Manager – In Registry



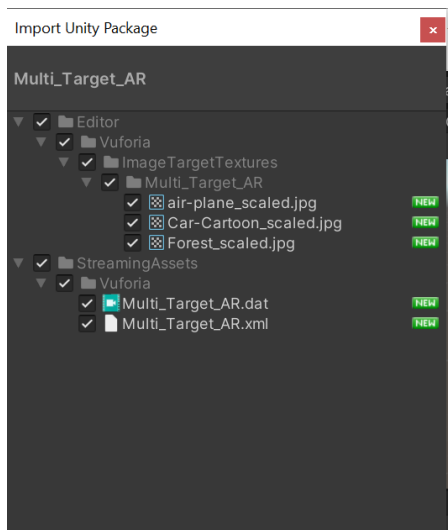
7. Berikutnya, hapus main camera dari Tab Hierarchy kemudian klik kanan pada Tab hierarchy – Vuforia Engine – AR Camera



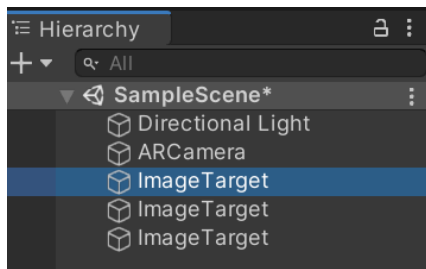
8. Masukkan lisence key yang dicopy dari Vuforia Developer ke dalam AR Camera dengan cara klik pada AR Camera pada Tab Hierarchy – Lihat pada Inspector panel – Open Vuforia Engine Configuration – Paste Lisence Key ke bagian App Lisence Key



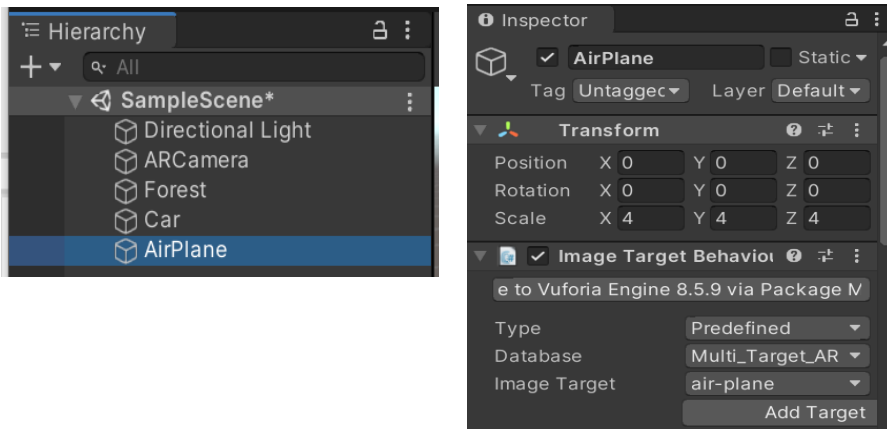
9. Buka file target marker database yang diunduh dari Vuforia dengan cara double klik pada file yang telah diunduh tersebut kemudian klik import



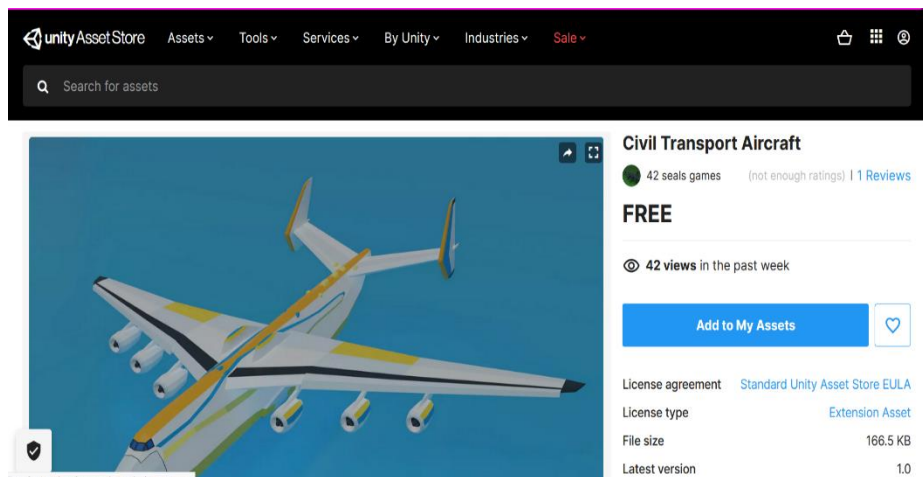
10. Tambahkan 3 Image Target pada tab Hierarchy (sesuai dengan jumlah image target) dengan cara klik kanan dalam tab Hierarchy – Vuforia Engine – Image



11. Ganti nama masing-masing image target pada tab inspector, kemudian ganti image target yang bersesuaian pada bagian Image Target Behaviour



12. Berikutnya tambahkan objek 3D pada masing-masing image target.
13. Klik pada tab Assets Store kemudian cari objek yang bersesuaian. Download dan import asset yang dipilih. Asset tersebut akan masuk ke dalam tab Assets. Atau Anda bisa download Asset melalui laman <https://assetstore.unity.com/>



14. Klik Add to My Assets. Akan muncul Asset Store Terms of Service

Asset Store Terms of Service and EULA

Last updated: July 31, 2020

1. Background

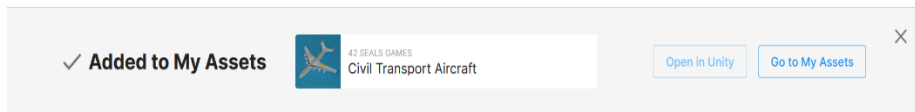
1.1

The Unity Asset Store ("Unity Asset Store") is owned and operated by Unity Technologies ApS (company no. 30 71 99 13), Niels Hemmingsens Gade 24, 1153 Copenhagen K, Denmark ("Unity"). Your use of the Unity Asset Store is governed by a legal agreement between you and Unity consisting of these Asset Store Terms of Service ("Terms") which you accept by checking the box indicating your acceptance of these Terms and/or registering as a user of

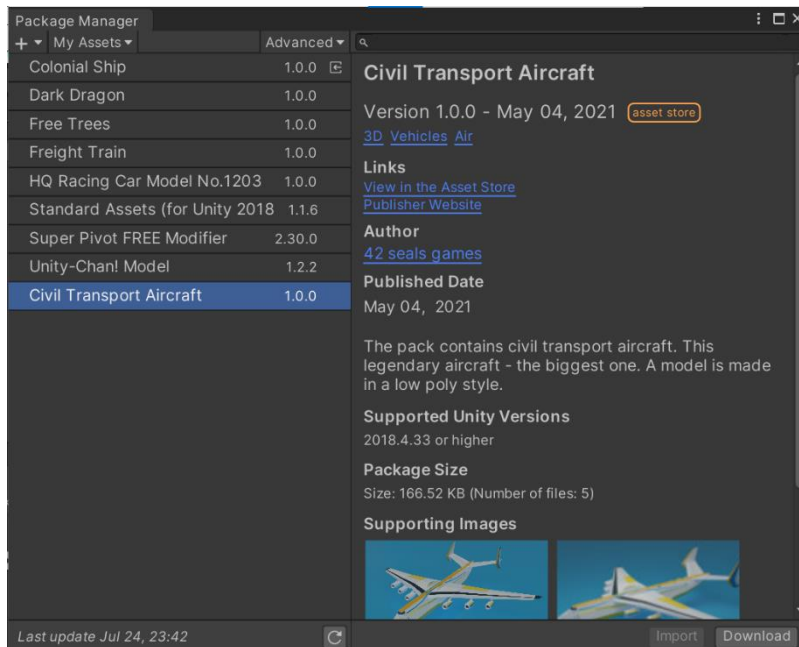
Cancel

Accept

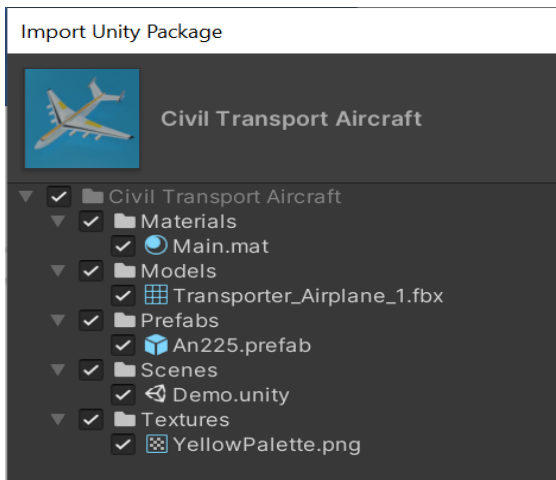
15. Klik Accept



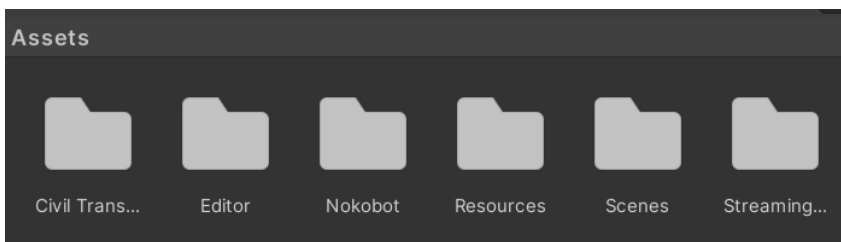
16. Klik Open in Unity. Maka asset akan terbuka pada tab Package Manager Unity



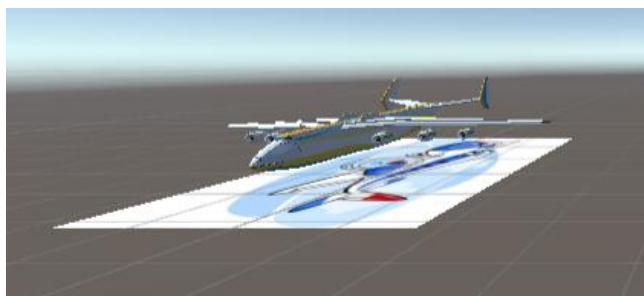
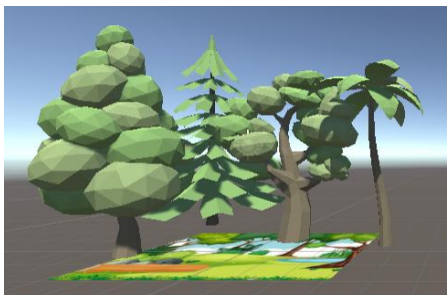
17. Klik Download – kemudian klik Import



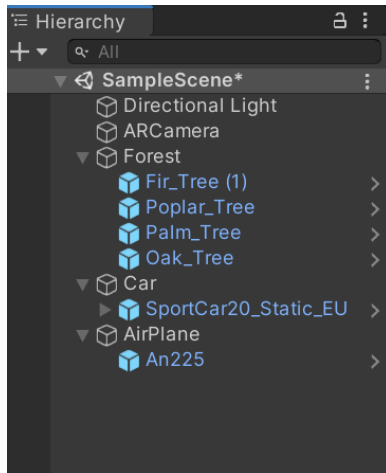
18. Asset yang telah berhasil diimport akan berada pada tab Assets



19. Buka Folder yang baru saja diimport pada Assets – temukan prefabs – Drag isi dari prefabs tersebut ke Tab Scene. Sesuaikan posisi dan scalanya



20. Pada tab hierarchy, masukkan 3D model ke dalam tiap-tiap image target sesuai dengan parent and child-nya

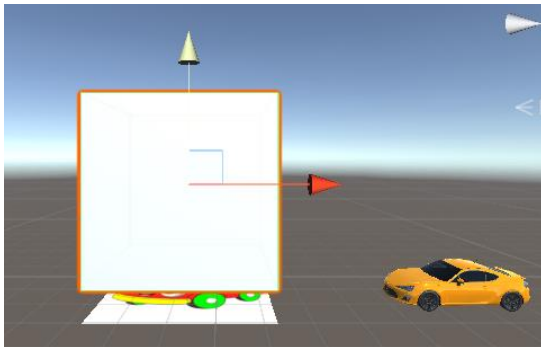


21. Siapkan marker Anda dan klik Play Button untuk mencoba.

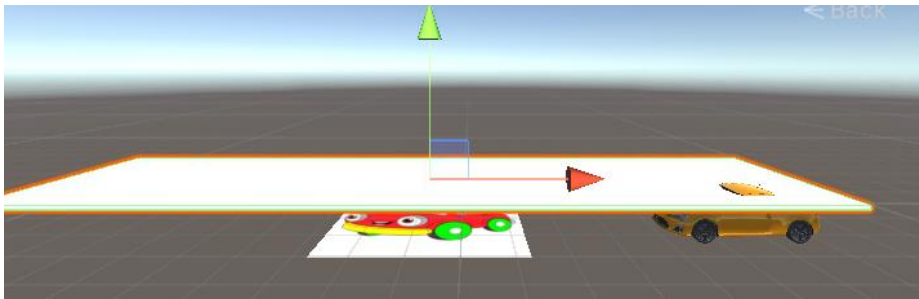
Project 2.

Pada bagian ini kita akan membuat mobil dapat bergerak pada jalan raya jika target marker disorot oleh kamera

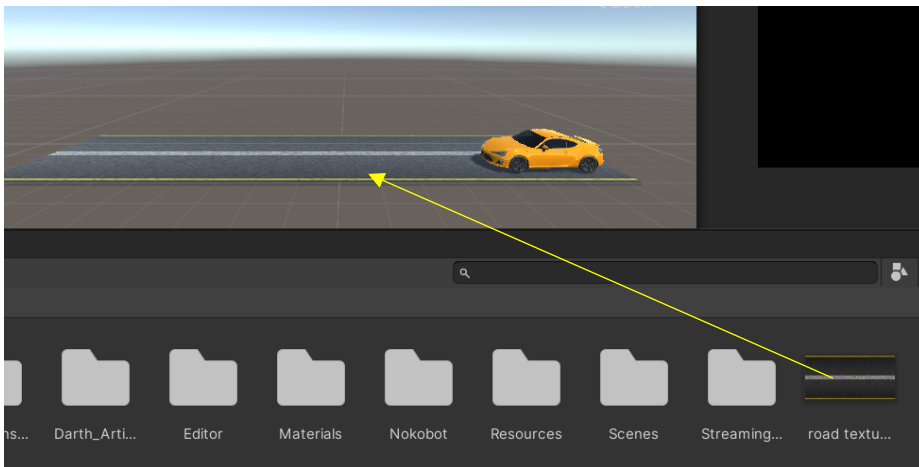
1. Klik kanan pada target marker Car – 3D object – Cube



2. Ubah ukuran Cube sehingga berbentuk seperti jalan raya



3. Temukan road texture dari mesin pencari (missal Google). Drag ke Assets kemudian drag ke Cube



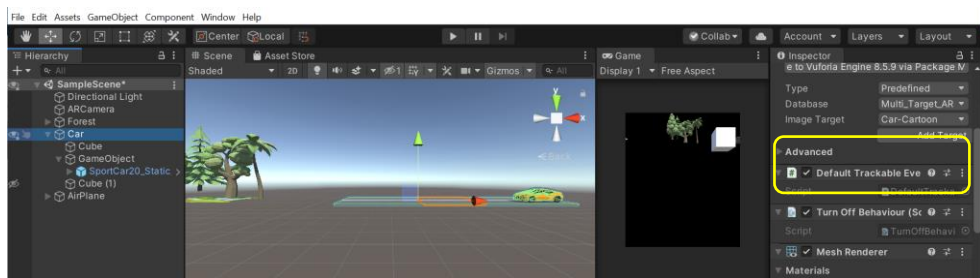
4. Berikutnya, buat script baru dengan cara klik kanan pada Assets – Create – C# Script. Beri nama CarScript. Kemudian ketikkan Script berikut.

```

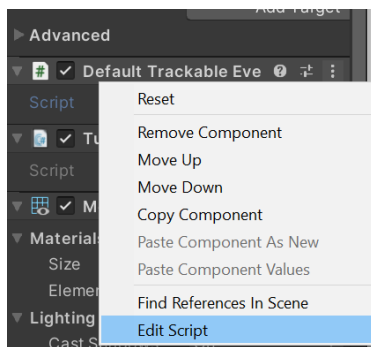
CarScript.cs [X]
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class CarScript : MonoBehaviour
6  {
7      Vector3 moveCar;
8      // Start is called before the first frame update
9      void Start()
10     {
11         moveCar = new Vector3(-2, 0, 0);
12     }
13
14     // Update is called once per frame
15     void Update()
16     {
17         transform.Translate(moveCar * Time.deltaTime);
18     }
19 }
20

```

5. Simpan Script kemudian Kembali ke Unity. Drag Script ke Car. Tekan tombol Play untuk mencoba. Jika Car bergerak ke arah yang tidak tepat, ubah-ubah nilai Vector3 pada Script.
6. Sampai disini kita telah berhasil menggerakkan object dengan translasi. Namun jika camera AR dijalankan, walaupun target marker belum terdeteksi object sudah mulai bergerak. Object akan terus bergerak sampai menghilang dari jangkauan layer. Object hanya akan terlihat Kembali jika kita me-restart camera AR. Untuk mengatasi masalah tersebut, kita harus mengubah script pada bagian Default Trackable Event Handler.
7. Klik pada sembarang GameObject. Kemudian pada Inspector lihat pada Default Trackable Event Handler



8. Buka Default Trackable Event Handler dengan cara klik pada tanda titik 3 – Edit Script



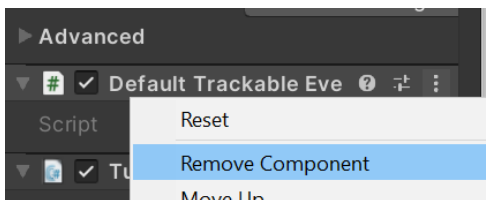
9. Akan terbuka Script Default Trackable Event Handler

```

1  1  /*=====
2  2  Copyright (c) 2017 PTC Inc. All Rights Reserved.
3  3
4  4  Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
5  5  All Rights Reserved.
6  6  Confidential and Proprietary - Protected under copyright and other laws.
7  7  =====*/
8  8
9  9  using UnityEngine;
10 10 using Vuforia;
11 11
12 12 <summary>
13 13 A custom handler that implements the ITrackableEventHandler interface.
14 14
15 15 Changes made to this file could be overwritten when upgrading the Vuforia version.
16 16 When implementing custom event handler behavior, consider inheriting from this class instead.
17 17 </summary>
18 18 public class DefaultTrackableEventHandler : MonoBehaviour, ITrackableEventHandler
19 19 {
20 20     PROTECTED_MEMBER_VARIABLES
21 21
22 22
23 23     UNITY_MONOBEHAVIOUR_METHODS
24 24
25 25
26 26     PUBLIC_METHODS
27 27
28 28     PROTECTED_METHODS
29 29 }
30 30
31 31
32 32
33 33

```

10. Kita tidak bisa mengubah isi dari Default Trackable Event Handler ini. Tapi kita bisa membuat Copy dari isi Script ini kemudian mengedit Script copy nya
11. Buat Script baru, beri nama DefaultTrackableEventHandler. Copykan isi Default Trackable Event Handler yang asli ke Script yang baru tersebut
12. Hapus Default Trackable Event Handler yang asli.



Gantikan dengan yang telah di copy dengan cara drag Script dari Assets ke GameObject yang dikehendaki.

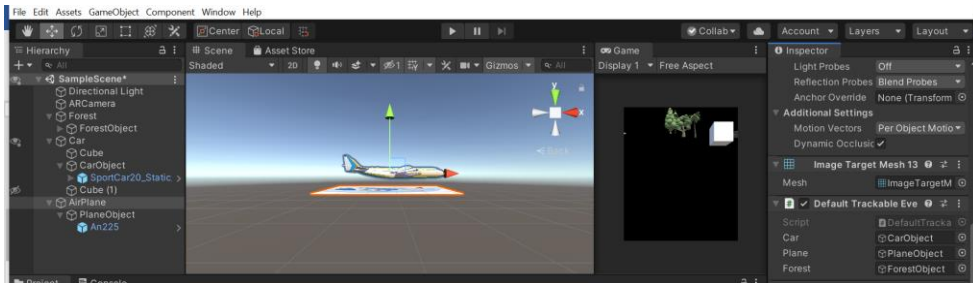
13. Ketikkan script berikut sebelum protected virtual void Start()

```

28 28 #region UNITY_MONOBEHAVIOUR_METHODS
29 29
30 30 public GameObject car, plane, forest;
31 31
32 32 protected virtual void Start()
33 33 {

```

14. Berikutnya drag Object ke Default Trackable Event Handler pada Inspector Panel



15. Edit pada Script Default Trackable event Handler pada bagian “protected virtual void OnTrackingFound()”

```
protected virtual void OnTrackingFound()
{
    var rendererComponents = GetComponentsInChildren<Renderer>(true);
    var colliderComponents = GetComponentsInChildren<Collider>(true);
    var canvasComponents = GetComponentsInChildren<Canvas>(true);

    // Enable rendering:
    foreach (var component in rendererComponents)
        component.enabled = true;

    // Enable colliders:
    foreach (var component in colliderComponents)
        component.enabled = true;

    // Enable canvas':
    foreach (var component in canvasComponents)
        component.enabled = true;
    car.SetActive(true);
    plane.SetActive(true);
    forest.SetActive(true);
}
```

Selanjutnya pada bagian “protected virtual void OnTrackingLost()”


```

protected virtual void OnTrackingLost()
{
    var rendererComponents = GetComponentsInChildren<Renderer>(true);
    var colliderComponents = GetComponentsInChildren<Collider>(true);
    var canvasComponents = GetComponentsInChildren<Canvas>(true);

    // Disable rendering:
    foreach (var component in rendererComponents)
        component.enabled = false;

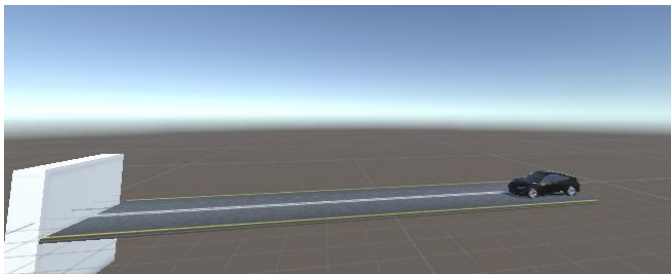
    // Disable colliders:
    foreach (var component in colliderComponents)
        component.enabled = false;

    // Disable canvas':
    foreach (var component in canvasComponents)
        component.enabled = false;

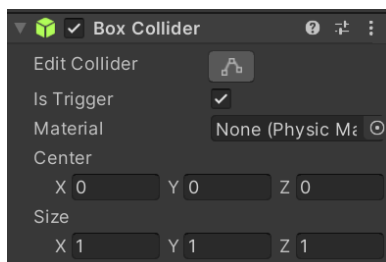
    car.SetActive(false);
    plane.SetActive(false);
    forest.SetActive(false);
}

```

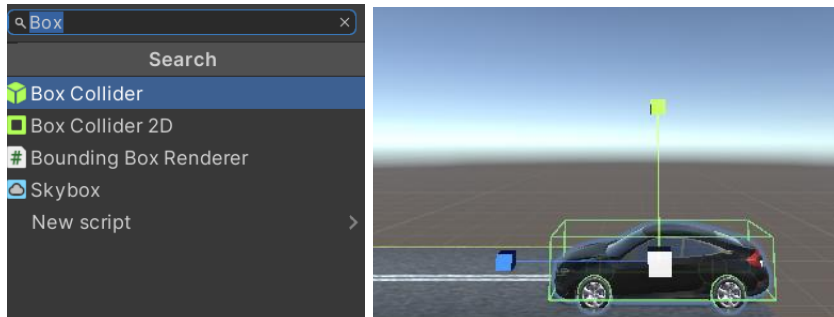
16. Selanjutnya bagaimana untuk membuat object kembali ke posisi awal. Tambahkan object di batas akhir gerakan objek, missal di ujung jalan raya. Tambahkan cube, letakkan di ujung jalan



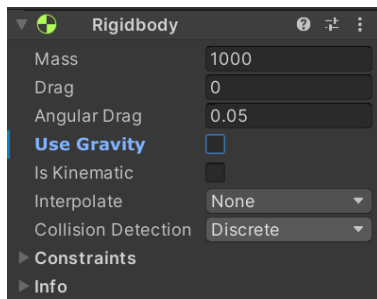
17. Klik pada cube. Kemudian pada Inspector Panel lihat pada bagian Box Collider. Beri tanda centang pada “is Trigger” untuk menjadikan box tersebut Trigger pada saat object Car dan Cube bertumbukan



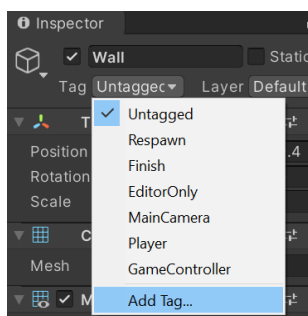
18. Selanjutnya kita tambahkan box Collider pada object Car supaya cube dapat di-trigger oleh collision (tumbukan). Caranya, klik pada GameObject (CarObject) kemudian pada Inspector klik Add Component – Box Collider



19. Masih pada GameObject, lihat pada Inspector – Rigidbody. Hilangkan tanda centang pada bagian Use Gravity

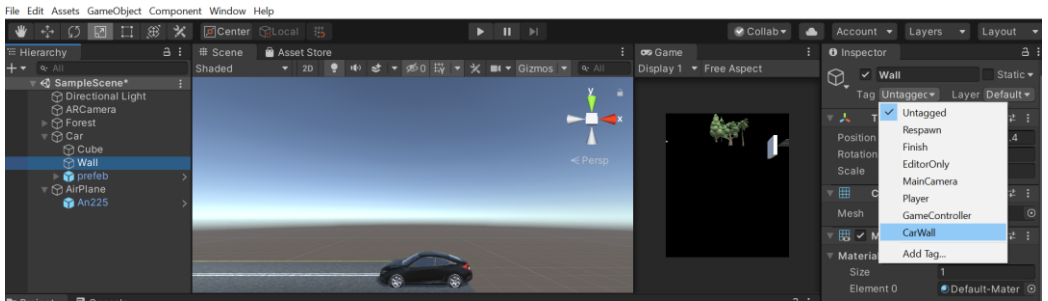


20. Berikutnya klik pada Wall. Kemudian pada Inspector klik pada tanda panah di sebelah Tag – pilih Add Tag



Beri nama Car Wall – kemudian Save

21. Masih pada Wall, pastikan pada bagian Tag dipilih CarWall



22. Buat Script baru dengan nama ResetPosition, kemudian ketikkan Script berikut

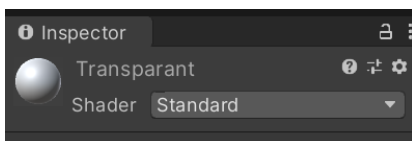
```

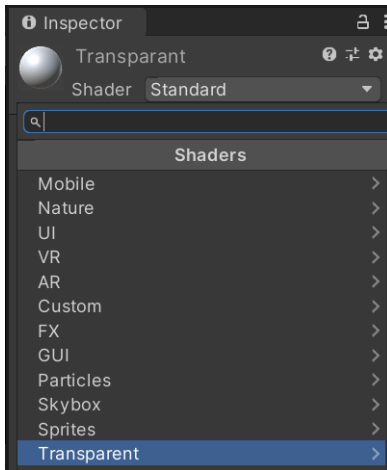
ResetPosition.cs
Miscellaneous Files
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class ResetPosition : MonoBehaviour
6 {
7     Vector3 originalPosition;
8     // Start is called before the first frame update
9     void Start()
10    {
11        originalPosition = transform.localPosition;
12    }
13
14    // Update is called once per frame
15    void Update()
16    {
17    }
18
19    private void OnTriggerEnter(Collider collide)
20    {
21        if(collide.gameObject.CompareTag("CarWall"))
22        {
23            transform.localPosition = originalPosition;
24        }
25    }
26
27 }
28
  
```

23. Terapkan Script pada Object 3D dengan cara drag Script dari Assets ke Object 3D pada hierarchy

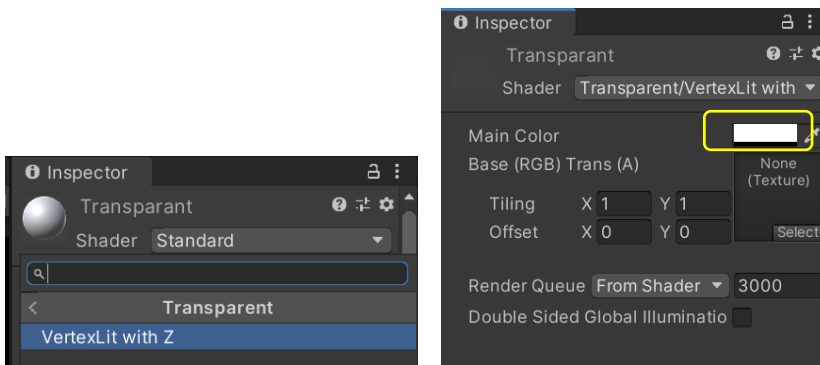
24. Buat material transparant untuk membuat Wall menjadi transparant. Buat material baru dengan cara klik kanan pada Assets – Create – Material – beri nama Transparant

25. Pada inspector Material Transparant, klik pada shader kemudian klik pada tanda panah ke bawah, pilih Transparent

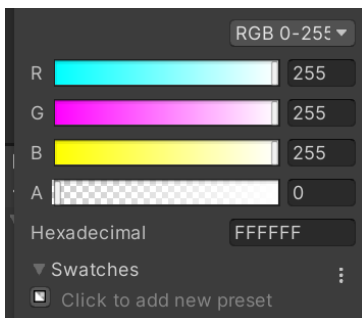




26. Klik VertexLit with Z kemudian klik warna putih pada Main color



27. Drag pada bagian A sehingga bernilai 0



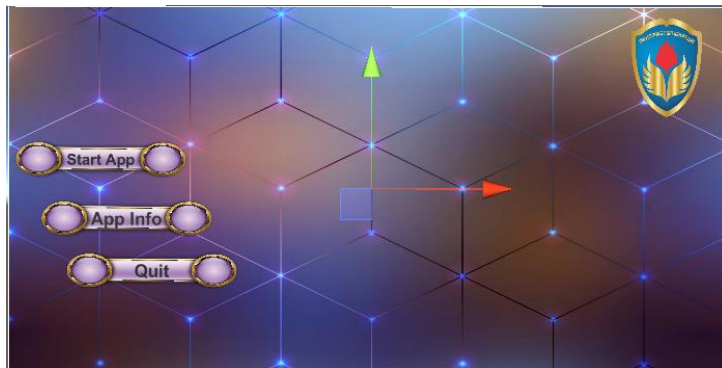
Drag material Transparent ke Wall

28. Klik tombol Play untuk mencoba

4

MEMBUAT APLIKASI MAIN MENU MENGUNAKAN UNITY

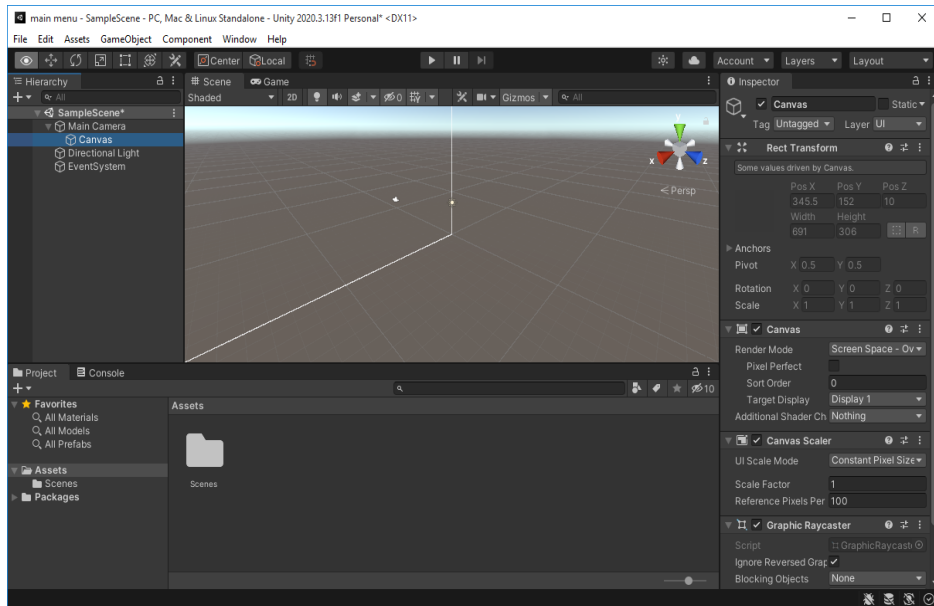
Pada sebuah aplikasi Augmented Reality, akan lebih menarik jika kita menambahkan halaman navigasi.



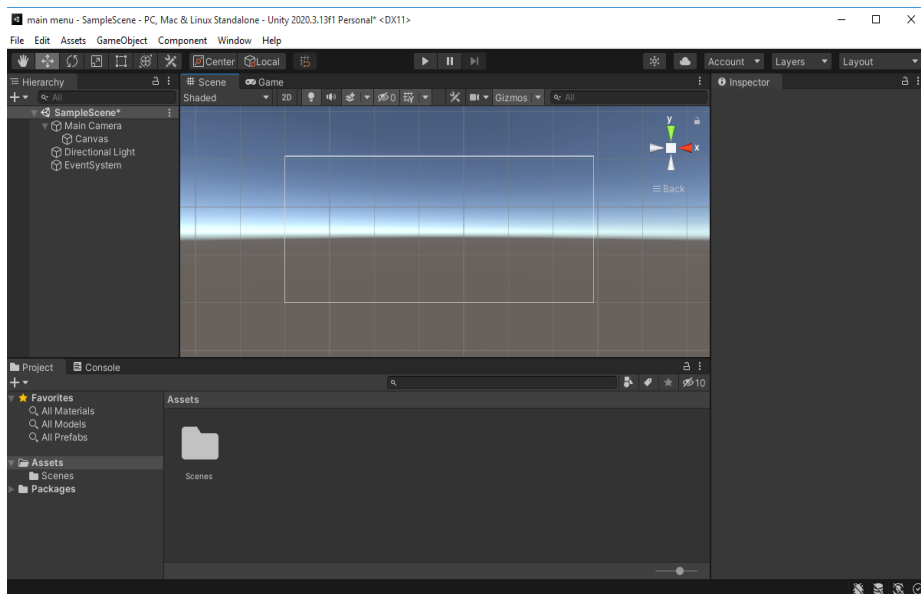
Pada bab ini kita akan membuat aplikasi yang berisi navigasi ke panel atau scene tertentu. Aplikasi ini kita beri nama Aplikasi **Main Menu**

Langkah-langkah :

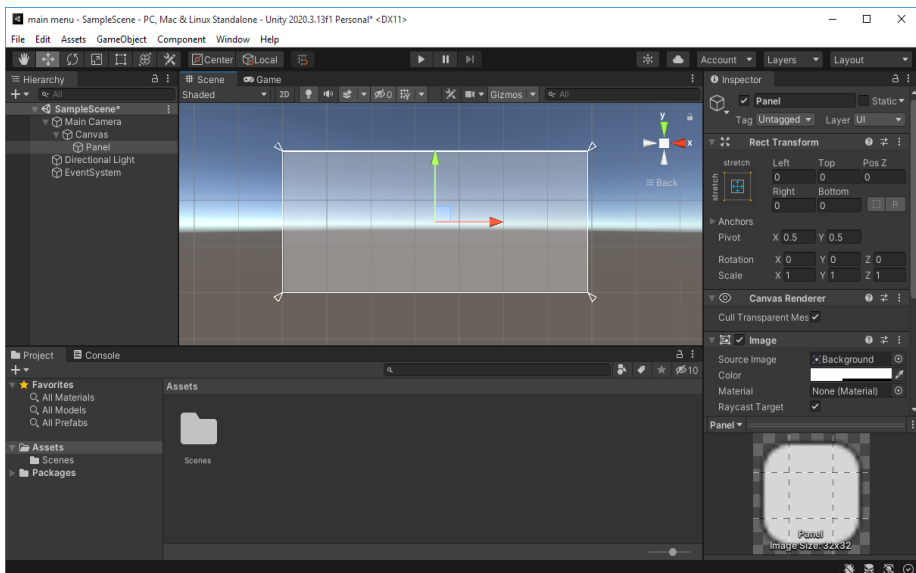
1. Buka Aplikasi Unity
2. Buat Canvas dengan cara klik kanan pada Main Camera di kolom hierarchy
– UI – Canvas



3. Canvas yang muncul akan sangat besar. Atur tampilan Canvas pada Scene dengan cara Scroll Mouse dan atur tampilan koordinat



4. Tambahkan Panel pada Canvas dengan cara klik kanan Canvas di kolom Hierarchy – UI – Panel

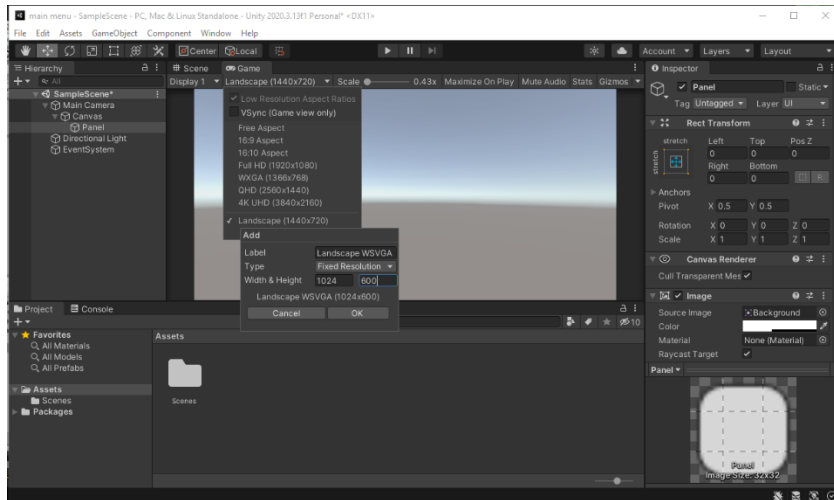


5. Klik Game untuk melihat tampilan yang akan muncul di PC atau Android. Pada setelan default, resolusi layar akan berada pada posisi free aspect. Jika Anda akan menjalankan Unity di PC, maka biarkan resolusinya dalam posisi Free Aspect. Namun jika Anda akan menjalankan aplikasi di handphone, Anda harus mengubah resolusinya sesuai dengan handphone Anda. Contoh :

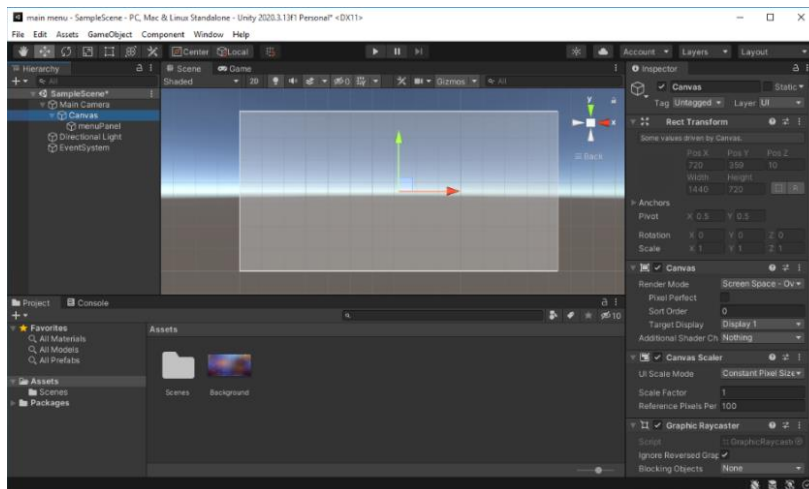
Ukuran Resolusi	Penyebutan	Contoh Perangkat
240X320	QVGA (Quarter VGA)	Samsung Galaxy Y S5360
320X480	HVGA (Half VGA)	Samsung Galaxy Ace S6802
480X854	FWVGA (Full Wide VGA)	Lenovo A516
640X960	qHD (Quarter HD)	Lenovo A850
720X1280	HD (High Definition)	Huawei Ascend G7
1920X1080	Full HD	OnePlus 3
1920X2160	Full HD+	ASUS Zenfone Max Pro M11
2560X1440	2K / QHD (Quad HD)	Samsung Galaxy S7
3840X2160	4K / UHD (Ultra HD)	Sony Xperia XZ Premium

Jika resolusi yang ada dalam pilihan belum tertera, maka Anda dapat menambahkan sesuai dengan resolusi HP Anda dengan cara klik tanda tambah (+) ketikkan nama (bisa apa saja yang mudah dikenali) kemudian

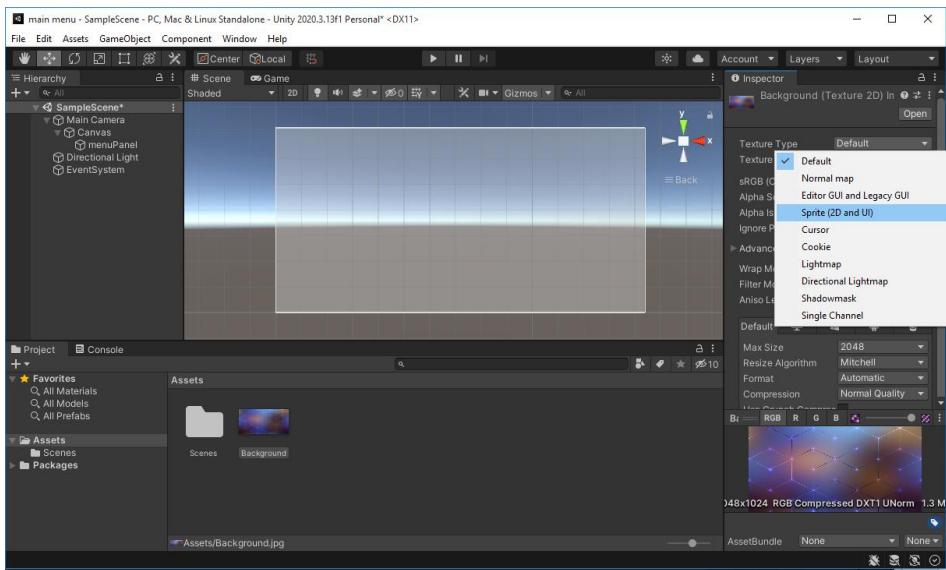
isikan width dan height (jika Anda ingin posisi aplikasi dalam kondisi Landscape maka isikan width lebih besar dari height)



6. Kembali ke Scene. Klik Panel pada kolom Hierarchy. Lihat pada Panel Inspector. Ganti nama panel menjadi menuPanel
7. Tahap selanjutnya kita akan menambahkan Background pada menuPanel. Klik kanan pada Assets kemudian klik import new assets – pilih background yang Anda inginkan

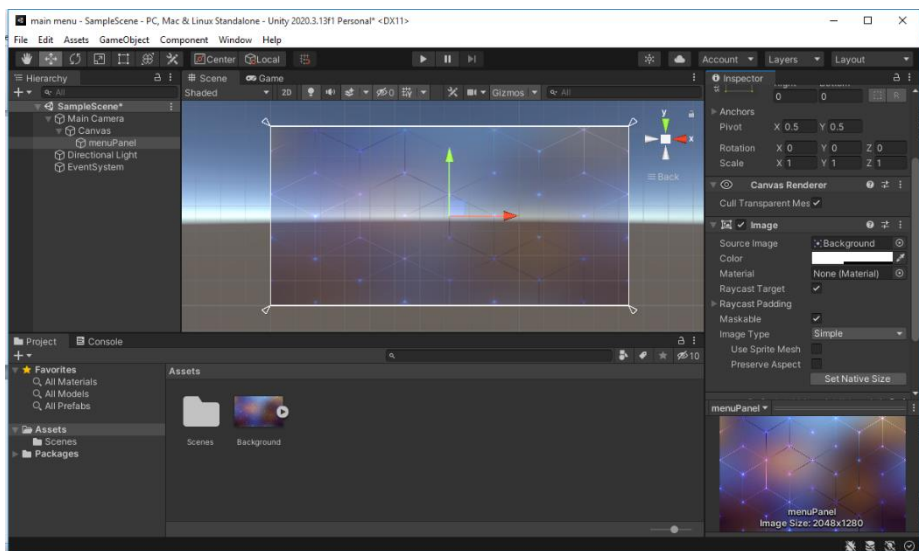


8. Selanjutnya klik pada Background yang baru Anda tambahkan. Kemudian pada Inspector ganti Texture Type menjadi Sprite (2D dan UI)

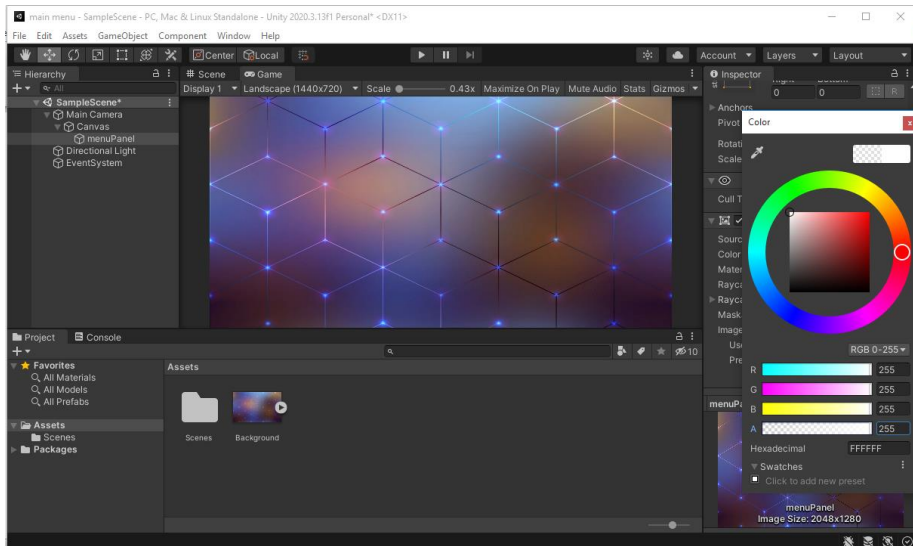


Scroll ke bawah pada Inspector dan klik Apply

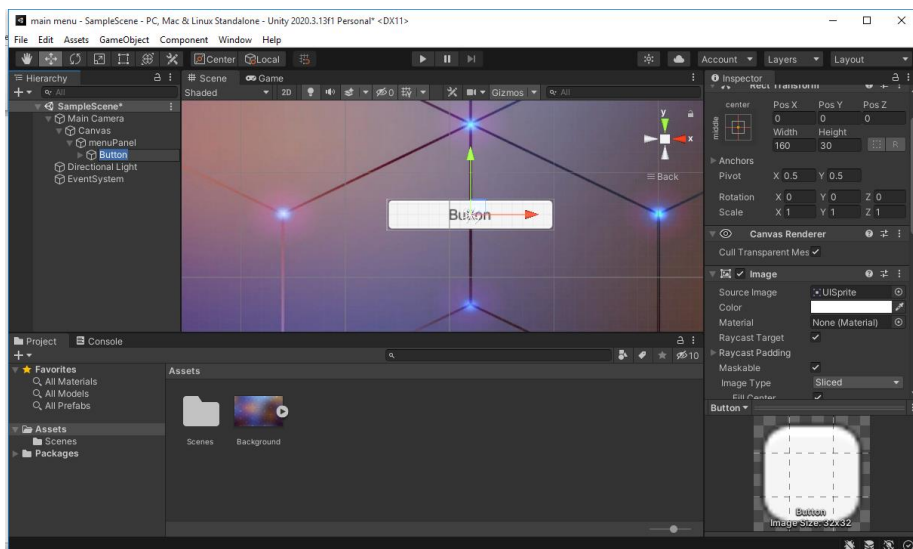
9. Berikutnya, klik menuPanel pada Inspector. Perhatikan bahwa pada menuPanel Inspector ada bagian source image. Drag gambar background ke dalam source image atau klik pada tanda lingkaran di sebelah source image dan pilih gambar background-nya



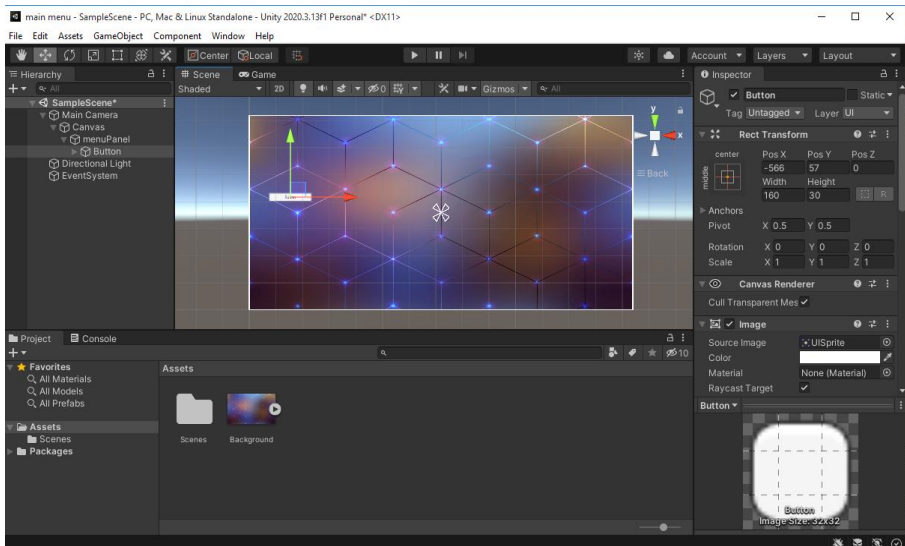
10. Jika Anda cek di Scene Game, maka terlihat background tersebut masih transparat. Untuk mengubah agar tidak transparat, klik pada Color di menuPanel Inspector, kemudian tarik bagian A ke kanan. Sesuaikan dengan keinginan Anda.



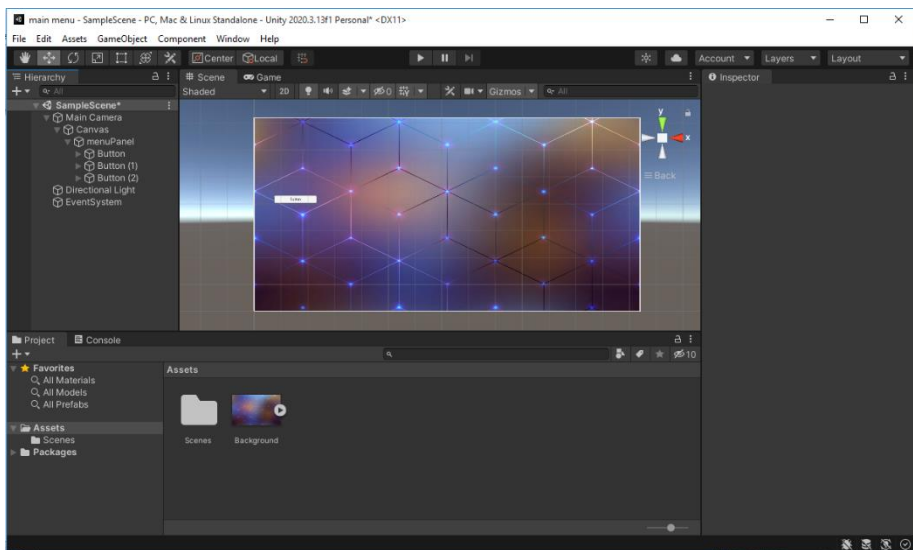
11. Berikutnya akan kita tambahkan tombol-tombol (button). Klik kanan pada menuPanel – UI – button



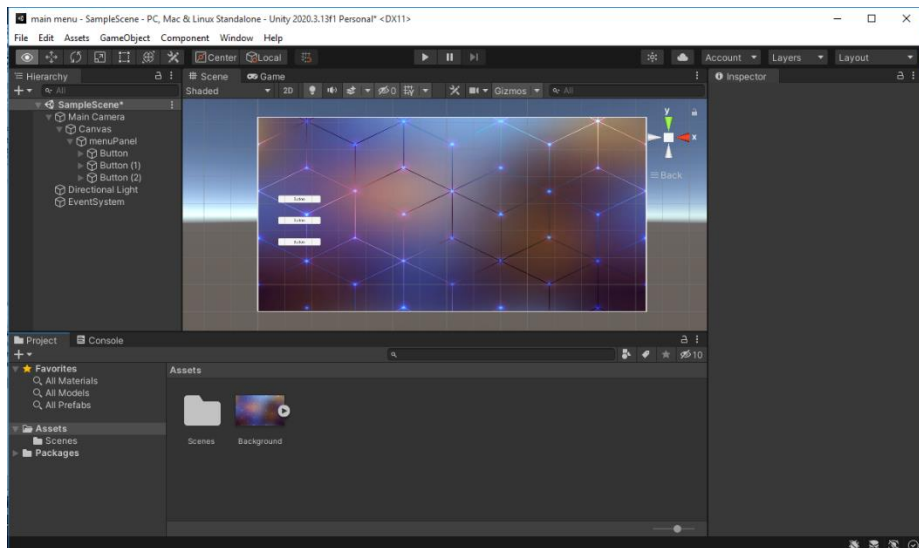
Atur posisi button dengan drag tanda panah.



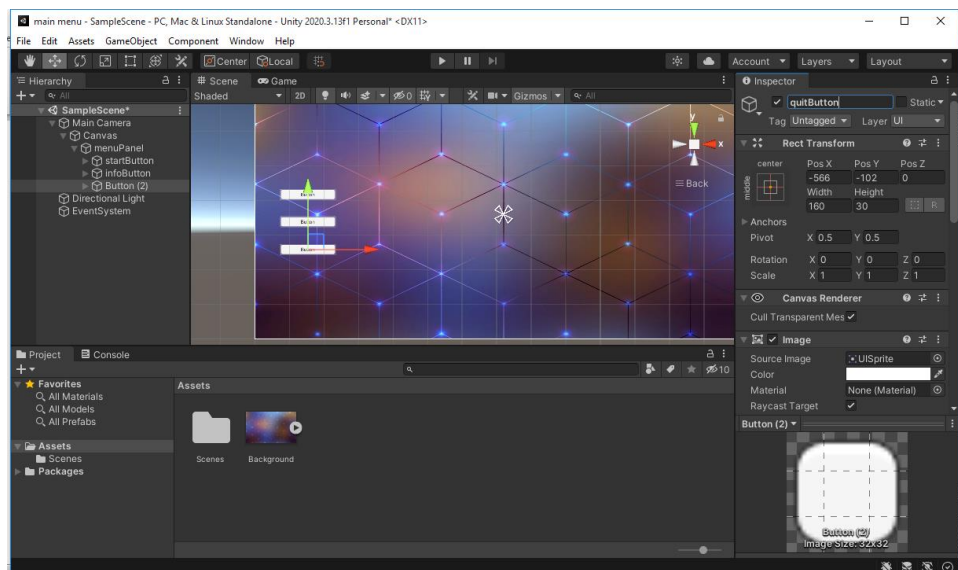
12. Kita akan membuat 3 buah button. Caranya bisa dengan mengulangi langkah 11 atau klik kanan button pada Hierarchy – Duplicate atau klik button pada hierarchy dan tekan Ctrl+D



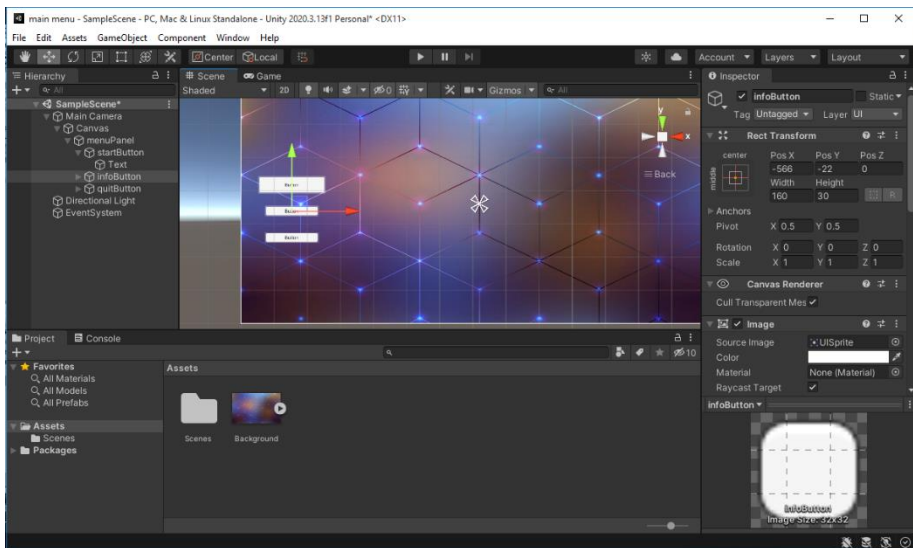
13. Atur posisi button sesuai keinginan



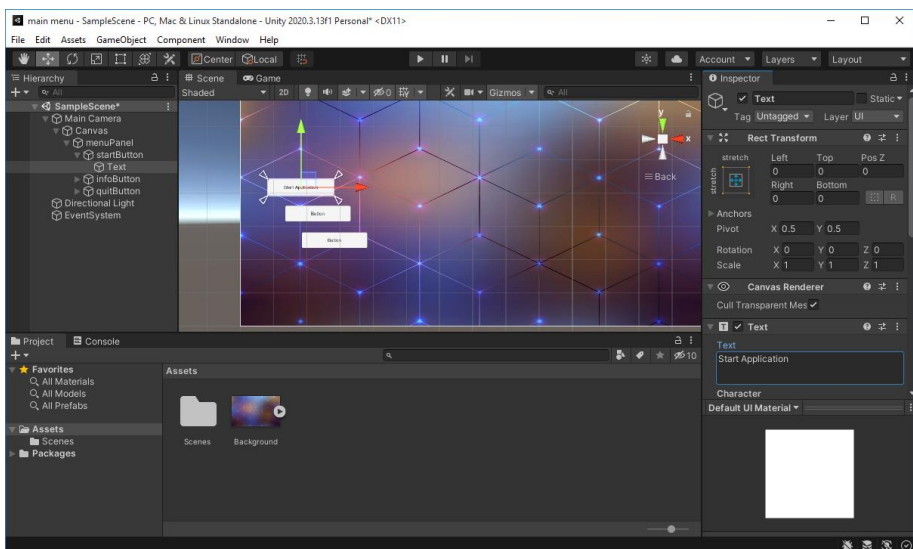
14. Klik pada masing-masing button, kemudian pada Inspector ganti namanya menjadi startButton, infoButton dan quitButton



15. Untuk mengubah ukuran button, lihat pada button Inspector, pada bagian Rect Transform Anda dapat mengubah Width dan Height sesuai keinginan

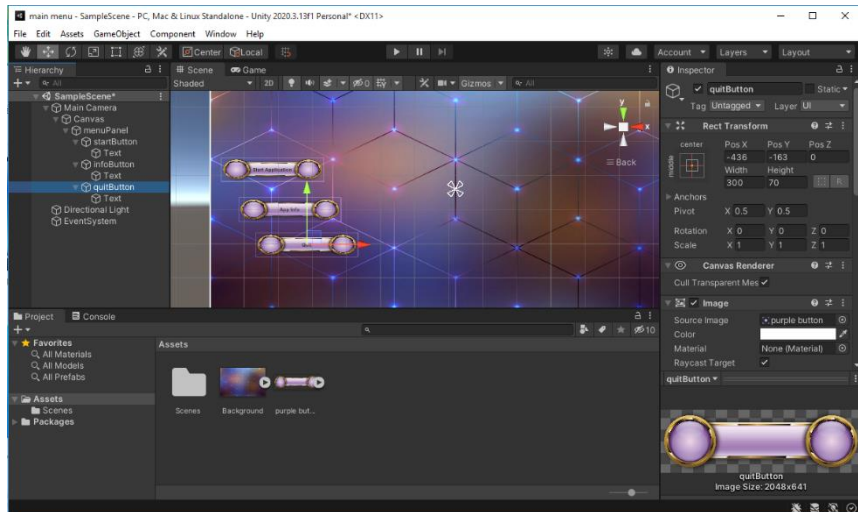


16. Untuk mengubah text pada button, klik tanda panah pada button – klik text – ubah text pada inspector

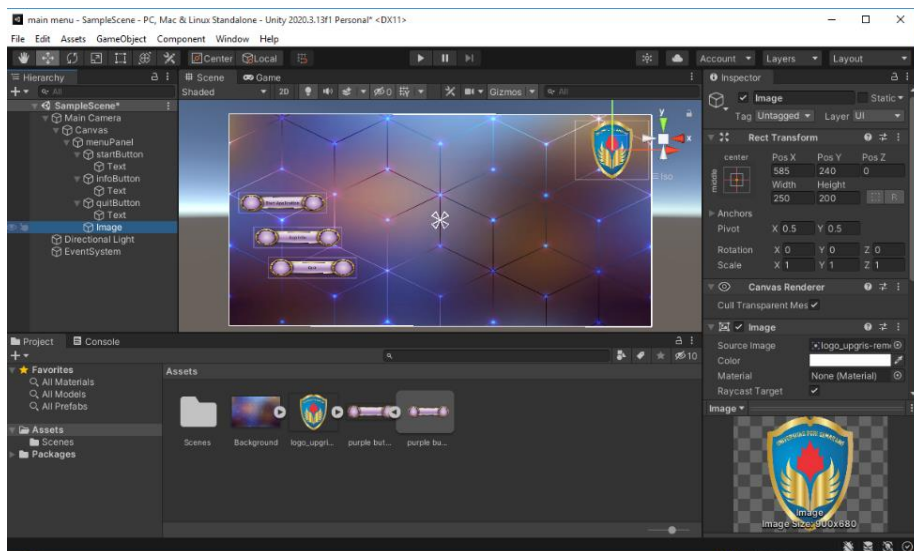


17. Untuk mengganti button background, klik kanan pada Assets – import new asset – pilih background untuk button yang Anda inginkan. Kemudian klik pada gambar yang baru Anda tambahkan. Pada Inspector ganti Texture Type menjadi Sprite (2D and UI). Kemudian klik button yang akan diubah backgroundnya, pada button Inspector klik pada tanda lingkaran di source

image kemudian pilih image yang diinginkan atau drag gambar background dari asset ke source image

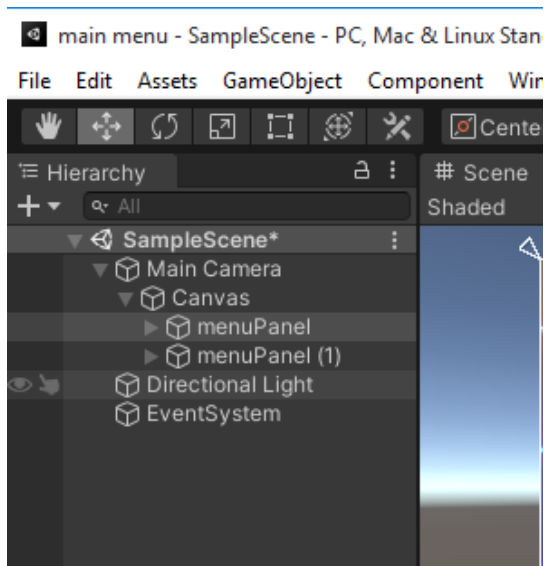


18. Jika Anda ingin mengubahkan gambar ke dalam menuPanel, klik kanan pada menuPanel – UI – image. Ini bisa Anda gunakan untuk menambahkan desain-desain yang lain (desain dilakukan di luar Unity)

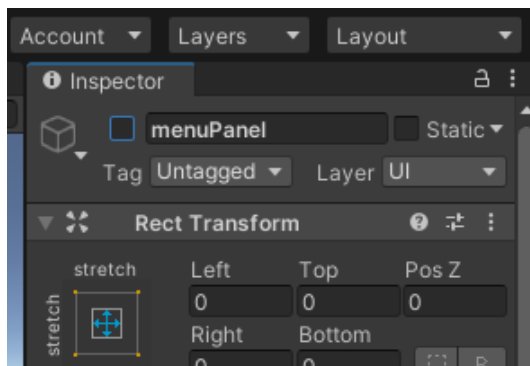


Kreasikan halaman awal ini sesuai dengan keinginan Anda

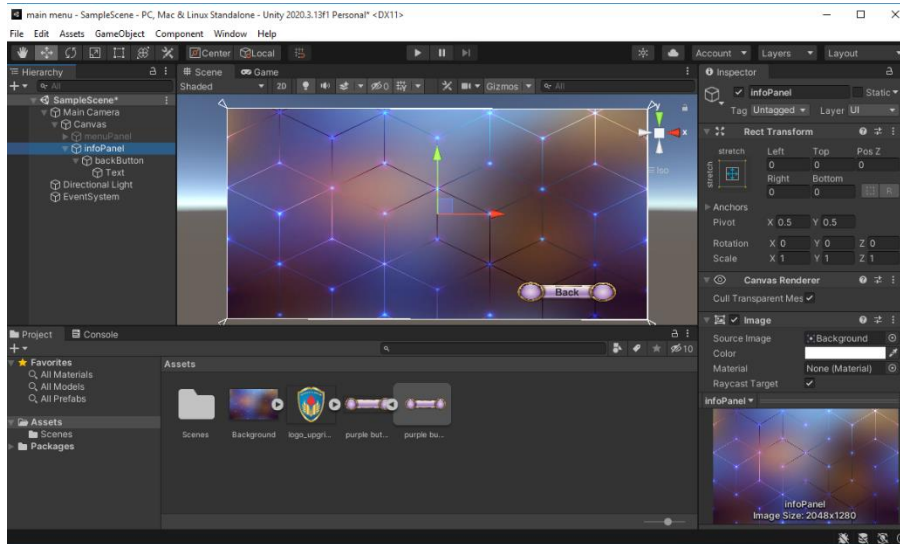
19. Langkah berikutnya, kita akan membuat halaman untuk Info Aplikasi. Duplikat menuPanel dengan klik menuPanel pada Hierarchy kemudian klik kanan – Duplicate atau tekan Ctrl+D



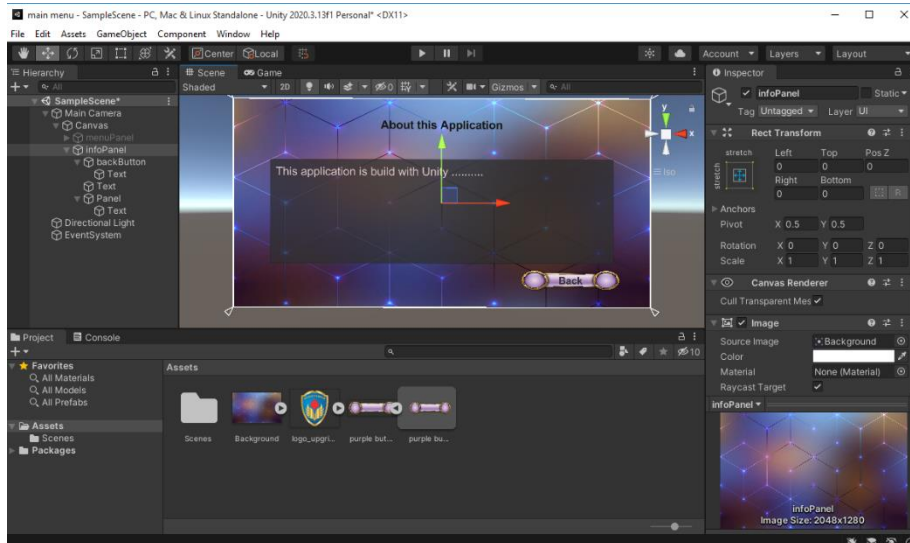
20. Berikutnya kita hide menuPanel dengan klik menuPanel, kemudian pada menuPanel Instructor hilangkan tanda centang pada menuPanel



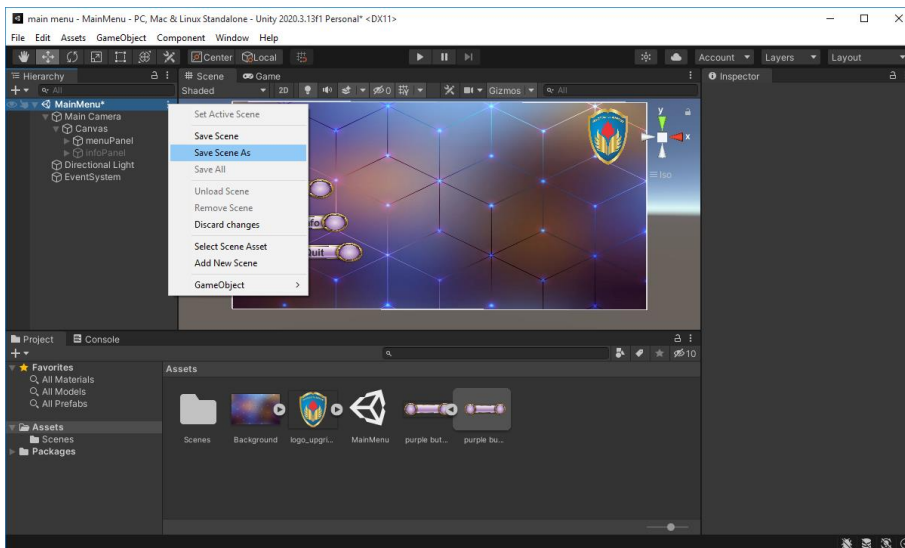
21. Berikutnya kita akan membuat tampilan halaman info aplikasi (halaman yang akan terbuka dengan menekan tombol infoApp). Klik menuPanel1. Ganti nama panel menjadi infoPanel. Hapus button yang ada. Sisakan satu button (ini akan kita ubah supaya menjadi tombol back). Langkah-langkah seperti di atas.



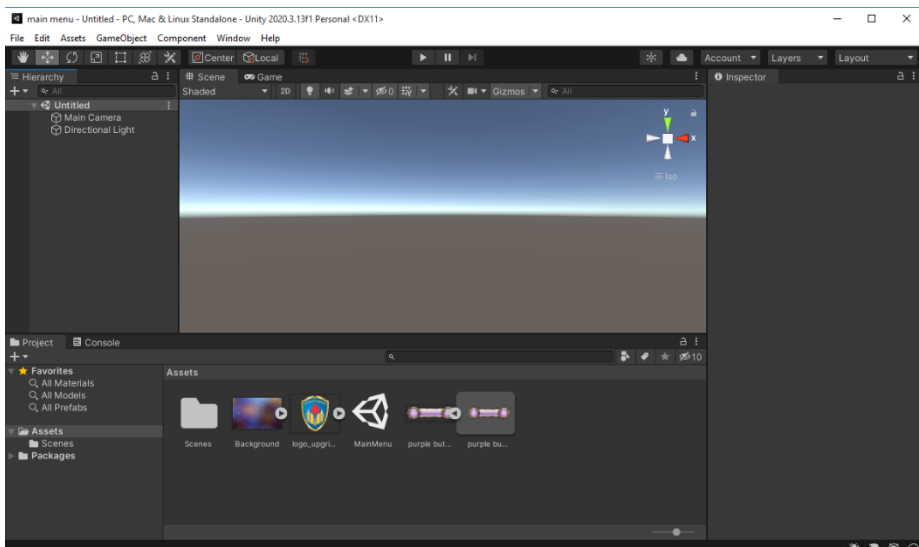
22. Kita bisa menambahkan Text atau panel baru di scene dengan klik kanan infoPanel – UI – Text atau infoPanel – UI – Panel. Kreasikan halaman ini!



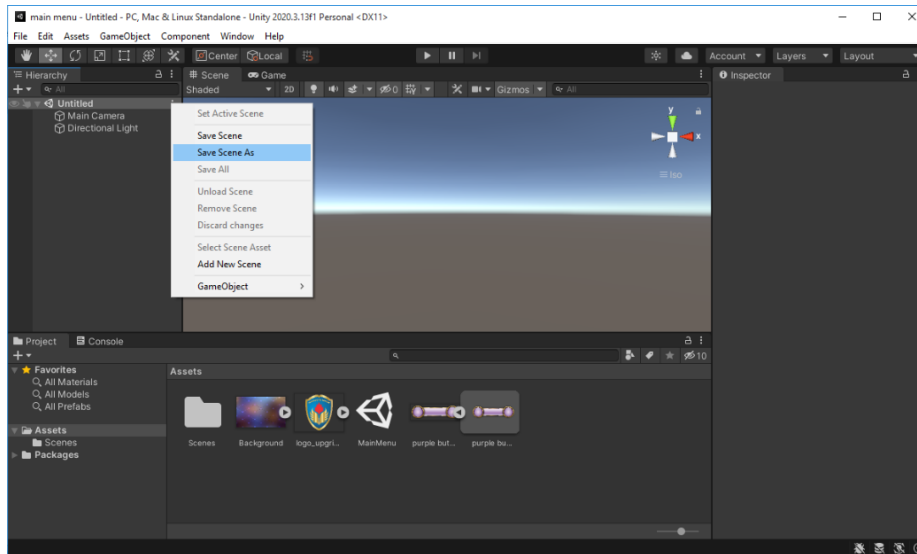
23. Berikutnya kita akan membuat Script untuk mainMenu. Langkah pertama, kita save dulu scene yang telah dibuat dengan nama mainMenu dengan cara klik tanda titik tiga di sebelah Sample Scene pada Hierarchy – Save Scene as – Tuliskan mainMenu



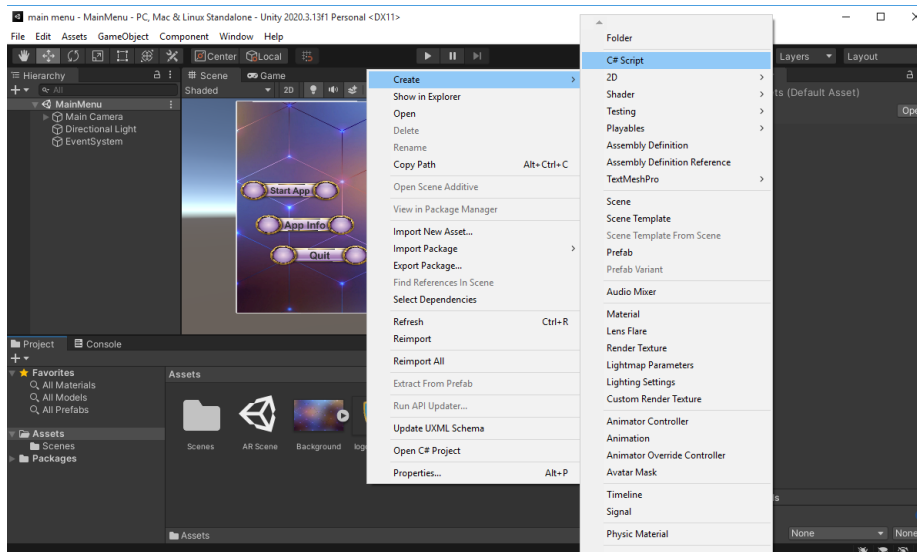
24. Tambahkan Scene baru untuk membuka aplikasi AR (jika diinginkan. Buat sesuai kebutuhan). Klik File – New Scene – basic build in – Create



Simpan Scene dengan nama AR scene dengan cara klik tanda titik tiga di sebelah Untitled – Save Scene as – tuliskan nama scene (AR Scene)



25. Kembali ke MainMenu Scene. Klik File - Open Scene- Main Menu. Berikutnya kita akan menuliskan script yang dibutuhkan. Klik kanan pada Assets – Create – C# Script



Ganti nama C# Script menjadi menuScript. Double klik menuScript untuk membuka Script Editor. Pada modul ini menggunakan Visual Studio Code. Anda bisa menggunakan Script Editor yang lain seperti Microsoft Visual Studio

```

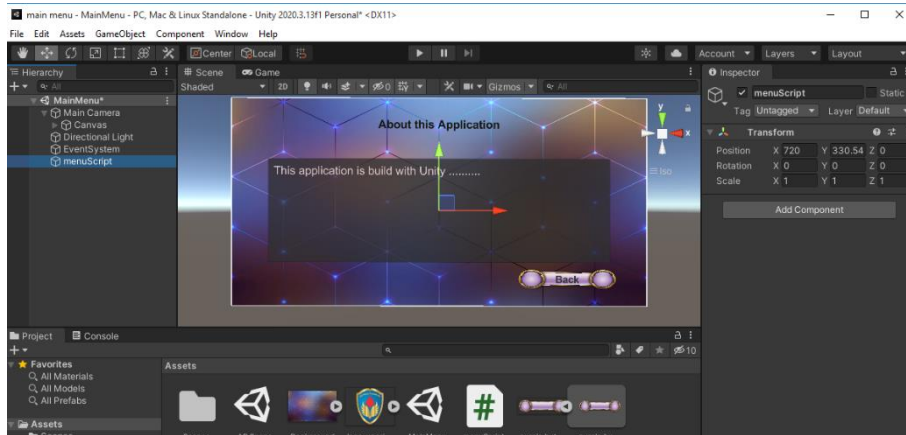
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.SceneManagement;

public class menuScript : MonoBehaviour
{
    public GameObject menuPanel;
    public GameObject infoPanel;
    // Start is called before the first frame update
    void Start()
    {
        menuPanel.SetActive(true);
        infoPanel.SetActive(false);
    }

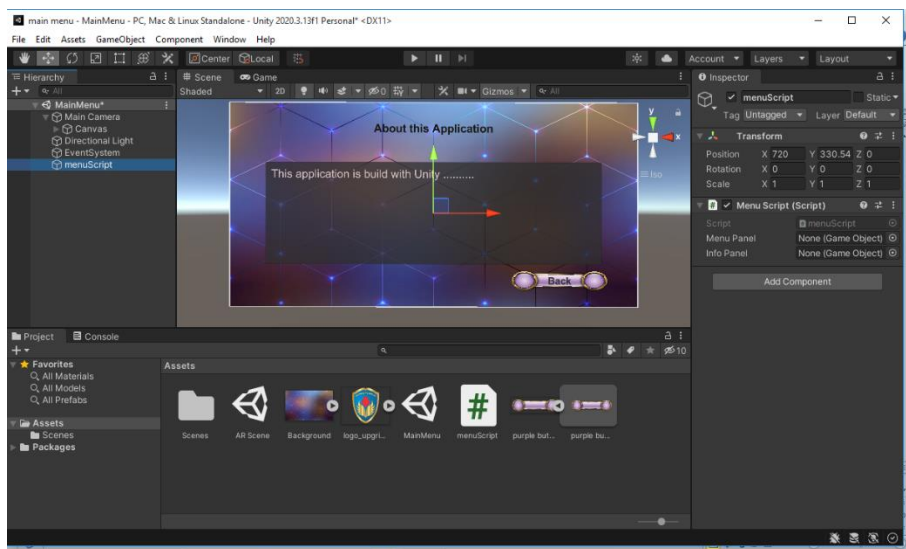
    // Update is called once per frame
    void Update()
    {
    }
    public void startButtonClicked()
    {
        Application.LoadLevel("AR Scene");
    }
    public void quitButtonClicked()
    {
        Application.Quit();
    }
    public void infoButtonClicked()
    {
        infoPanel.SetActive(true);
        menuPanel.SetActive(false);
    }
    public void backButtonClicked()
    {
        infoPanel.SetActive(false);
        menuPanel.SetActive(true);
    }
}

```

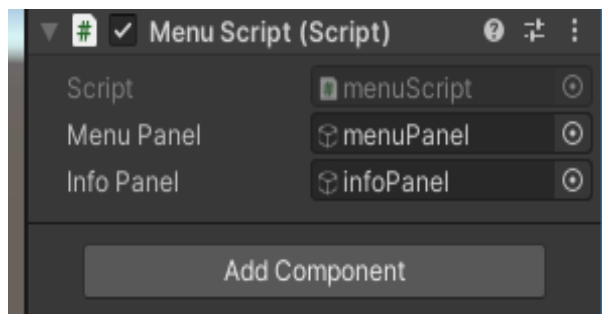
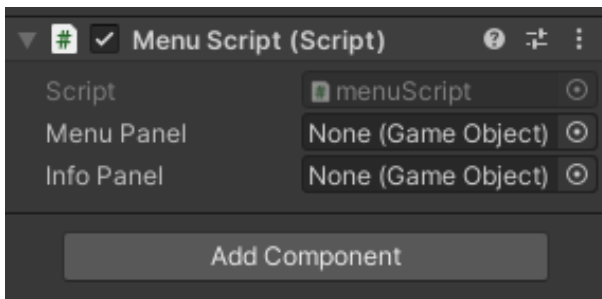
26. Klik Save, kemudian tutup Script Editor. Kembali ke Unity. Kita akan membuat tempat untuk memasukkan script. Klik kanan pada Hierarchy – Create Empty. Akan muncul Game Object. Ganti namanya pada tab Inspector menjadi menuScript



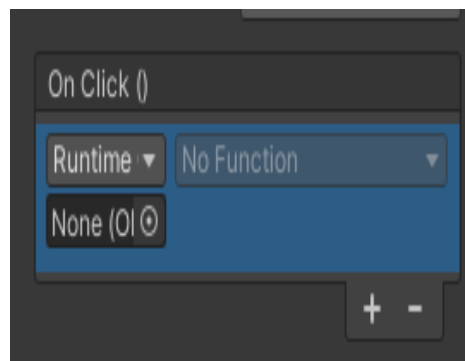
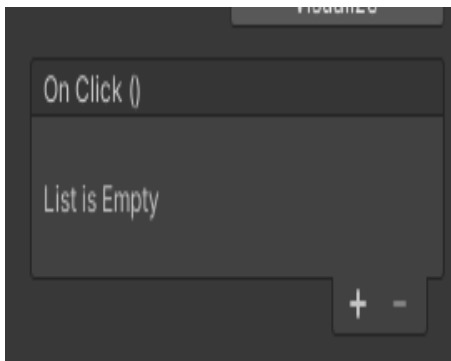
27. Berikutnya drag menuScript pada Assets ke dalam menuScript pada Hierarchy. Maka tampilan Inspectornya akan berubah



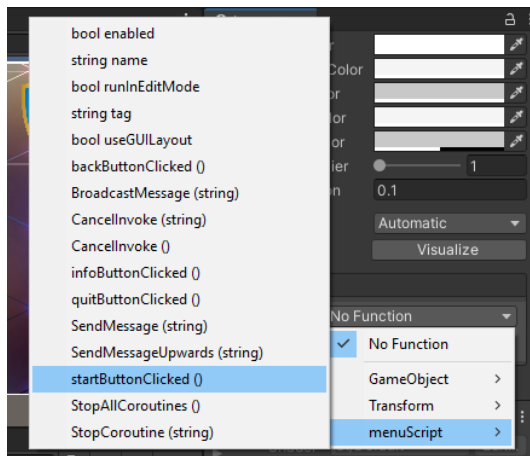
28. Perhatikan menu Script pada menuScript Inspector. Klik tanda lingkaran pada menu Panel kemudian klik menuPanel. Demikian juga untuk tab info Panel.



29. Selanjutnya kita akan membuat button-button yang ada pada menuPanel berfungsi. Caranya adalah dengan klik menuPanel (Aktifkan hanya menuPanel saja supaya terlihat dengan jelas button yang akan diberi perlakuan). Klik startButton, lalu pada inspector lihat pada bagian On Click

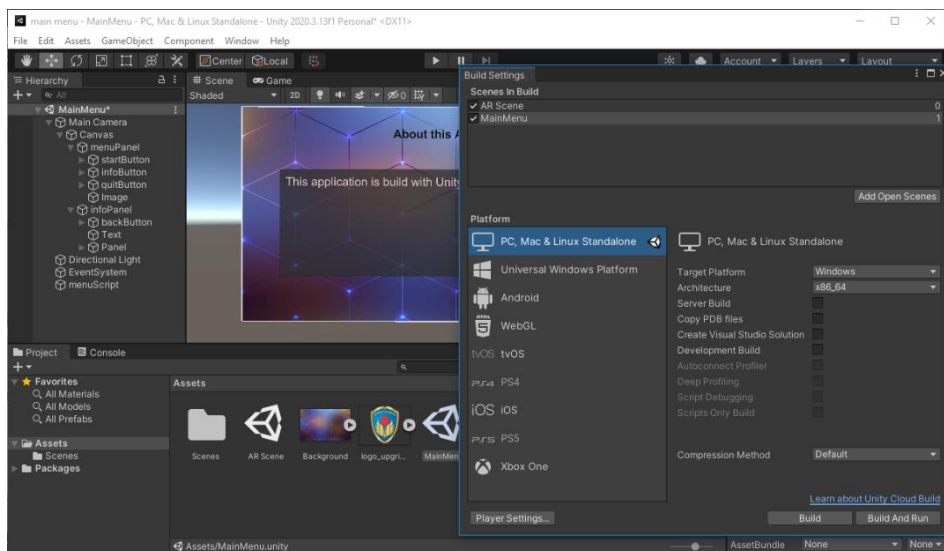


Klik tanda +. Drag menuScript dari Hierarchy ke bagian yang bertuliskan None. Kemudian klik tanda panah pada bagian yang bertuliskan No Function, pilih menuScript – startButtonClicked()



Lakukan hal yang sama untuk tombol-tombol yang lain

30. Klik Tombol Play untuk mencoba.
31. Untuk membangun aplikasi .apk, klik file – build setting – drag menuScene dan AR Scene dari Assets ke build setting. Kemudian klik Build



32. Aplikasi .apk sudah siap digunakan.

MEMBUAT ANIMASI OBJEK SEDERHANA DENGAN MENGGUNAKAN ANIMATOR

Suatu aplikasi Augmented Reality akan menjadi lebih menarik jika memiliki animasi di dalamnya. Ada beberapa cara untuk menambahkan animasi pada sebuah Objek 3D.

1. Objek 3D yang di download dari Assets Store atau dari penyedia yang lain telah memiliki animasi, sehingga langsung bisa diterapkan pada object 3D
2. Menggunakan Script untuk membuat animasi
3. Menggunakan Animator untuk memberikan animasi berupa translasi, rotasi atau scale.

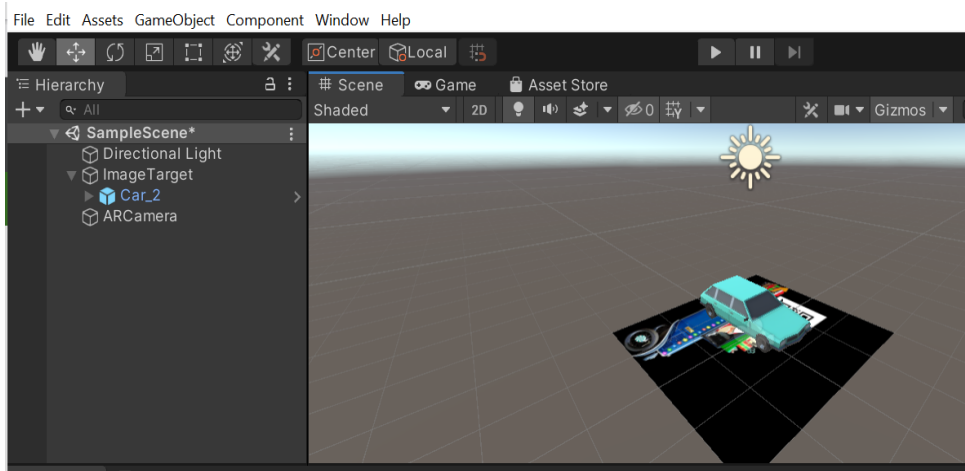
Pada bab ini kita menggunakan Animator untuk membuat animasi pada Objek 3D. Ada 2 project yang akan kita buat. Project pertama kita belajar bagaimana menggunakan Animator Translasi untuk membuat sebuah object berpindah tempat. Sedangkan pada project kedua menggunakan animator rotasi untuk membuat objek berputar.

Project 1

1. Dapatkan lisence Key dan daftarkan target marker di Vuforia.

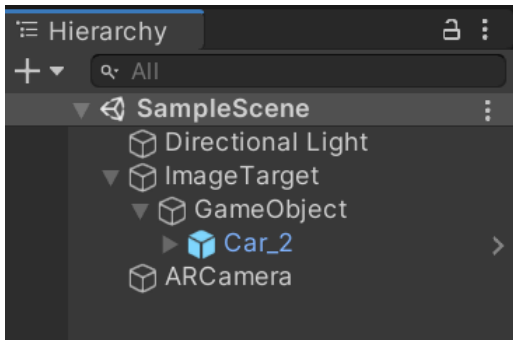


2. Buka project baru di unity. Hapus bagian Main Camera dari Panel Hierarchy.
3. Install Vuforia Engine AR pada project kemudian tambahkan AR Camera pada Panel Hierarchy.
4. Import image target yang telah di download dari Vuforia dengan cara double klik file yang berisi image target kemudian import.
5. Dapatkan 3D object yang akan digunakan dari Assets Store kemudian import ke Unity.

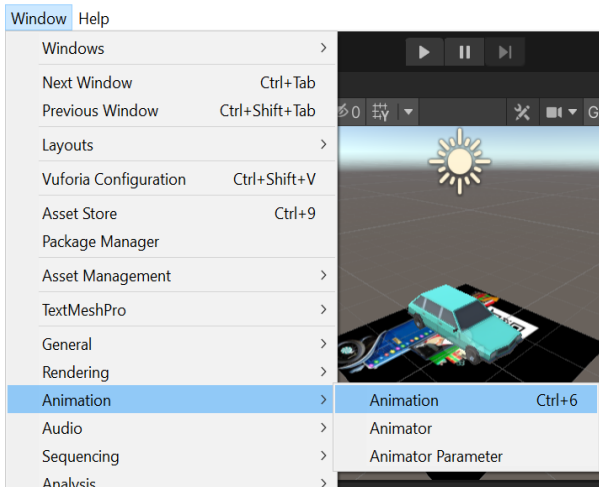


Berikutnya kita akan memberikan animasi sederhana pada objek mobil tersebut.

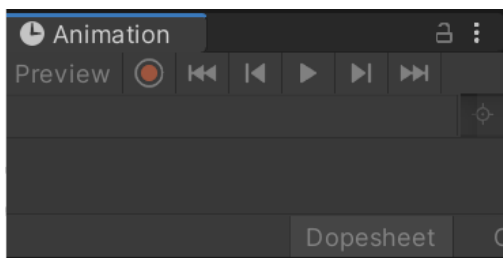
6. Pada Panel hierarchy buat GameObject baru dengan cara klik kanan – Create – Empty. Kemudia drag 3D object ke dalam Game Object. Berikutnya drag GameObject ke dalam Image Target



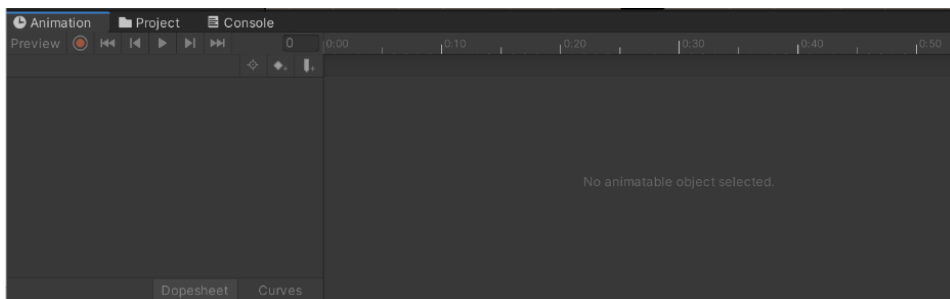
7. Klik Windows – Animation – Animation



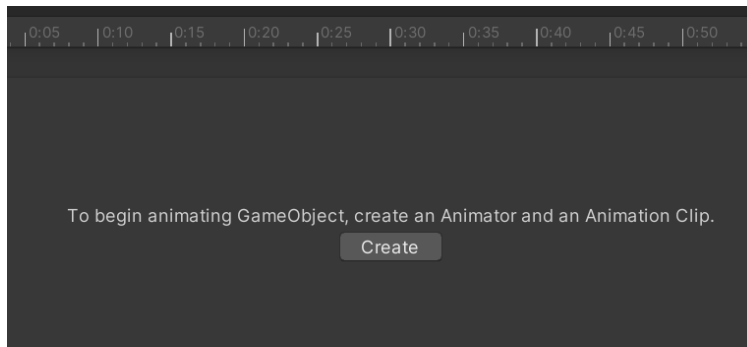
8. Akan muncul panel animation



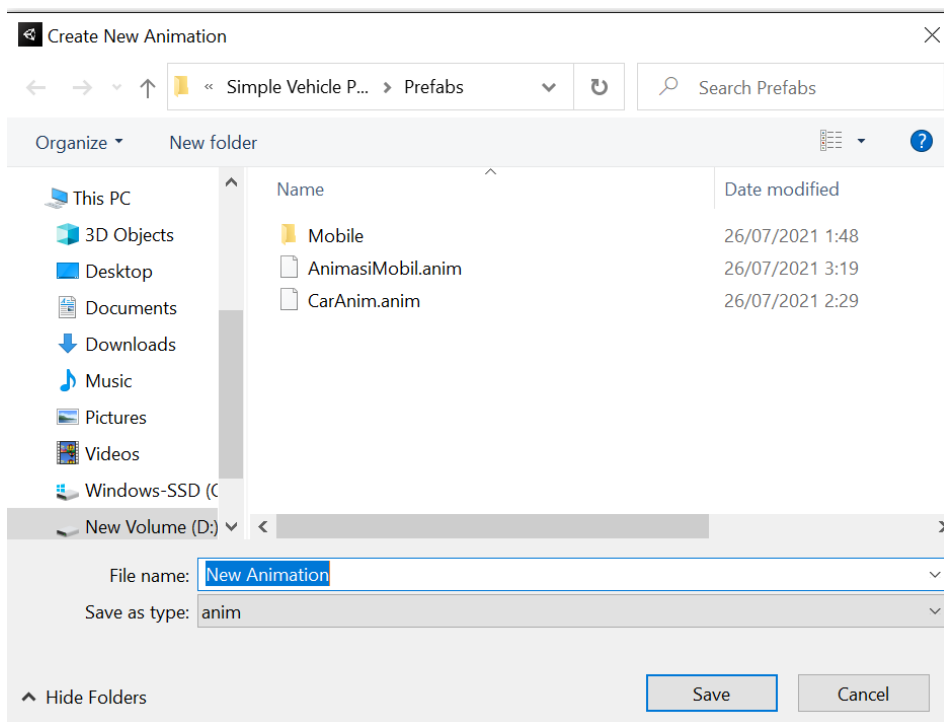
9. Drag ke bagian bawah, di sebelah panel project



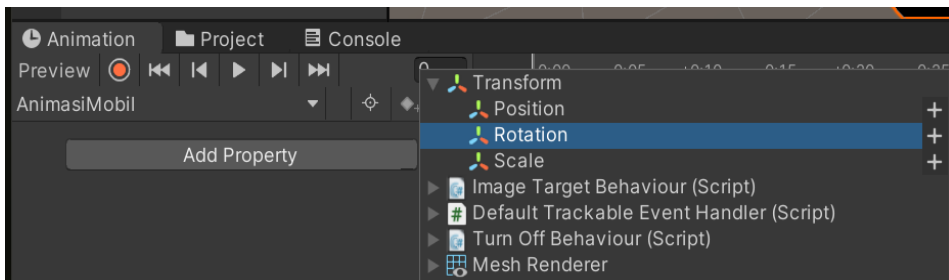
10. Klik pada GameObject yang akan diberi animasi, kemudian klik create,



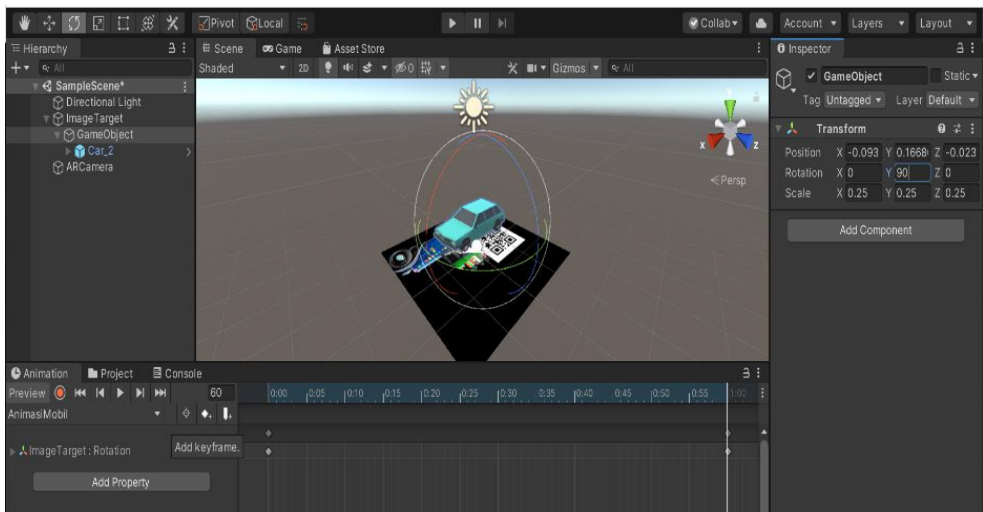
beri nama sesuai keinginan Anda, kemudian klik Save.



11. Berikutnya pada panel animation, klik Add property – Pada bagian Transform pilih rotation



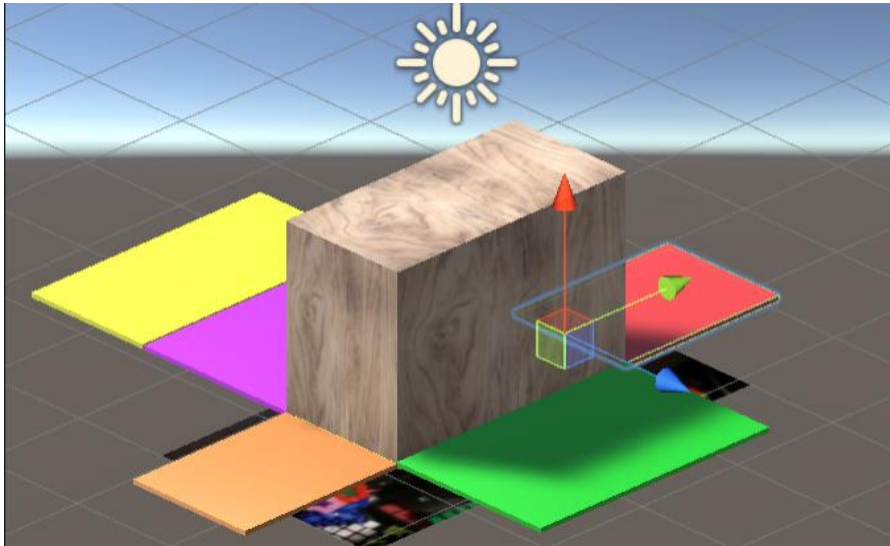
12. Untuk memberi animasi rotasi pada game object langkah yang harus dilakukan adalah : Klik pada tab waktu – klik add property – pilih gameobject yang akan diberi animasi (rotasi/transform/scale) – add frame



13. Klik pada tombol play pada preview animation untuk melihat hasilnya. Anda dapat menambahkan animasi lain sesuai dengan keinginan. Selamat mencoba.

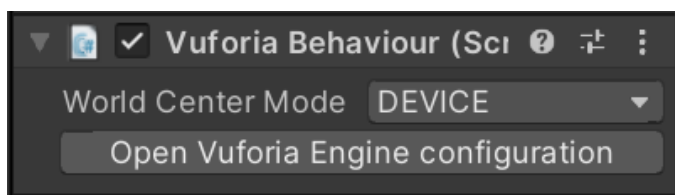
Project 2

Pada bagian ini kita akan membuat animasi pada sebuah kubus sehingga jaring-jaring kubus dapat bergerak terbuka. Aplikasi ini dapat digunakan sebagai media pembelajaran Matematika.

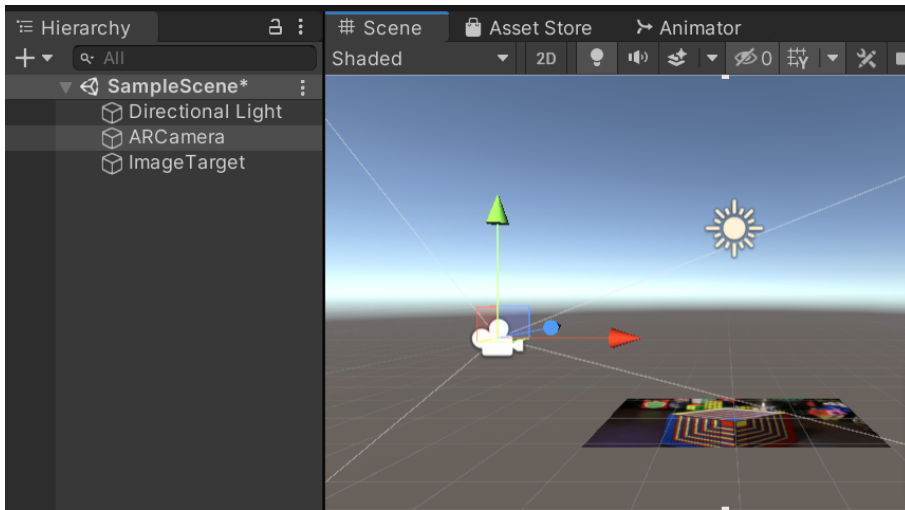


Langkah – Langkah yang harus dilakukan adalah:

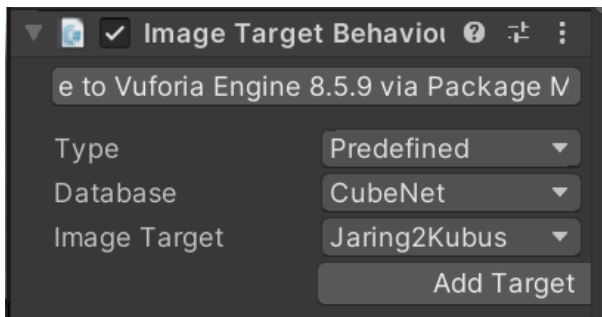
1. Buka Vuforia Developer untuk mendapatkan Liscence Key dan mengunggah Image Target
2. Buka Project Baru di Unity. Install Vuforia Engine AR.
3. Tambahkan AR Camera dengan cara klik kanan pada Panel Hierarchy – Vuforia Engine – AR camera. Kemudian klik pada AR Camera, pada bagian Inspector klik pada Open Vuforia Engine Configuration. Masukkan Liscence Key pada bagian App liscence key



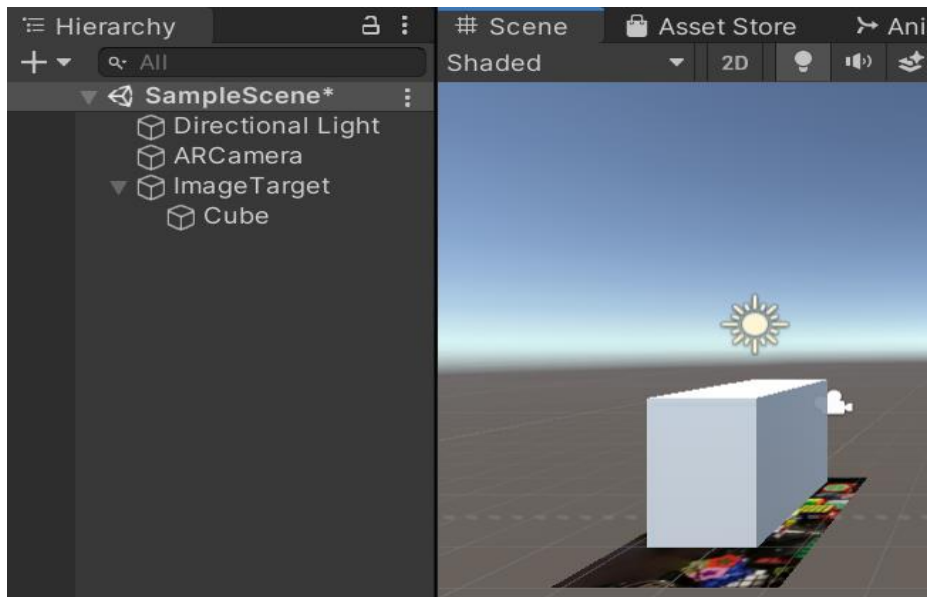
4. Import image target yang telah diunduh dari Vuforia, kemudian masukkan image target dengan cara klik kanan pada Panel Hierarchy – Vuforia Engine – Image.



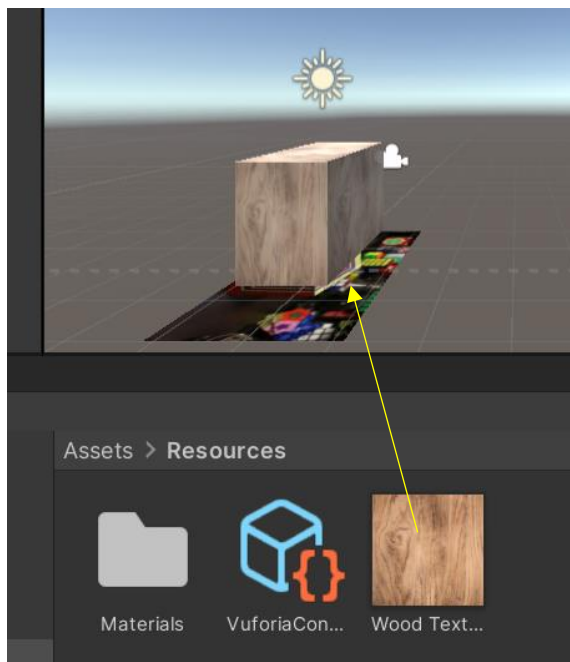
5. Sesuaikan Database dan image target yang Anda inginkan pada bagian image target behaviour. Jika Anda memiliki beberapa Database atau beberapa Image Target, Anda dapat memilihnya pada bagian ini.



6. Tambahkan Cube di atas image target. Agar Cube dapat muncul di lokasi yang sama dengan image target, klik kanan pada image target pada Panel Hierarchy – 3D object – cube. Atur ukuran cube sehingga membentuk balok



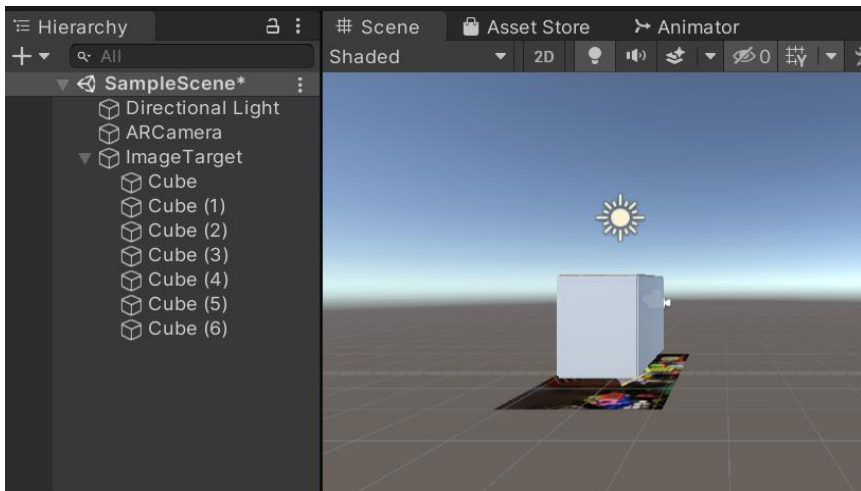
7. Tambahkan material texture kayu dengan cara drag material Wood Texture dari Windows explorer ke Assets. Kemudian drag dari Assets ke Cube



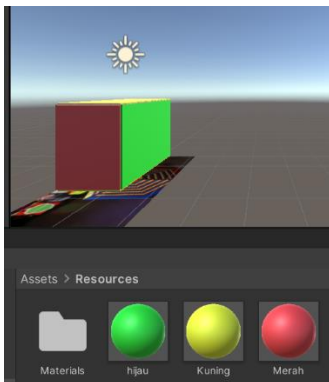
8. Atur tampilan pada Scene sehingga tampak permukaan balok seperti berikut



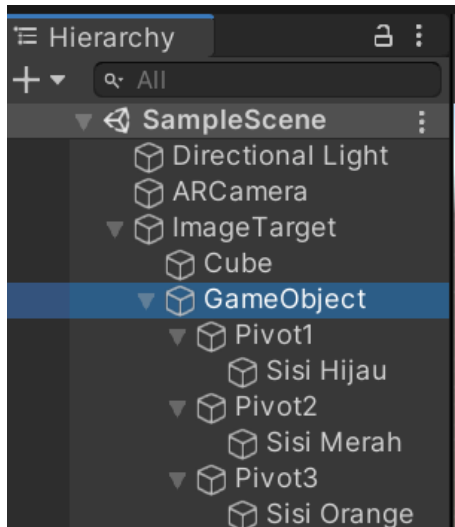
9. Tambahkan cube yang menutupi setiap permukaan balok. Buat ketebalan cube menjadi 0.01



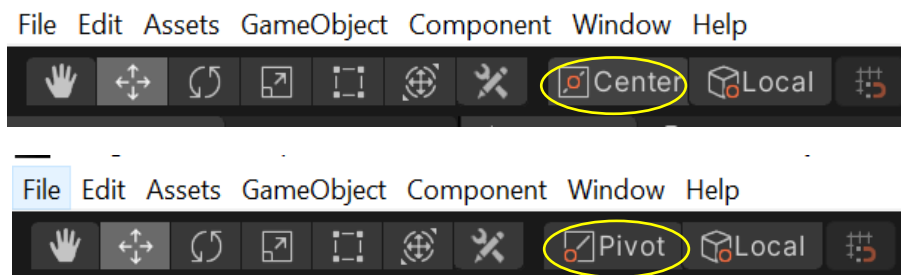
10. Beri warna berbeda untuk masing-masing permukaan untuk mempermudah



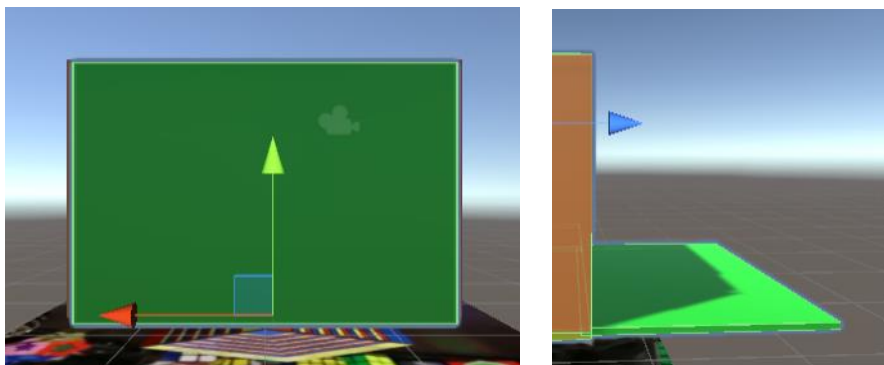
11. Berikutnya akan kita tambahkan pivot yang akan dijadikan poros untuk memutar/rotasi setiap permukaan. Klik kanan pada hierarchy – create – empty. Maka akan muncul game object baru. Game Object ini yang akan kita gunakan sebagai pivot. Ubah nama GameObject menjadi pivot 1.



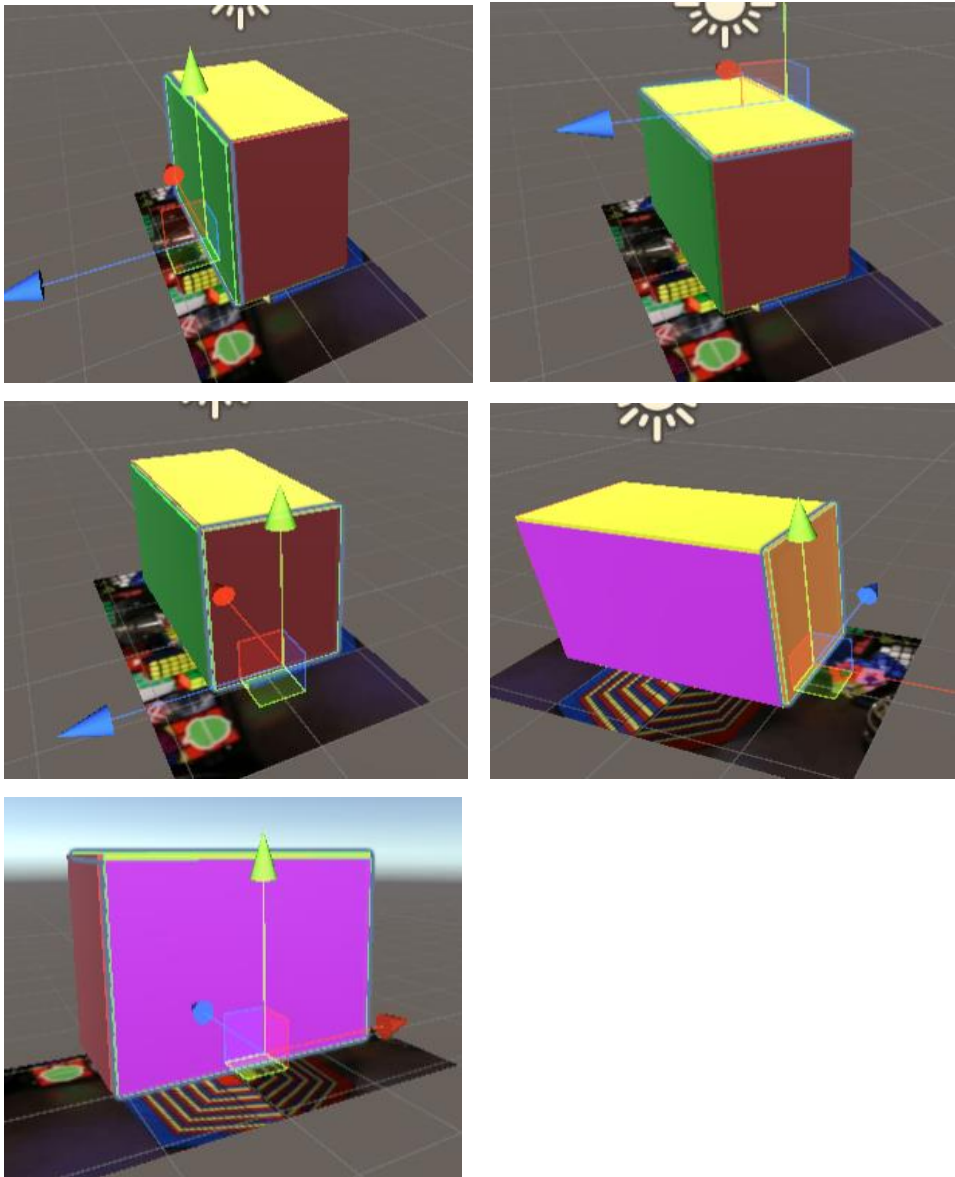
Klik pada toolbar yang bertuliskan Center hingga berubah menjadi pivot



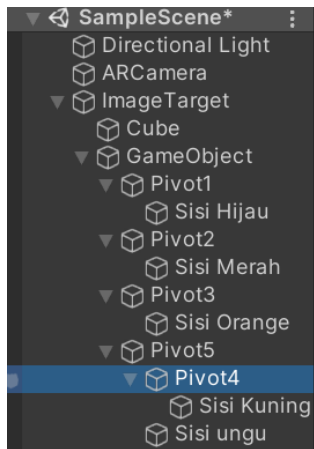
12. Letakkan pivot 1 pada poros sisi yang akan diputar



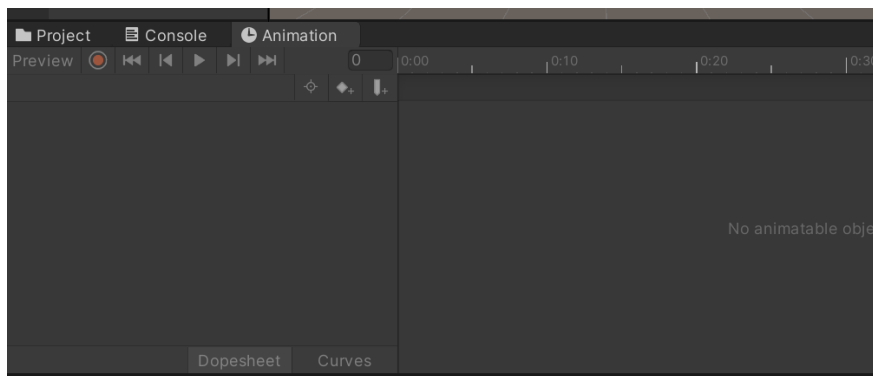
13. Lakukan hal yang sama untuk semua sisi.



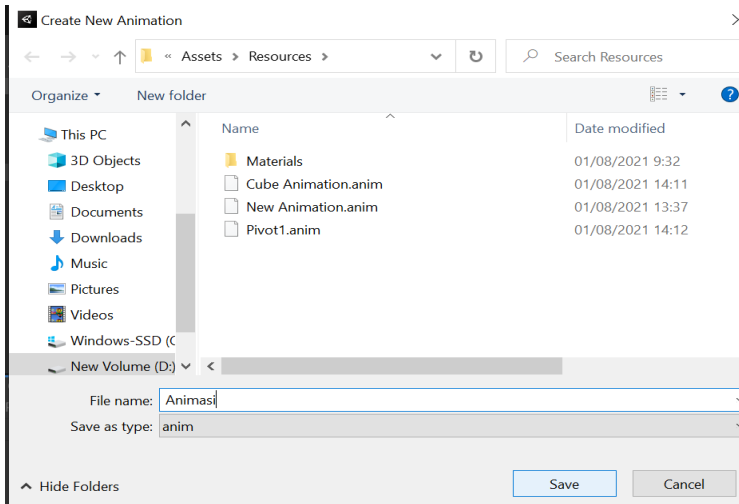
14. Sisi kuning dan pivotnya (pivot 4) menjadi child dari pivot sisi ungu karena sisi kuning akan terbuka dua kali. Rotasi pertama menggunakan pivot sisi kuning (pivot 4) dan rotasi kedua bersamaan dengan sisi ungu menggunakan pivot sisi ungu (pivot 5)



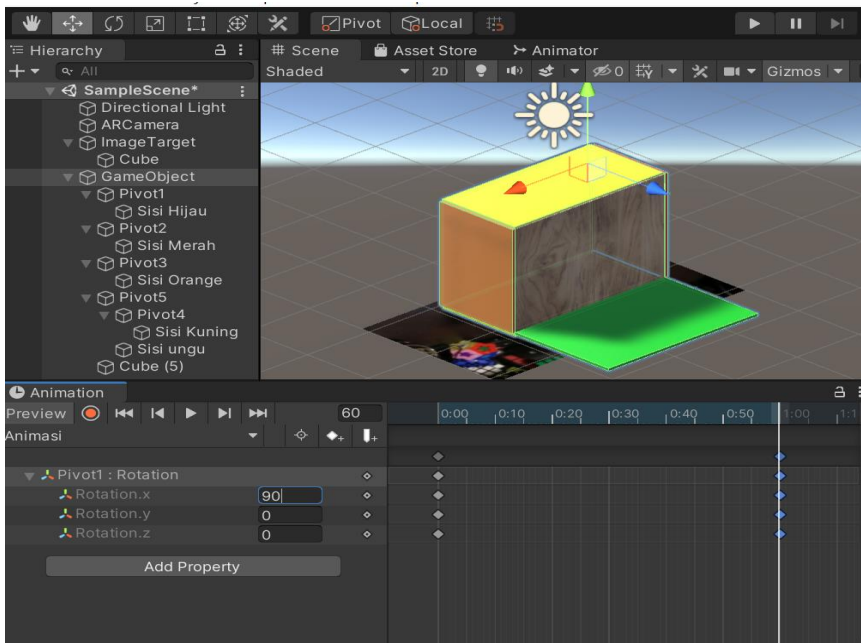
15. Berikutnya buka Scene Animation dengan cara klik Windows – Animation – Animation. Lalu Drag ke bawah Tab Animation ke sebelah tab Project



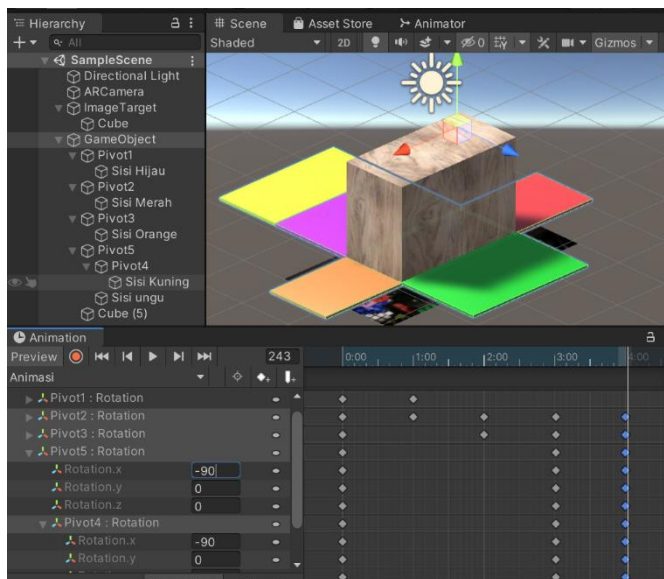
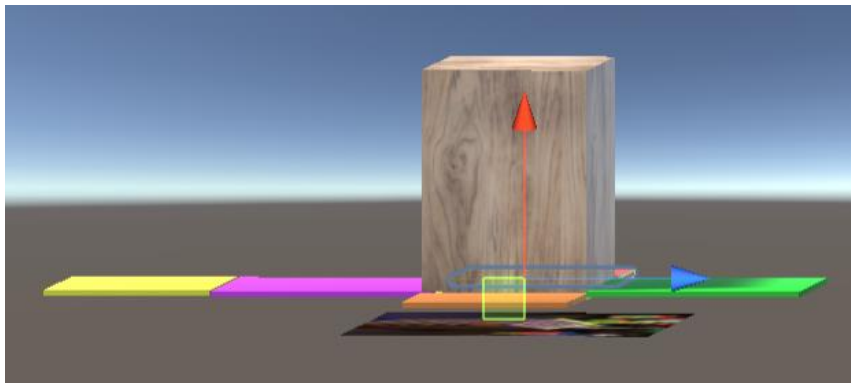
16. Klik pada GameObject kemudian klik Create pada Animation. Beri nama Animasi kemudian Save



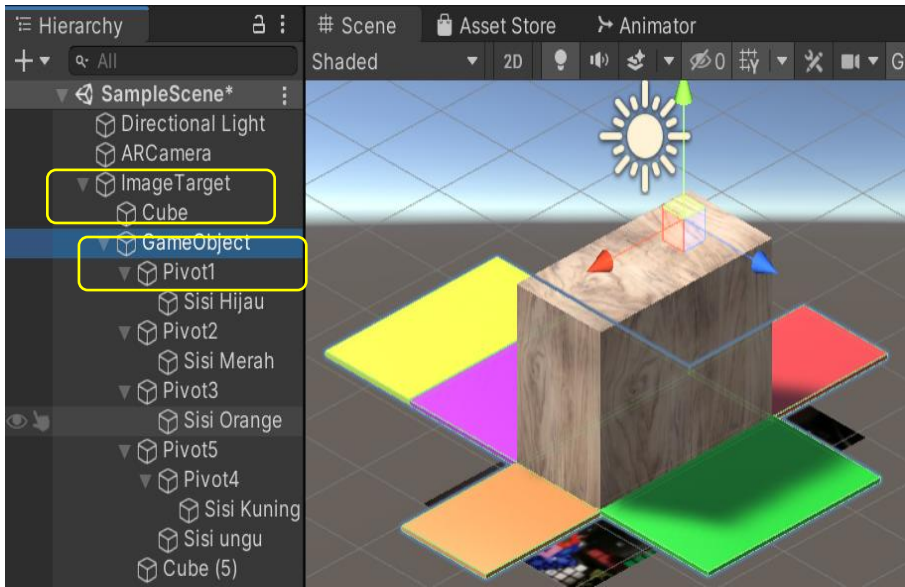
17. Klik pada menit 1.00 kemudian Add Property – Pivot 1 – Transform – Klik pada tanda + di sebelah Rotate. Atur Rotasi sisi hijau kemudian klik Add Keyframe.



18. Lakukan hal yang sama untuk semua sisi sehingga dapat terbuka seperti pada gambar berikut



19. Agar Kubus dan animasinya dapat dilakukan pada saat kamera mengenai Image Target, Drag GameObject menjadi Child dari Image Target.



20. Klik Tombol Play untuk menjalankan aplikasi

ANIMASI OBJEK 3D MENGGUNAKAN TOMBOL UI

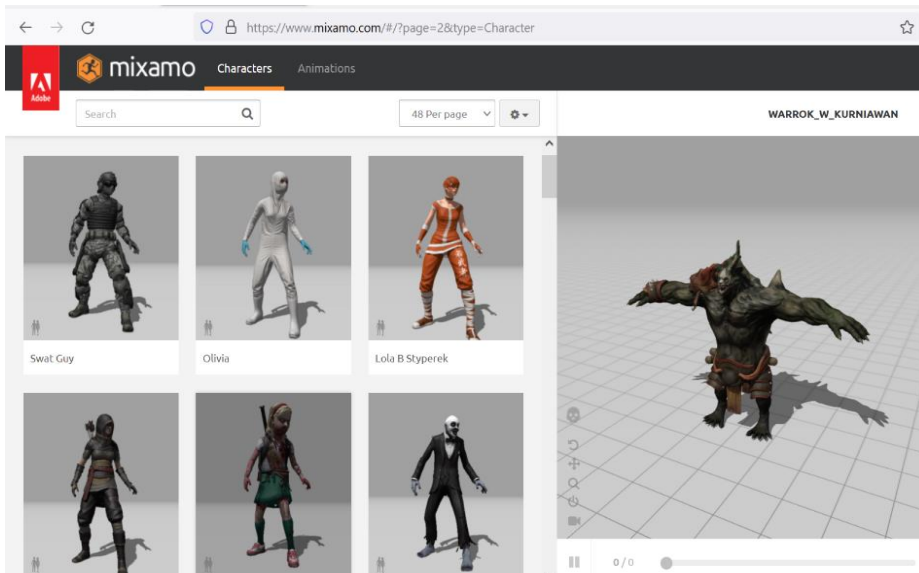


Pada bagian ini kita akan belajar bagaimana membuat aplikasi yang memuat objek 3D dan tombol untuk memilih animasinya. Untuk membuat aplikasi ini kita memerlukan

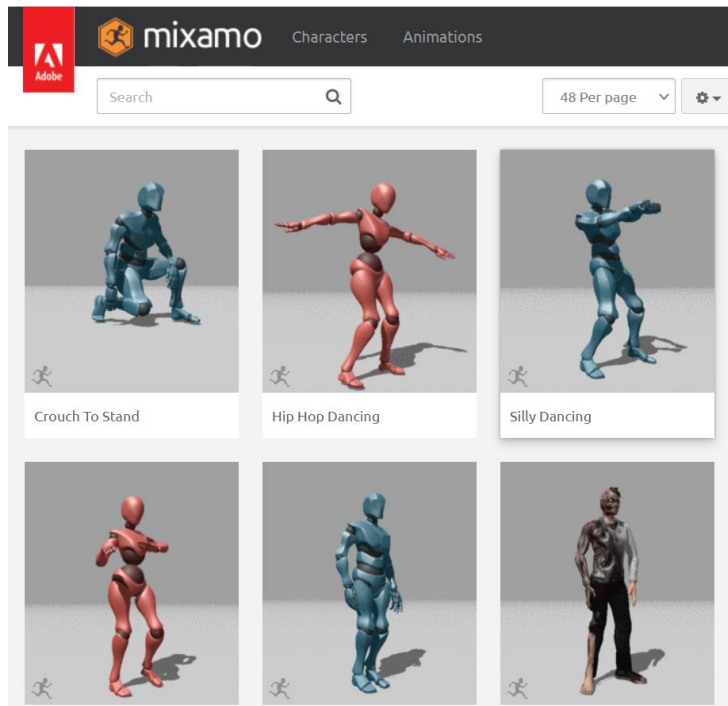
object yang memiliki animasi. Anda bisa membuatnya di aplikasi lain misalnya Blender atau 3Ds Max. Tapi tentu saja hal ini membutuhkan proses yang cukup panjang. Untuk keperluan pembuatan aplikasi Augmented Reality di bab ini kita tidak akan membangun object 3D sendiri, melainkan kita memanfaatkan sebuah website yang telah menyediakan karakter beserta animasinya yaitu Mixamo (<https://www.mixamo.com>).

Mixamo merupakan sebuah website penyedia rigging dan animasi secara otomatis yang bisa didapatkan secara gratis. Dengan menggunakan Mixamo anda dapat mengubah bagian biped mesh (posisi tulang) ataupun titik yang sudah ditentukan oleh Mixamo seperti bagian dagu, pergelangan tangan, lutut dan siku.

Anda dapat menggunakan akun Google untuk melakukan proses log in dan memilih karakter yang Anda butuhkan.



Mixamo telah menyediakan rigging dan animasi otomatis yang dapat diterapkan pada karakter yang Anda pilih.



Pilih karakter dan animasi yang diinginkan. Kemudian klik download.

DOWNLOAD

SEND TO AERO

UPLOAD CHARACTER

Silly Dancing ×

Energy Level 50

Overdrive 50

Character Arm-Space 50

Trim 0 100

Mirror

Pilih Format FBX for Unity. Pada saat download karakter dan animasi yang pertama, pada bagian skin pilih **with skin**. Untuk download animasi yang berikutnya pilih **without skin**.

DOWNLOAD SETTINGS

Format **FBX for Unity(.fbx)** Skin **With Skin**

Frames per Second 30 Keyframe Reduction none

CANCEL DOWNLOAD

DOWNLOAD SETTINGS

Format **FBX for Unity(.fbx)** Skin **Without Skin**

Frames per Second 30 Keyframe Reduction none

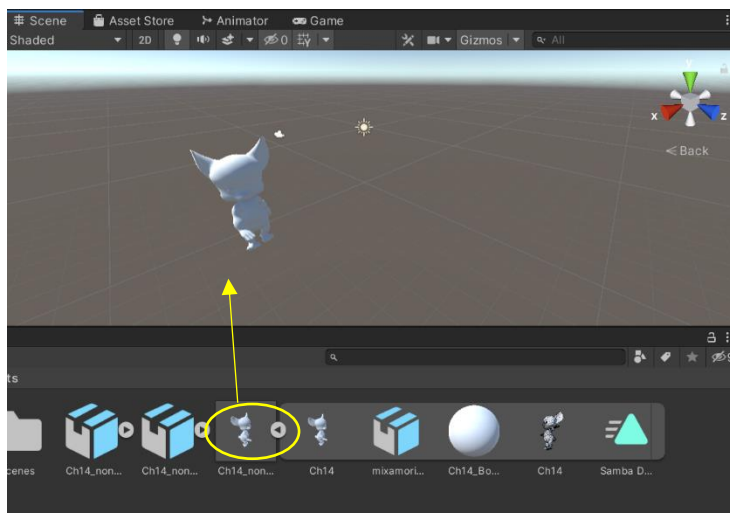
CANCEL DOWNLOAD

Berikutnya kita akan mulai membuat aplikasi di Unity.

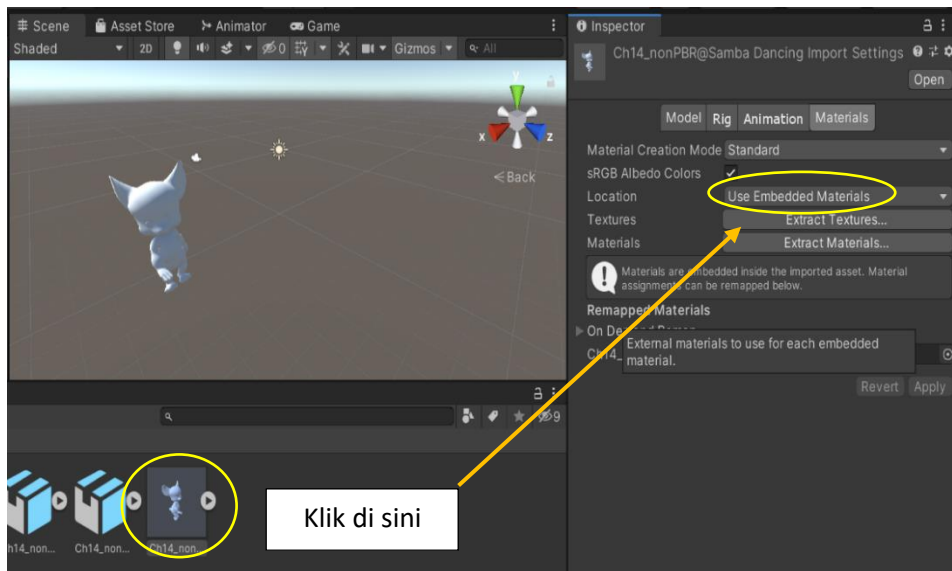
1. Buka project baru di Unity
2. Drag Objek 3D dan animasi yang telah di download dari Mixamo ke bagian Assets dengan cara drag file dari Windows explorer ke Panel Assets



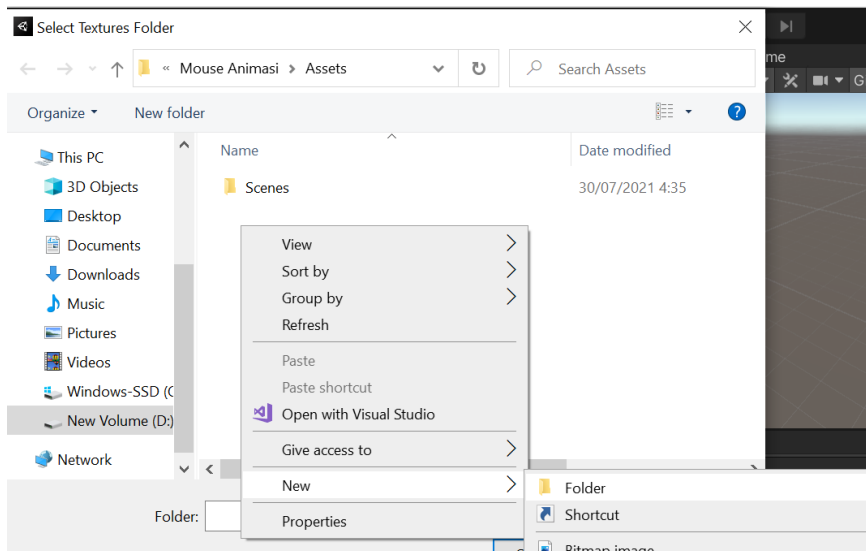
3. Klik tanda panah pada asset yang memiliki gambar karakter, kemudian drag objek ke Scene



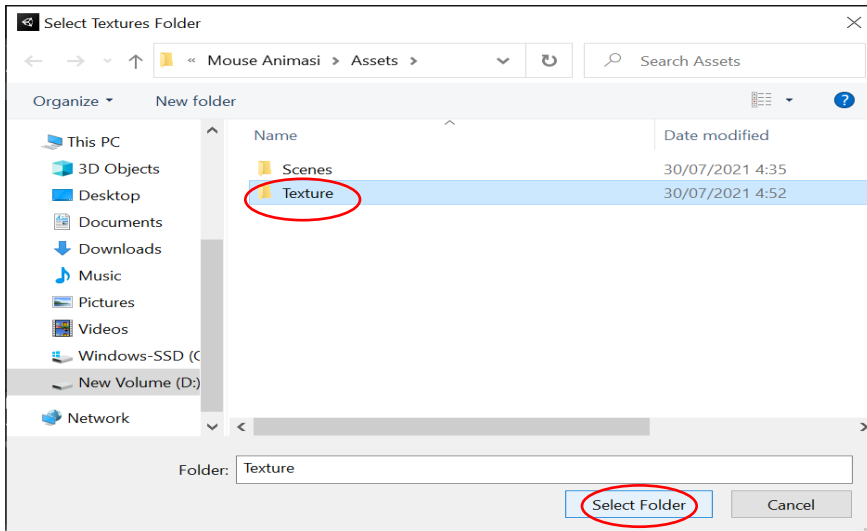
4. Pada bagian ini, object yang ditambahkan belum memiliki texture. Untuk memberinya texture, klik Object pada Assets – klik material pada inspector



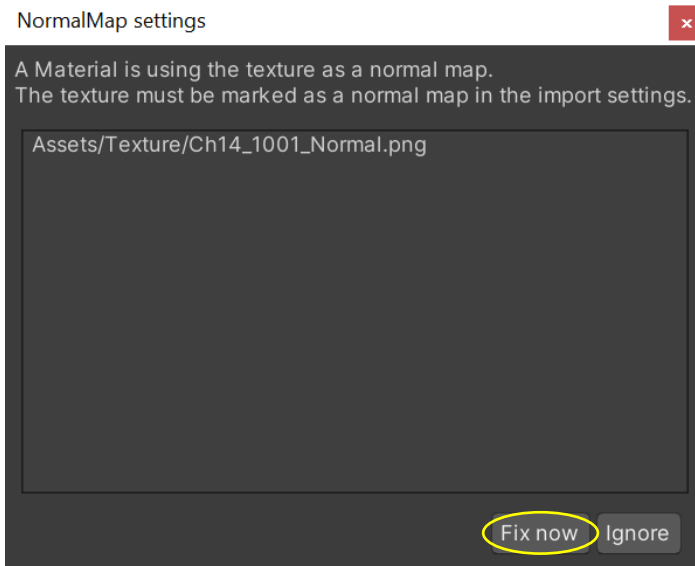
5. Klik Extract Textures. Akan muncul Tab Select Texture Folder. Buat Folder baru dengan nama Texture dengan cara klik kanan – New – Folder – beri nama Texture



6. Klik pada folder Texture kemudian klik Select Folder



7. Akan muncul NormalMap Settings. Kik Fix Now



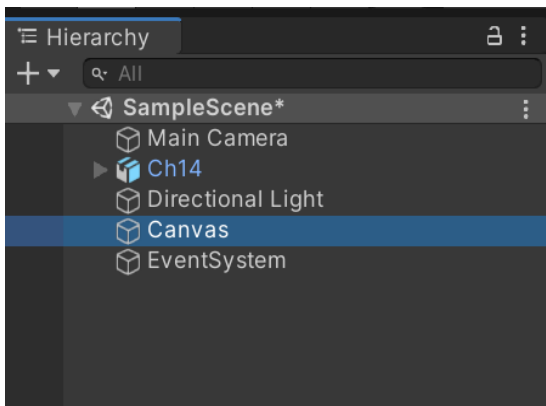
8. Tampilan Objek telah memiliki texture



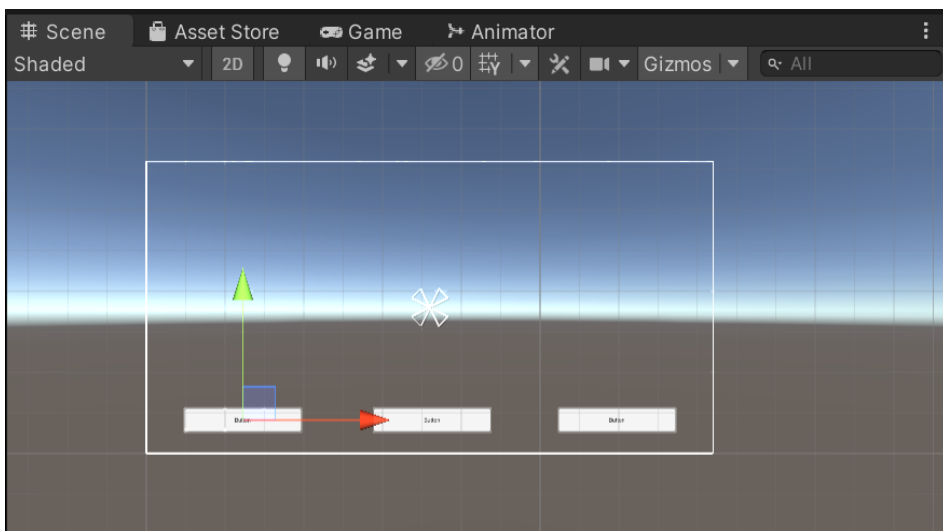
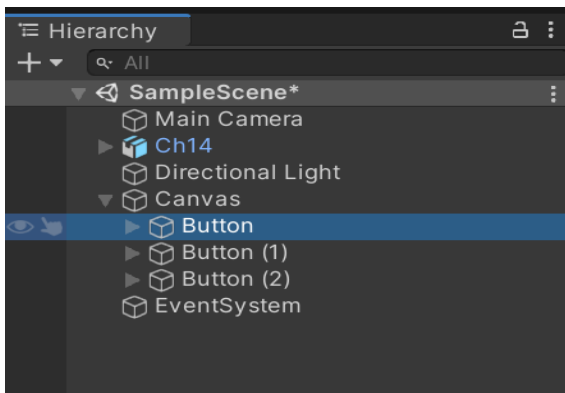
9. Atur tampilan objek sehingga pada saat di klik Tab game akan terlihat seperti berikut



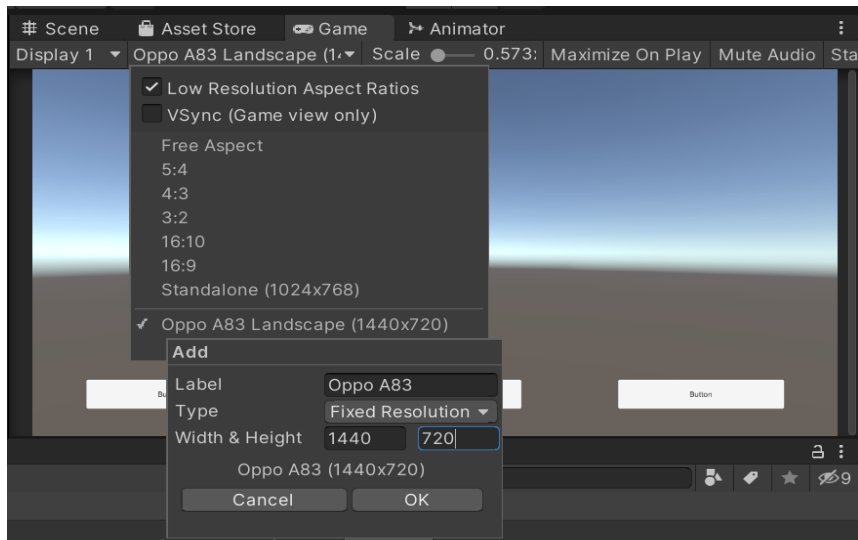
10. Selanjutnya tambahkan Canvas dengan cara klik kanan pada tab Hierarchy – UI - Canvas



11. Kemudian tambahkan 3 Button pada canvas. Cara menambahkan button, klik kanan pada canvas – UI – Button. Tempatkan button seperti pada gambar berikut



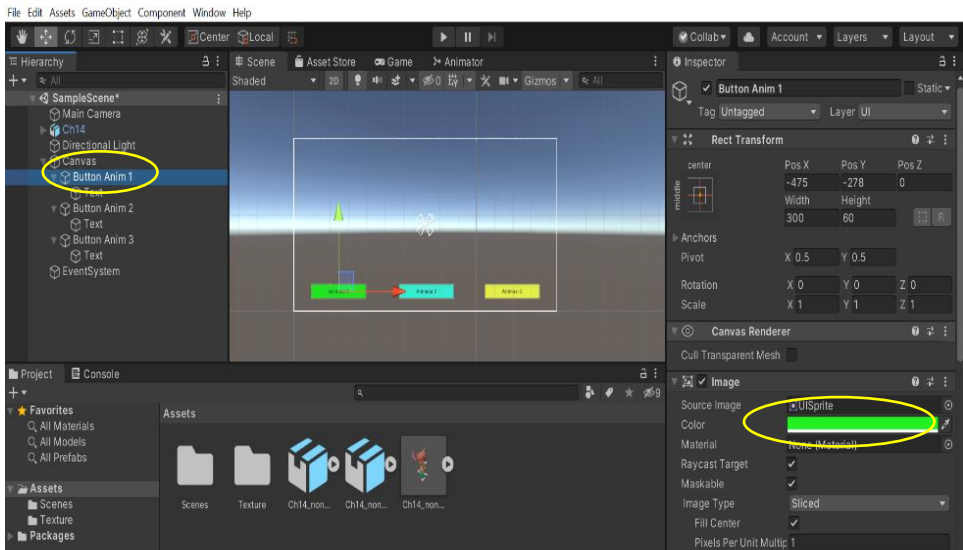
12. Atur tampilan pada game dengan cara klik pada Tab Game. Klik pada bagian bertuliskan Free Aspect, kemudian Add. Tambahkan Label dan Width & Height sesuai layer HP Anda.



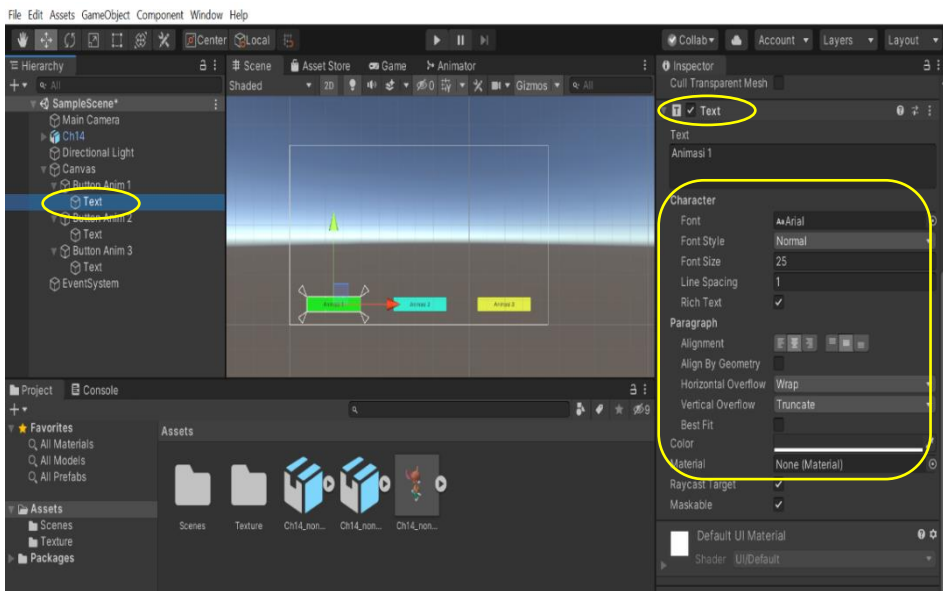
13. Tampilan pada game menjadi seperti berikut



14. Untuk mengubah warna button, klik button pada hierarchy, kemudian pada inspector klik color.



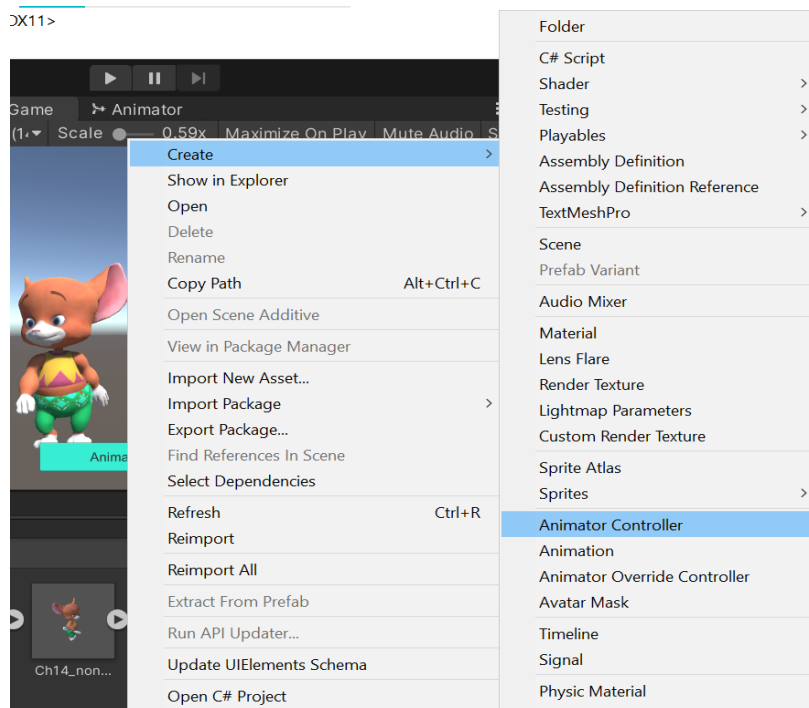
15. Untuk mengatur text, klik pada text di kolom hierarchy, kemudian atur text dari inspector



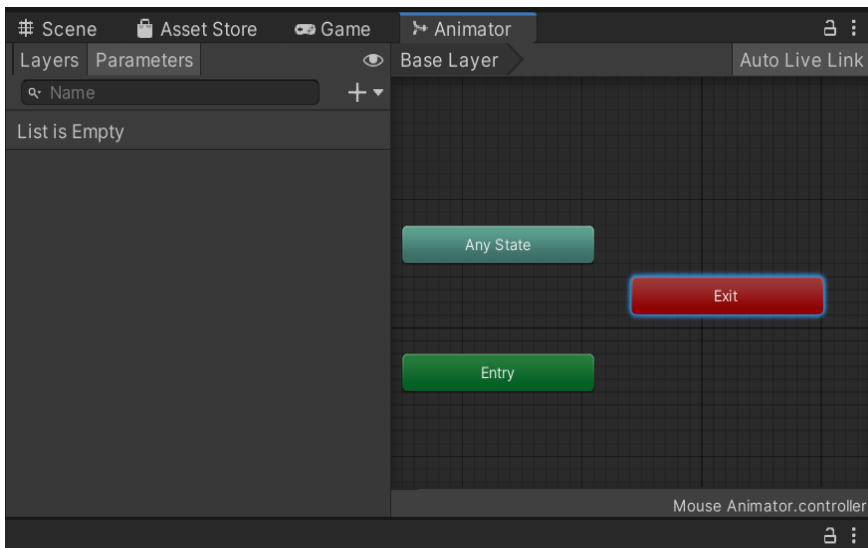
16. Tampilannya menjadi seperti berikut



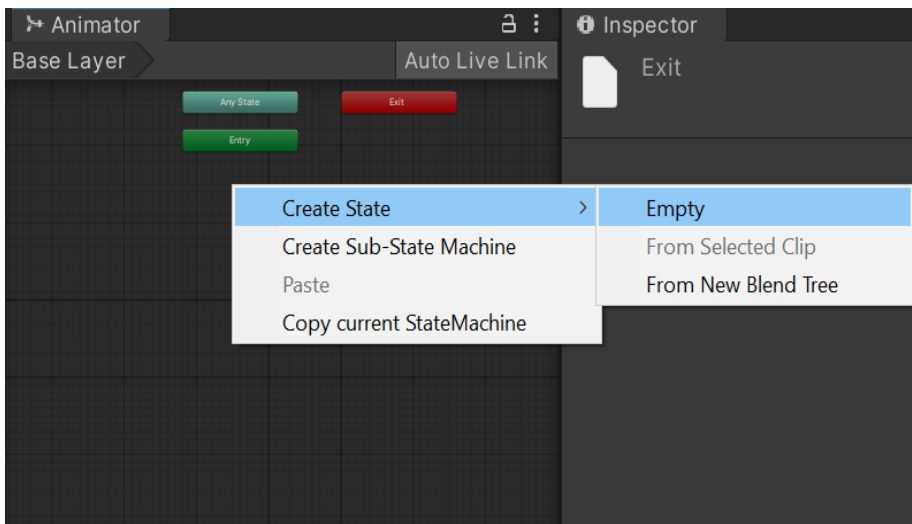
17. Berikutnya tambahkan Animator Controller untuk mengontrol animasi object dengan cara klik kanan pada Tab Assets – Create – Animator Controller



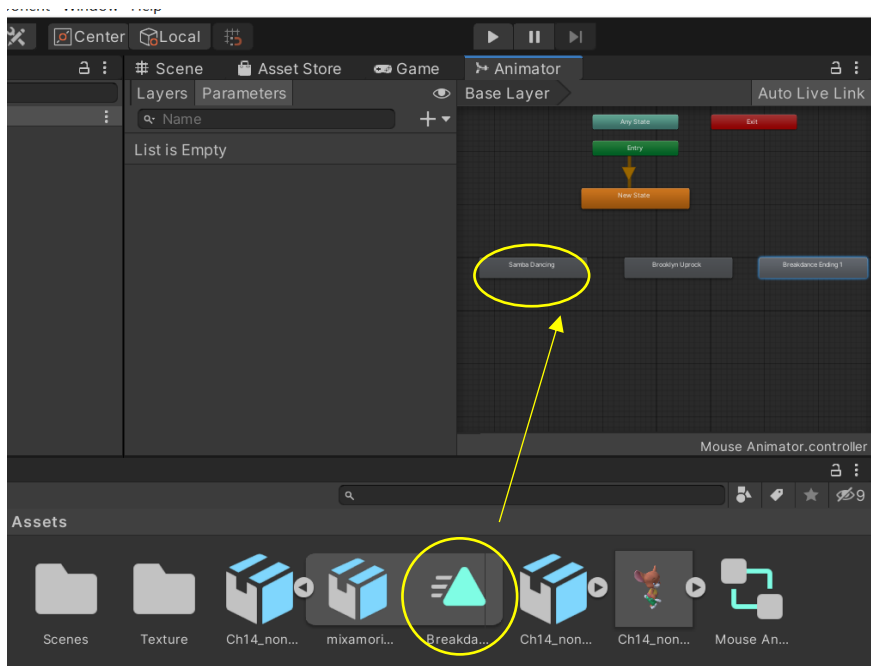
18. Tampilan awal animator controller



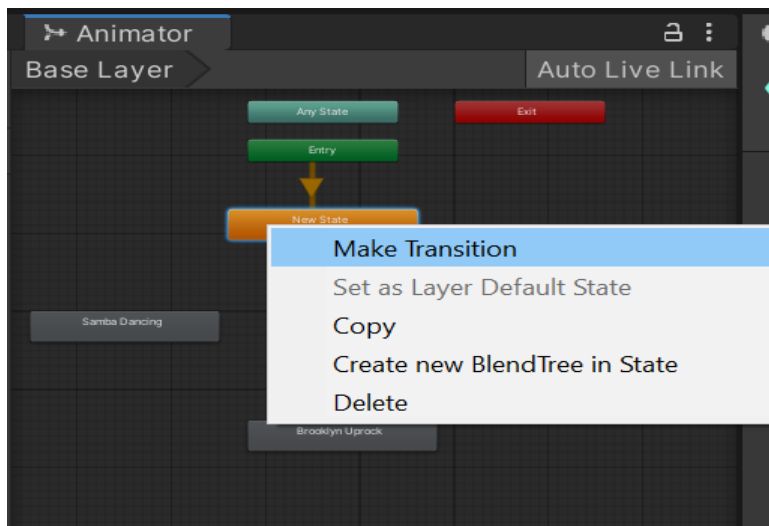
19. Klik kanan pada animator controller – create state – empty



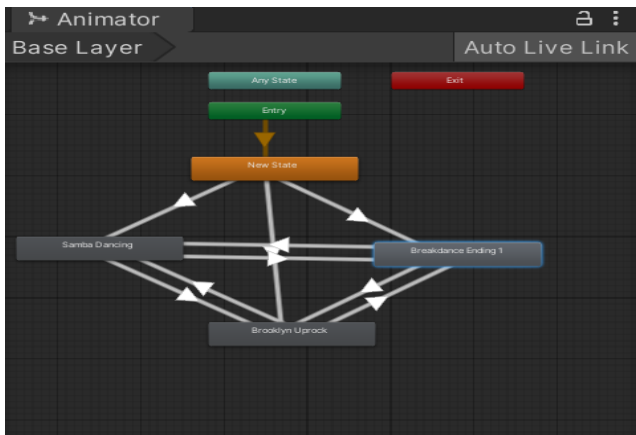
20. Drag animasi dari assets ke animator controller



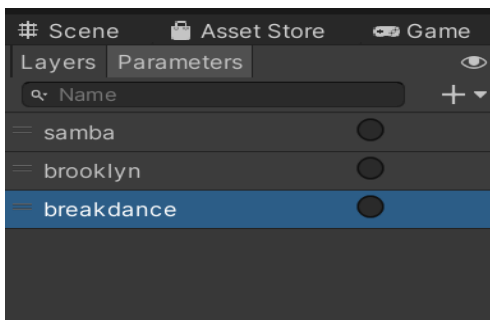
21. Beri transisi untuk setiap animasi dengan klik kanan pada New State – Make Transition. Arahkan ke setiap box



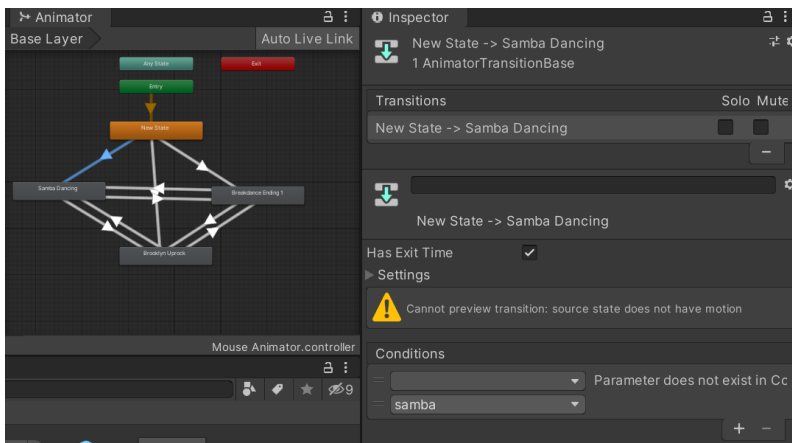
22. Lakukan hal yang sama sehingga semua box terhubung seperti berikut

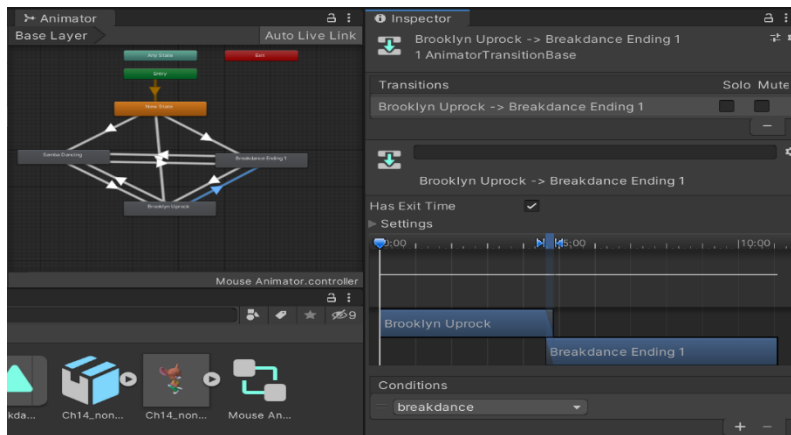


23. Klik tanda plus pada parameters – klik trigger – beri nama – enter. Lakukan sebanyak animasi yang ada.



24. Berikutnya, klik pada tanda panah pada base layer, kemudian pada condition klik tanda +. Pilih sesuai dengan arah panah. Misal dari New state ke Samba Dancing, maka pilih samba pada Condition.





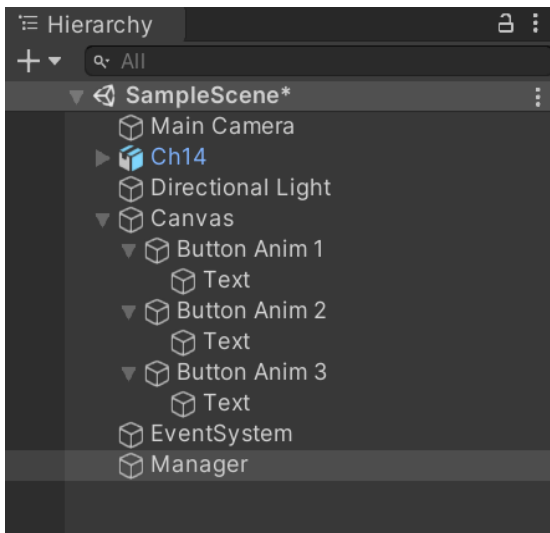
25. Setelah semua panah selesai diberi kondisi, drag animation controller ke object 3D pada Hierarchy. Berikutnya kita lanjutnya membuat script untuk memberi perlakuan pada Button. Create New C# Script, beri nama kemudian enter. Kemudian ketikkan Script berikut

```

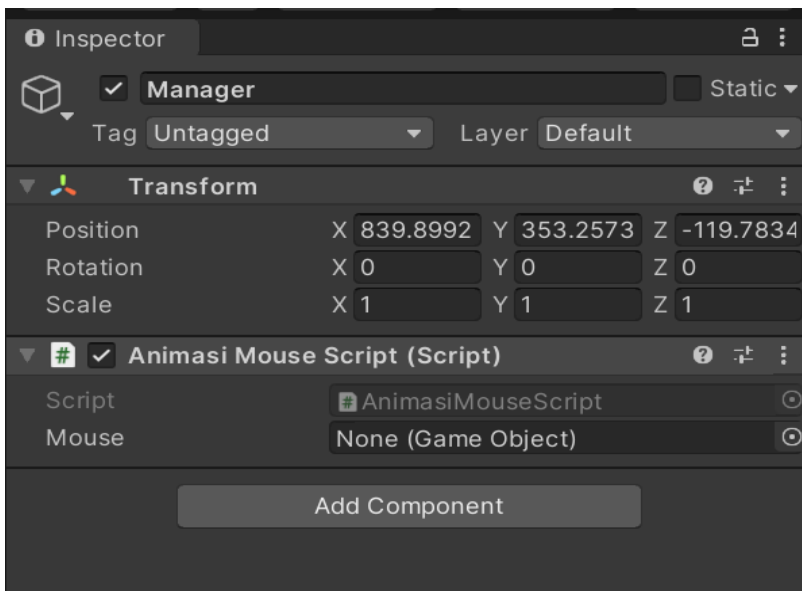
Miscellaneous Files AnimasiMouseScript
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class AnimasiMouseScript : MonoBehaviour
6  {
7      public GameObject Mouse;
8      private Animator anim;
9      // Start is called before the first frame update
10 void Start()
11 {
12     anim = Mouse.GetComponent<Animator>();
13 }
14
15 public void Samba()
16 {
17     anim.SetTrigger("samba");
18 }
19 public void Brooklyn()
20 {
21     anim.SetTrigger("brooklyn");
22 }
23 public void Breakdance()
24 {
25     anim.SetTrigger("breakdance");
26 }
27 }

```

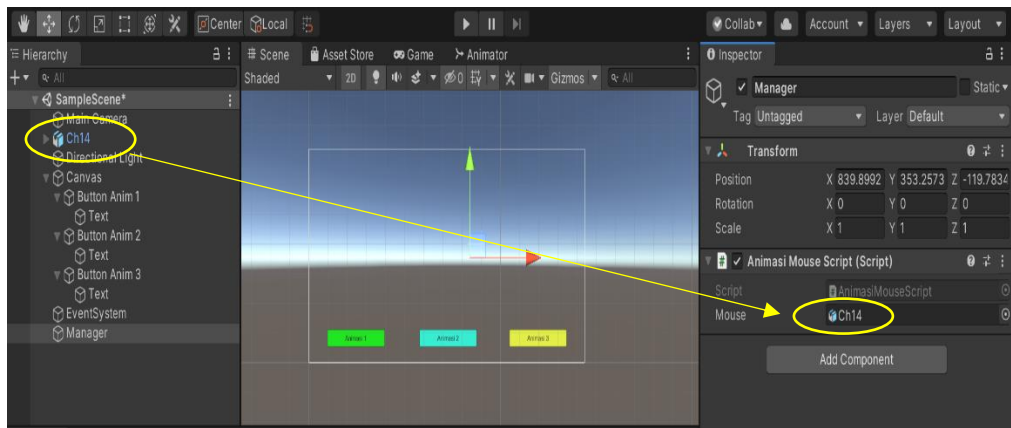
26. Selanjutnya pada Tab hierarchy, Create empty Object, beri nama Manager.



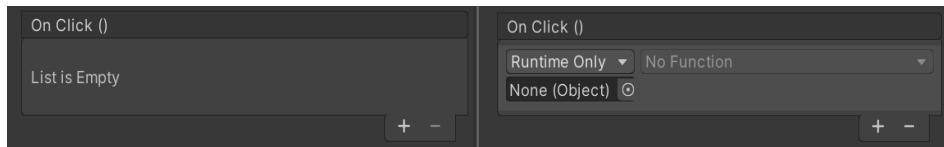
27. Drag Script ke manager, sehingga inspectornya berubah seperti pada gambar di bawah



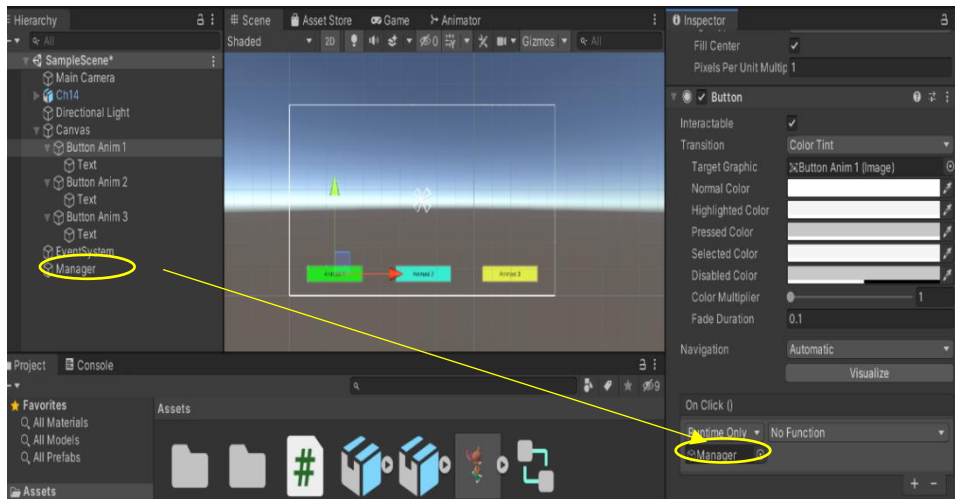
28. Klik pada Manager. Drag Objek dari Hierarchy ke Mouse pada Animasi Mouse Script



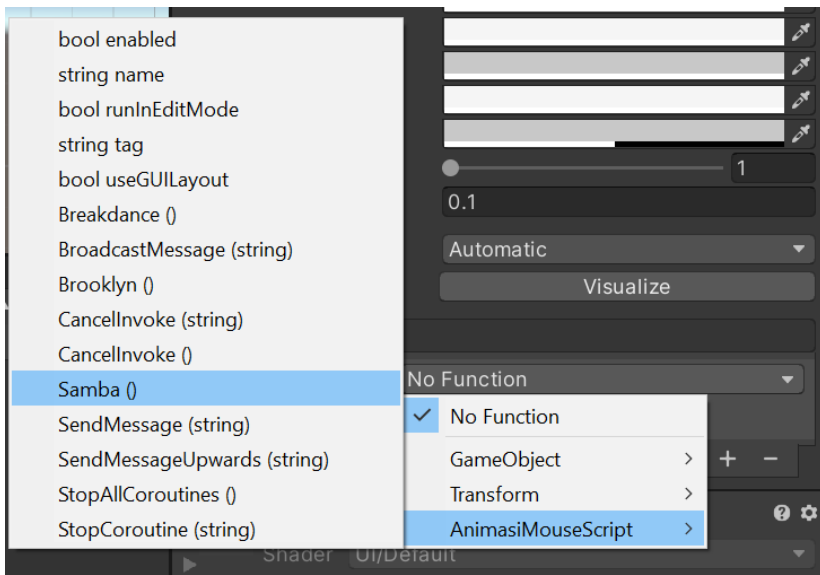
29. Berikutnya kita beri perintah pada setiap button. **Klik Button** yang pertama. Kemudian lihat pada inspector, klik tanda + pada On Click ().



30. Drag manager pada bagian None (Object)



31. Kemudian pada bagian yang bertuliskan no fuction, klik AnimasiMouseScript (sesuaikan dengan nama script Anda) kemudian pilih variable yang bersesuaian dengan button yang dipilih. Lakukan hal yang sama untuk setiap button.



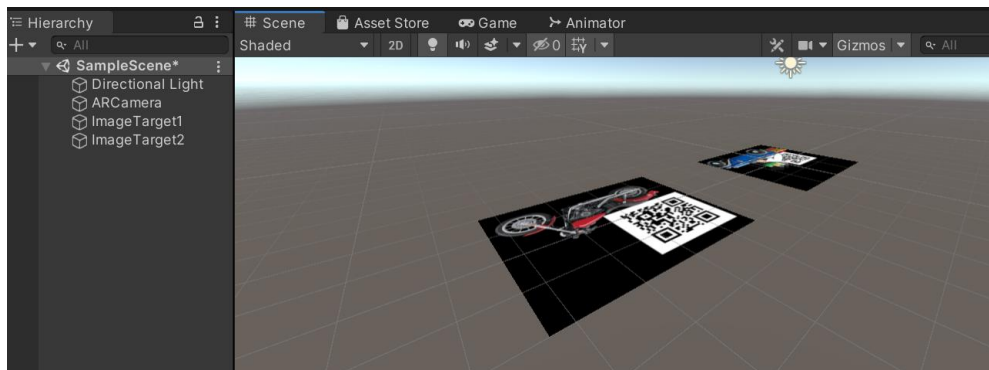
32. Jalankan game dengan cara klik tombol Play.

APLIKASI AR DENGAN LEBIH DARI 1 IMAGE TARGET

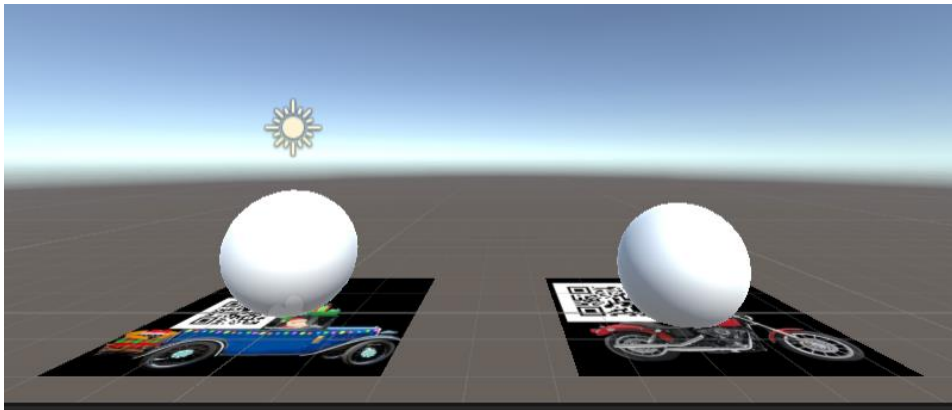
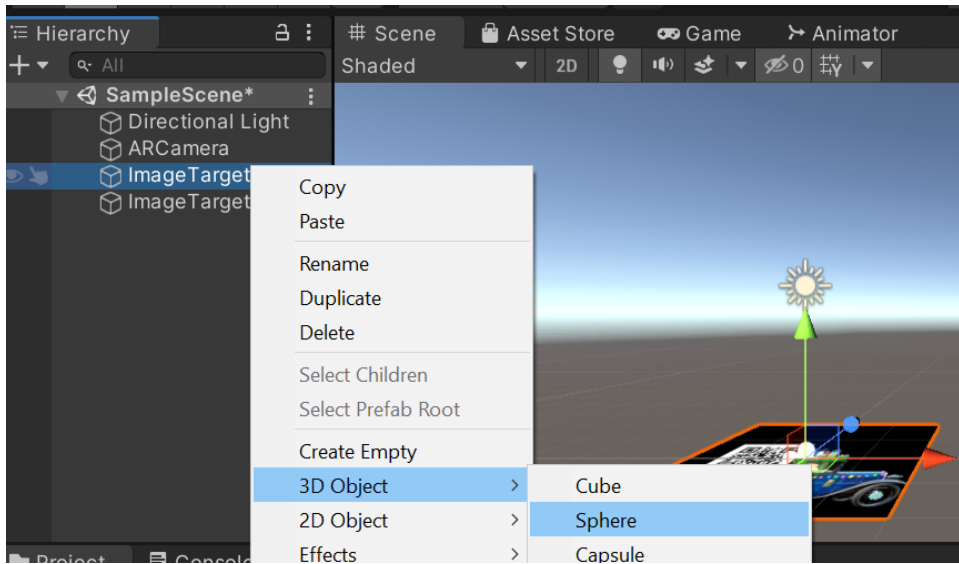
Bab ini terdiri dari 4 project. Pada project satu kita akan membuat aplikasi yang dapat mengambil komponen jarak dari GameObject dan menampilkannya pada layer. Pada project kedua kita akan membuat objek yang dapat berubah warna jika berada pada jarak tertentu. Pada project ketiga kita akan membuat suatu karakter 3D bergerak jika berada pada jarak tertentu. Sedangkan pada project 4 kita akan membuat sebuah aplikasi matematika

Project 1.

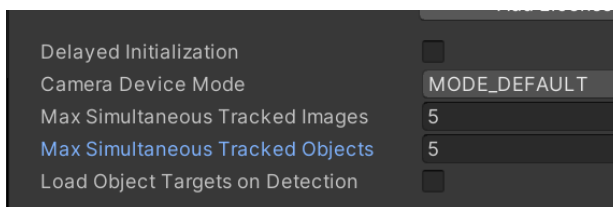
1. Buka project baru di Unity 3d
2. Dapatkan Lisence Key dan upload 2 buah Image Target ke Vuforia kemudian import database image target ke unity



3. Tambahkan 3D object berupa Sphere di atas image target dengan cara klik kanan pada image target – 3D object – Sphere

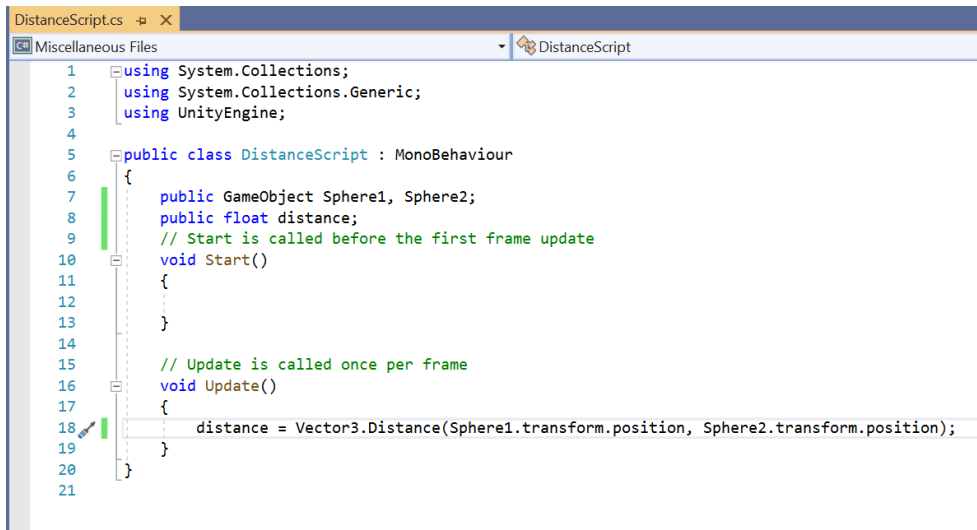


4. Pada kondisi default, AR Camera hanya dapat mendeteksi 1 buah image target dan menampilkan 1 buah object. Untuk dapat mendeteksi 2 image target secara bersamaan, klik AR Camera pada tab Hierarchy – Open Vuforia Engine – Ubah nilai Max Simultaneous Tracked Images dan Max Simultaneous Tracked Objects



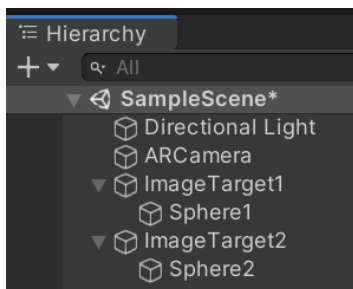
Anda dapat menuliskan sejumlah image target yang akan dimunculkan secara bersamaan.

5. Berikutnya kita tambahkan Script baru untuk menentukan jarak diantara 2 Sphere. Klik kanan pada Tab Assets - Create – C# Script – beri nama DistanceScript. Ketikkan Script di bawah ini

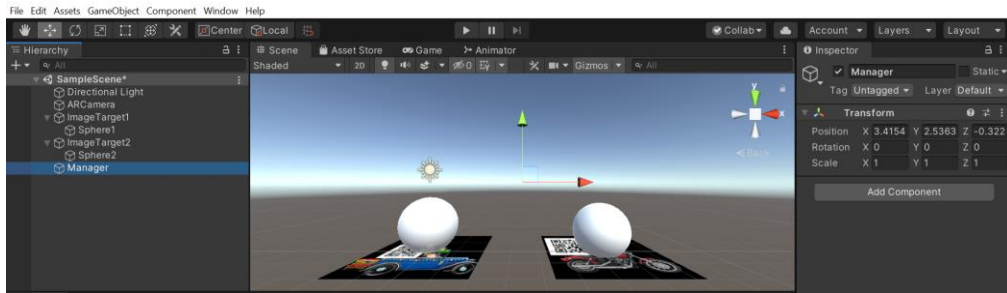


```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class DistanceScript : MonoBehaviour
6 {
7     public GameObject Sphere1, Sphere2;
8     public float distance;
9     // Start is called before the first frame update
10    void Start()
11    {
12    }
13
14
15    // Update is called once per frame
16    void Update()
17    {
18        distance = Vector3.Distance(Sphere1.transform.position, Sphere2.transform.position);
19    }
20
21 }
```

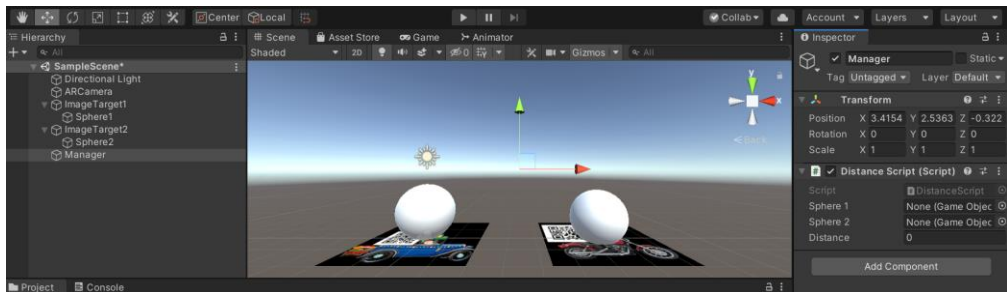
6. Simpan Script kemudian Kembali ke Unity.
7. Untuk memudahkan dalam mengenali Sphere 1 dan Sphere 2, ganti nama Sphere pada Hierarchy dengan Sphere1 dan Sphere2



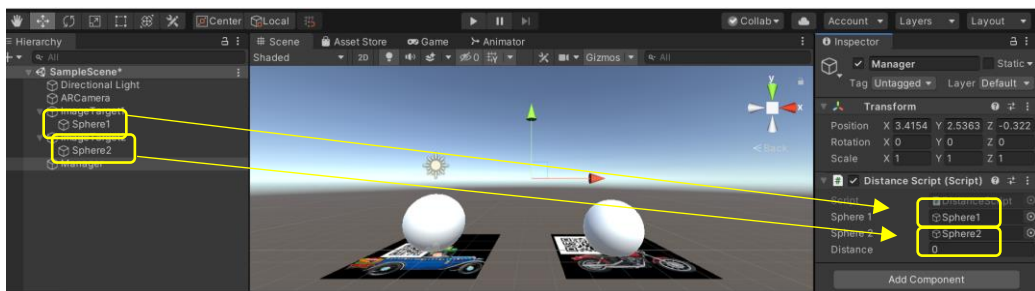
8. Selanjutnya kita tambahkan manager dengan cara Klik kanan pada Tab Hierarchy-create empty. Kemudian pada Inspector ubah nama GameObject menjadi Manager



9. Drag DistanceScript dari Assets ke Manager pada Hierarchy atau pada inspector (bisa salah satu). Tampilan Inspector Manager menjadi seperti berikut.



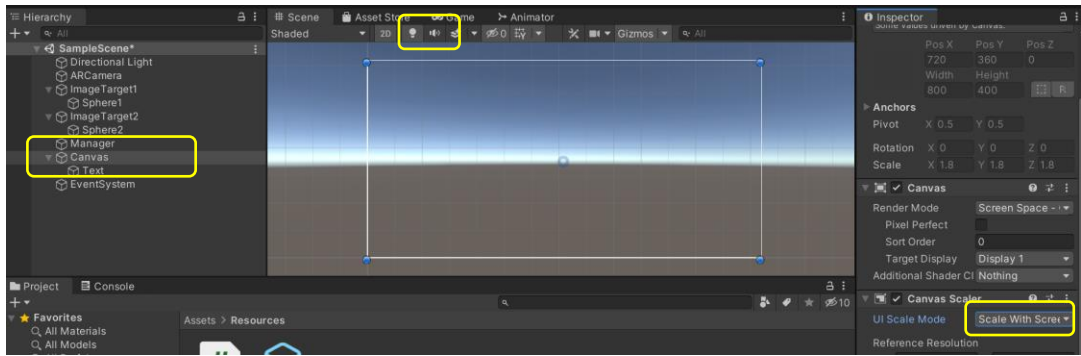
10. Drag Sphere 1 dan Sphere 2 dari Hierarchy ke Inspector Manager



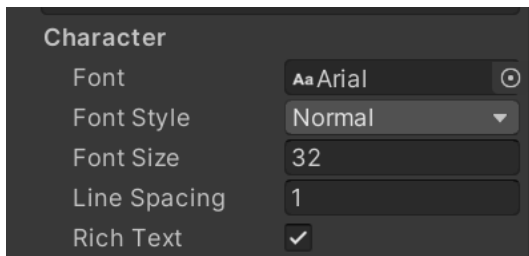
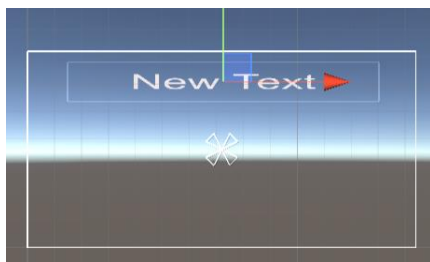
11. Klik Play untuk mencoba.

12. Untuk memunculkan jarak antara kedua Sphere di layar, kita perlu menambahkan text pada hierarchy dengan cara klik kanan pada hierarchy – UI – text. Ubah tampilan menjadi 2D.

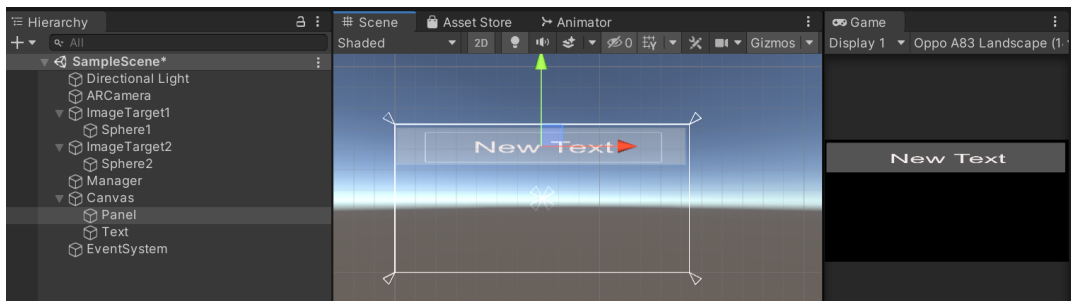
13. Kemudian Klik pada Canvas. Pada Inspector Canvas – Canvas Scaler – UI Scale Mode. Ubah UI Scale Mode menjadi Scale with Screen



14. Atur text supaya terlihat dengan jelas pada Game



15. Agar tulisan dapat terlihat dengan lebih jelas, tambahkan panel dengan cara klik canvas pada hierarchy – UI – Panel. Atur letak panel supaya berukuran sama dengan text.



16. Kemudian buka Distance Script dan ketikkan Script berikut

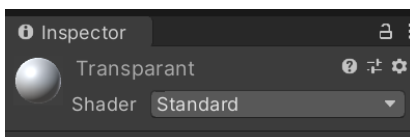
```
DistanceScript.cs  X
Miscellaneous Files  DistanceScript
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class DistanceScript : MonoBehaviour
7  {
8      public GameObject Sphere1, Sphere2;
9      public float distance;
10     public Text label;
11     // Start is called before the first frame update
12     void Start()
13     {
14     }
15
16
17     // Update is called once per frame
18     void Update()
19     {
20         distance = Vector3.Distance(Sphere1.transform.position, Sphere2.transform.position);
21         label.text = "Distance is " + distance.ToString() + " cm";
22     }
23
24 }
```

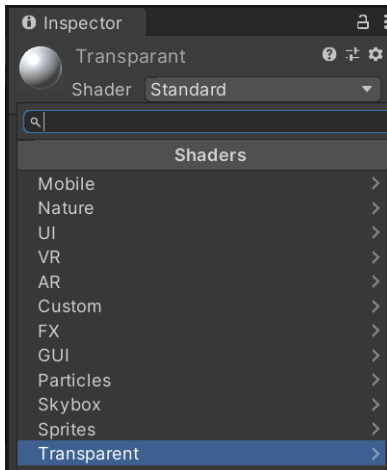
17. Klik tombol play untuk mencoba

Project 2

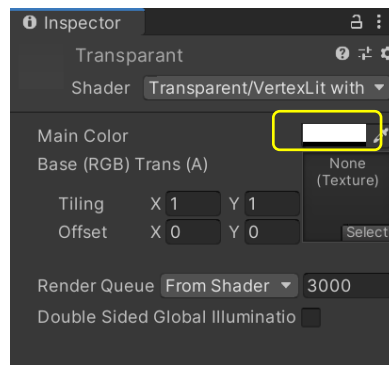
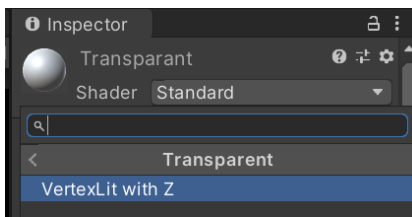
Kita akan memodifikasi aplikasi di atas sehingga dua buah sphere yang semula ada menjadi tidak terlihat (transparent), kemudian ada sebuah cube yang akan berubah warna jika jarak antara kedua Sphere kurang dari jarak tertentu.

1. Untuk membuat Sphere menjadi transparent, buat material baru dengan cara klik kanan pada Assets – Create – Material – beri nama Transparant
2. Pada inspector Material Transparant, klik pada shader kemudian klik pada tanda panah ke bawah, pilih Transparent





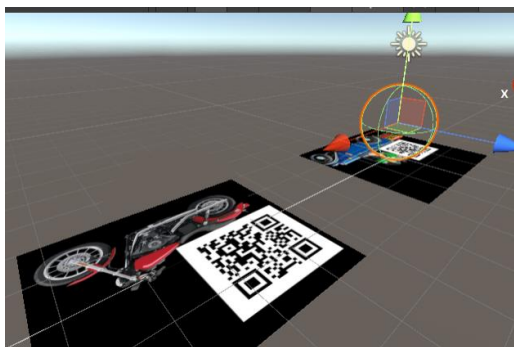
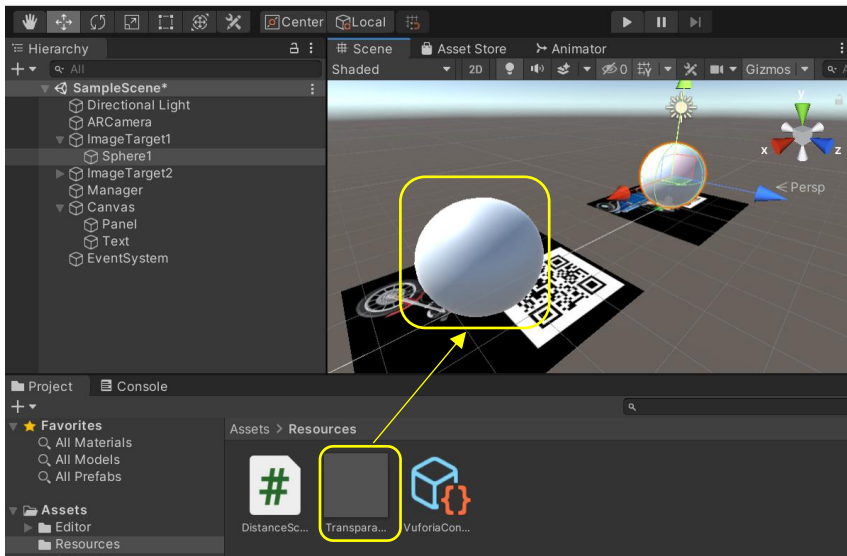
3. Klik VertexLit with Z kemudian klik warna putih pada Main color



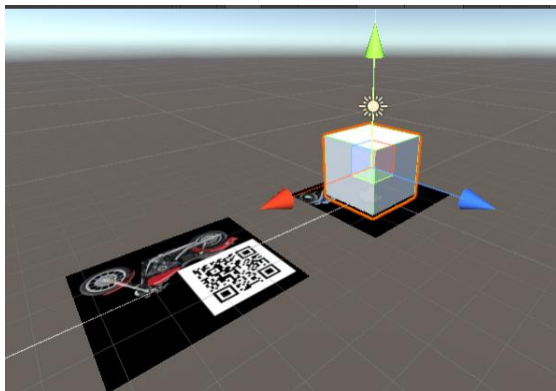
4. Drag pada bagian A sehingga bernilai 0



5. Drag material Transparent ke Sphere 1 dan Sphere 2



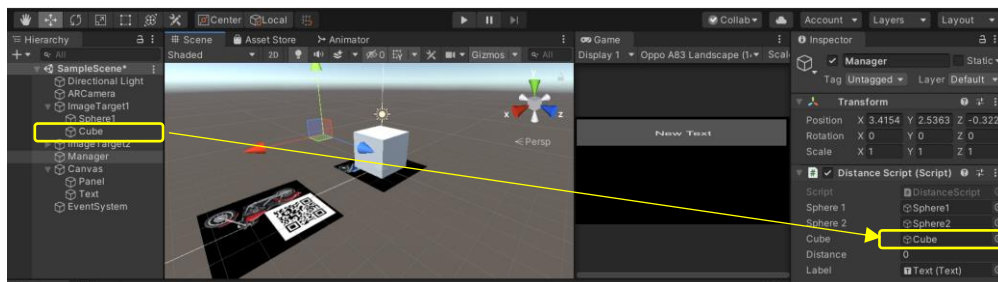
6. Berikutnya tambahkan Cube di atas Image Target 2



7. Ubah Script menjadi seperti berikut

```
DistanceScript.cs* -> X
Miscellaneous Files -> DistanceScript
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class DistanceScript : MonoBehaviour
7 {
8     public GameObject Sphere1, Sphere2, cube;
9     public float distance;
10    public Text label;
11    // Update is called once per frame
12    void Update()
13    {
14        distance = Vector3.Distance(Sphere1.transform.position, Sphere2.transform.position);
15        label.text = "Distance is " + distance.ToString() + " cm";
16        if (distance < 3)
17        {
18            cube.GetComponent<Renderer>().material.color = Color.red;
19        }
20        else if (distance > 3 && distance < 6)
21        {
22            cube.GetComponent<Renderer>().material.color = Color.yellow;
23        }
24        else
25        {
26            cube.GetComponent<Renderer>().material.color = Color.green;
27        }
28    }
29 }
30
```

8. Drag Cube dari Hierarchy ke Inspector

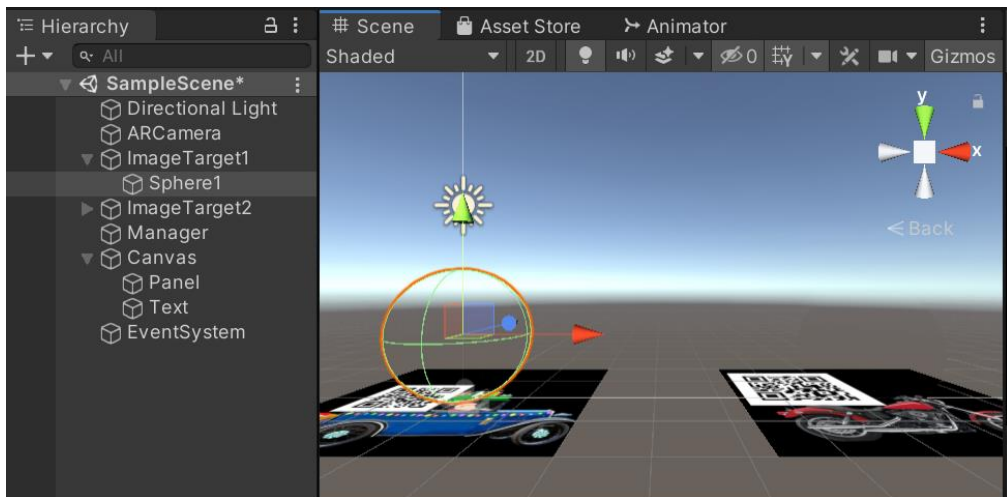


9. Klik Play untuk melihat hasilnya

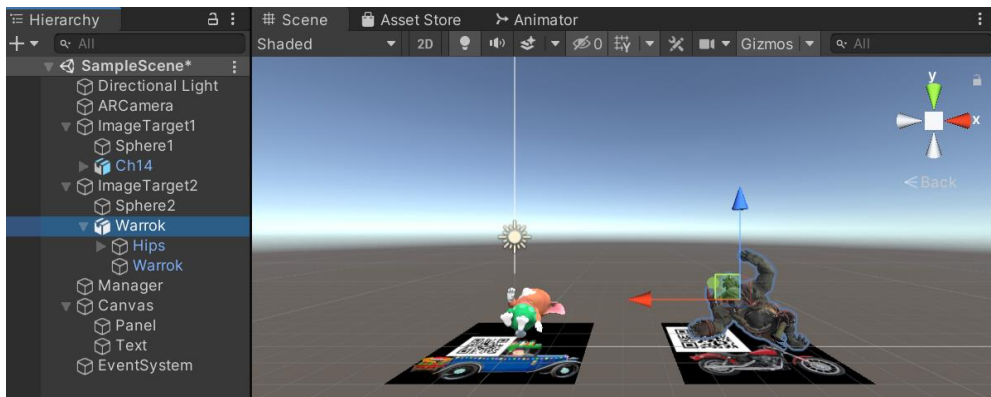
Project 3

Berikutnya kita akan memodifikasi aplikasi di atas sehingga yang muncul adalah dua buah karakter animasi yang dapat bergerak jika keduanya berada pada jarak tertentu

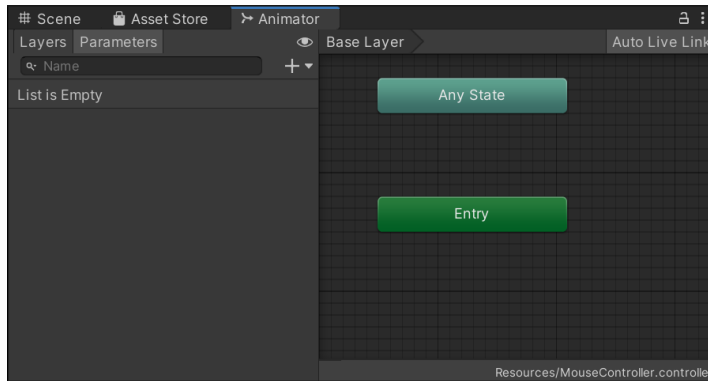
1. Langkah pertama, hapus cube. Pastikan dua buah Sphere transparent masih berada di atas image target.



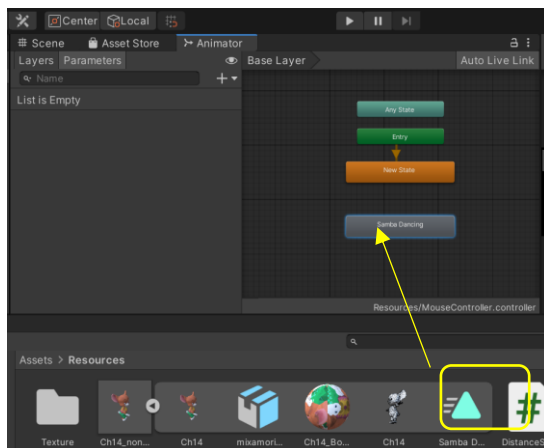
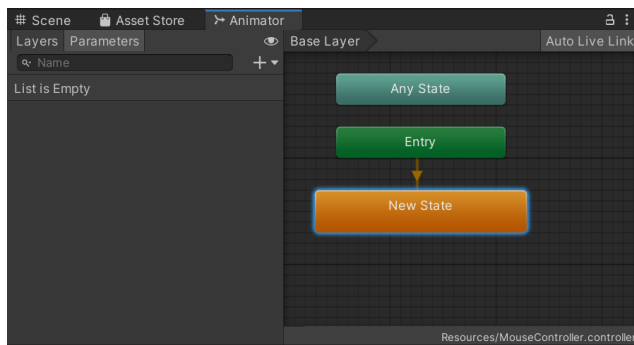
2. Tambahkan karakter 3D ke atas masing-masing image target.



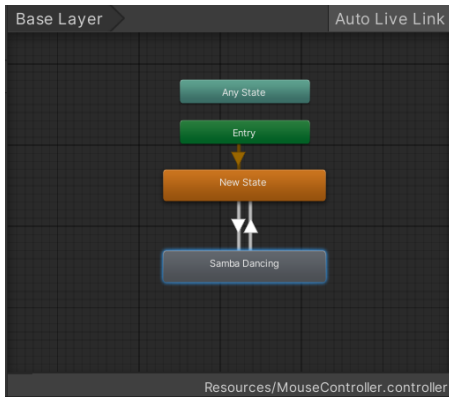
3. Tambahkan animation controller dengan cara klik kanan pada Assets – create – animation controller. Beri nama MouseController
4. Double click untuk membuka animation controller



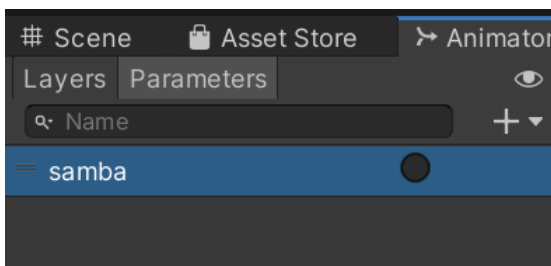
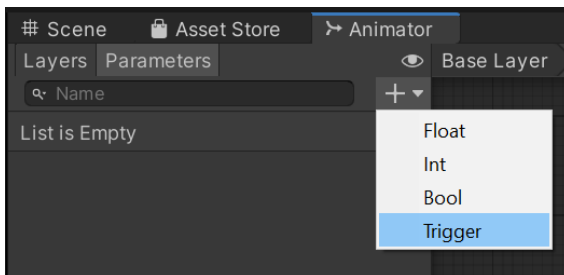
5. Tambahkan New State dengan cara klik kanan pada Base Layer – Create State – empty. Kemudian drag animasi dari Assets ke Base Layer



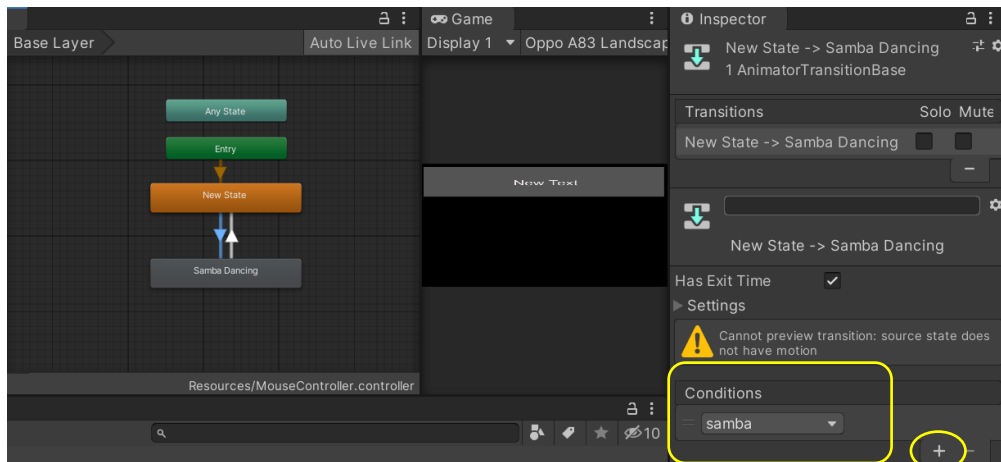
6. Klik kanan pada New State – Make Transition. Arahkan pada Samba Dancing. Kemudian klik kanan pada Samba Dancing – Make Transition. Arahkan ke New state



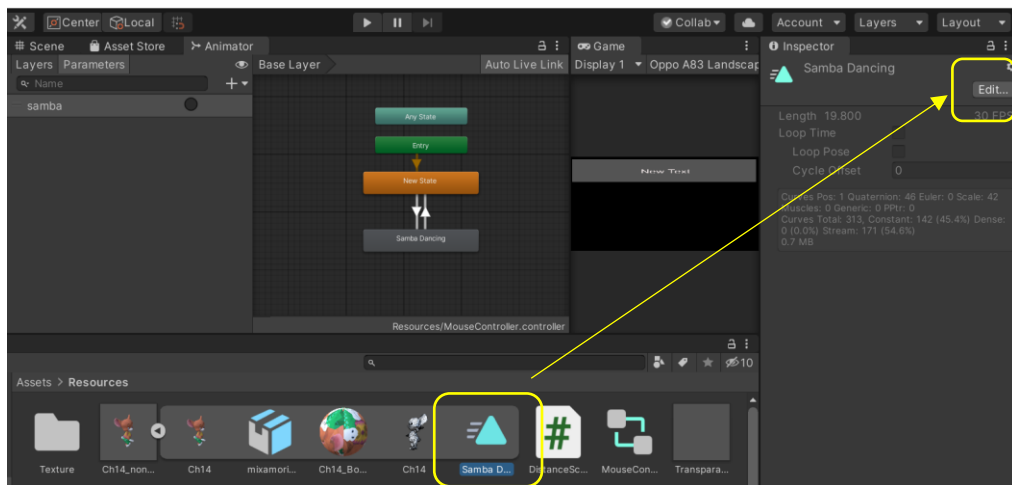
7. Berikutnya tambahkan trigger dengan cara klik tanda + pada Parameters – Trigger – Beri nama samba kemudian Enter

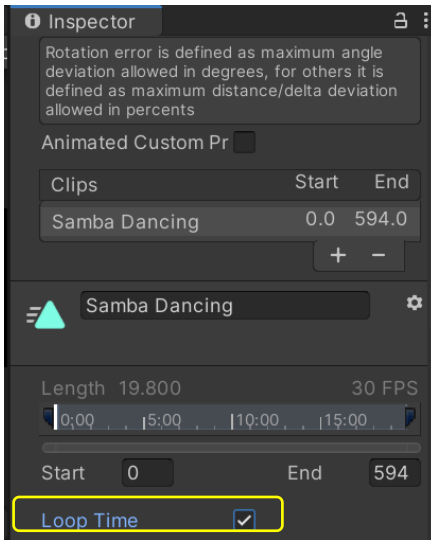


8. Tambahkan kondisi dengan cara klik pada tanda panah di Base Layer. Kemudian pada inspector lihat pada Conditions. Klik tanda panah kemudian pilih trigger yang sesuai (pada hal ini triggernya adalah samba)

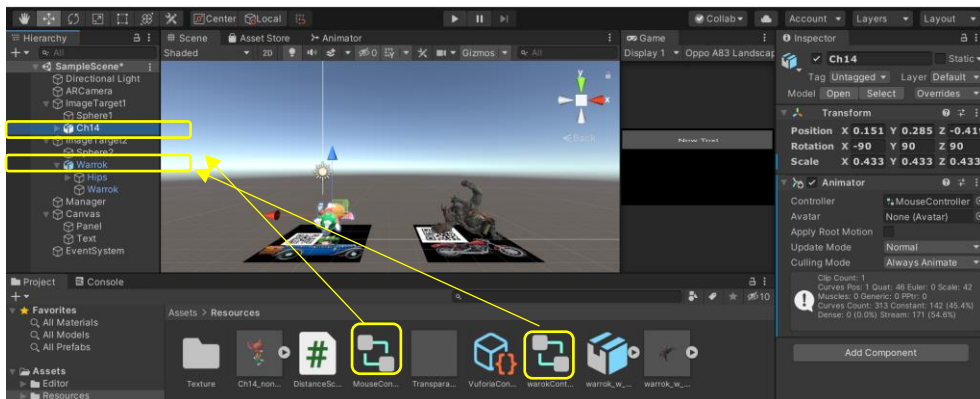


9. Agar animasi dapat berulang, kita harus menambahkan loop time dengan cara klik animasi pada Assets, kemudian klik edit pada inspector. beri tanda centang pada Loop Time kemudian klik Apply





10. Lakukan hal yang sama untuk karakter 3D yang berikutnya (Warok)
11. Berikutnya drag animation controller ke masing-masing karakter pada hierarchy



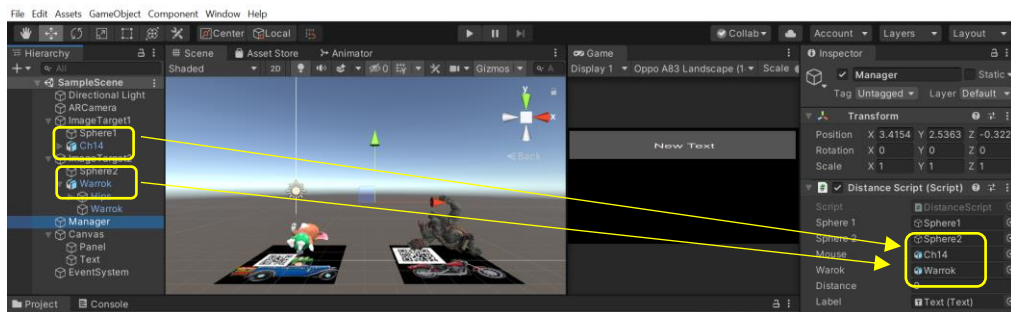
12. Edit Distance Script menjadi seperti di bawah ini

```

DistanceScript.cs
Miscellaneous Files
DistanceScript
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5
6 public class DistanceScript : MonoBehaviour
7 {
8     public GameObject Sphere1, Sphere2, mouse, warok;
9     public float distance;
10    public Text label;
11    private Animator MouseAnim, WarokAnim;
12
13    void Start()
14    {
15        MouseAnim = mouse.GetComponent<Animator>();
16        WarokAnim = warok.GetComponent<Animator>();
17    }
18
19    void Update()
20    {
21        distance = Vector3.Distance(Sphere1.transform.position, Sphere2.transform.position);
22        label.text = "Distance is " + distance.ToString() + " cm";
23        if (distance < 5)
24        {
25            MouseAnim.SetTrigger("samba");
26            WarokAnim.SetTrigger("capoeira");
27        }
28    }
29 }
30

```

13. Klik Manager. Drag karakter dari Hierarchy ke Manager Inspector



14. Klik tombol Play untuk mencoba

Project 4.

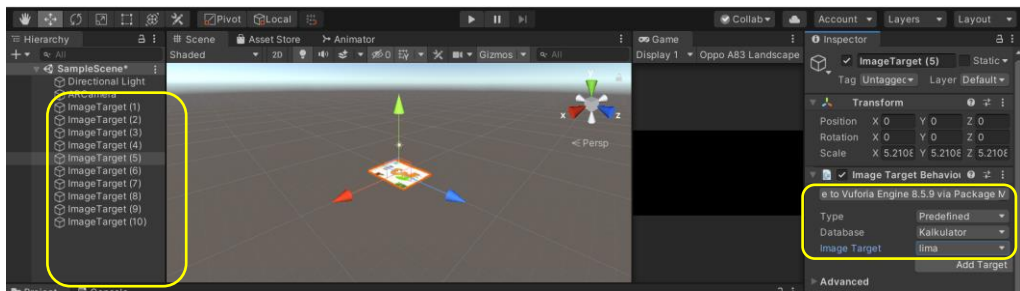
Pada bagian ini kita akan membuat aplikasi matematika dengan menggunakan beberapa image target.

1. Langkah pertama yang perlu dilakukan adalah menyiapkan target marker. Misalnya seperti gambar berikut





2. Upload target market di Vuforia kemudian import di unity.
3. Dapatkan lisence key dari Vuforia kemudian masukkan ke Unity pada bagian inspector AR Camera
4. Masukkan 10 image target. Masukkan image target yang sesuai.



5. Atur image target supaya mudah dikenali



6. Tambahkan image target baru, beri nama IMStars



7. Kita akan memasukkan gambar bintang di atas IMStars. Dapatkan objek 3D stars di <https://assetstore.unity.com/> , temukan Simple Collectible Packs, kemudian import



DILLON WALLACE

Simple Collectibles Pack

(not enough ratings) | ❤️ (10)

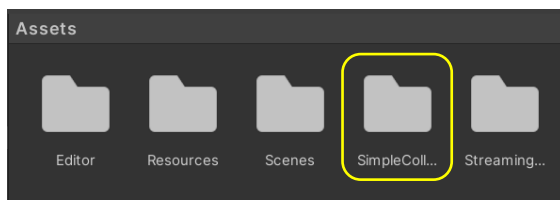
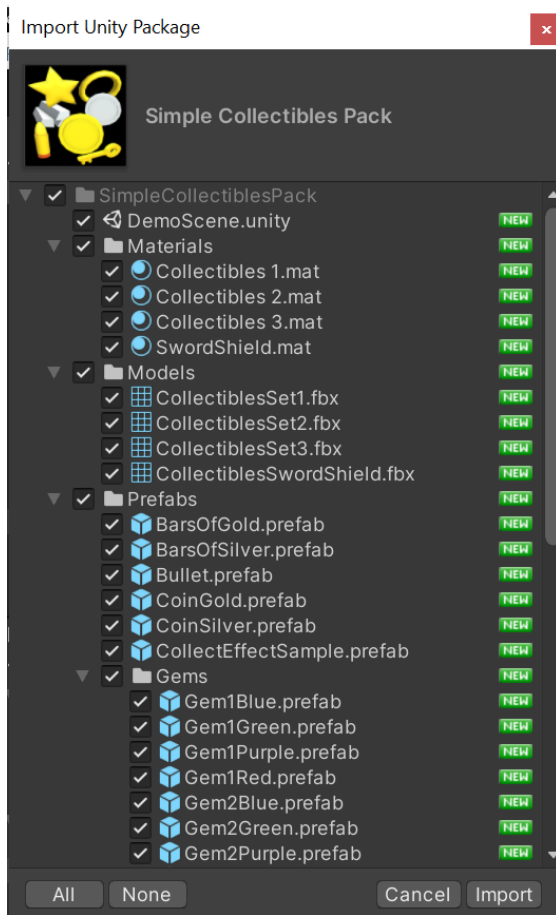
FREE

The screenshot shows the Unity Package Manager window. On the left, a list of packages is displayed with 'Simple Collectibles Pack' highlighted. The right pane shows the details for this pack, including its version (1.0.0), author (Dillon Wallace), and a description: 'Set your players on a hunt with the Simple Collectibles Pack! With over 30 Collectibles to choose from, ranging from More...'. At the bottom right, 'Import' and 'Download' buttons are visible and highlighted with a yellow box.

Package Name	Version
3LE Low Poly Cloud Pack	1.0.0
Cartoon Cat	1.0.0
Civil Transport Aircraft	1.0.0
Colonial Ship	1.0.0
Dark Dragon	1.0.0
Free chibi cat	1.0.0
Free Trees	1.0.0
Freight Train	1.0.0
HQ Racing Car Model No.1203	1.0.0
low poly car	1.3.0
Low-poly Sports car #20	1.0.0
Simple Cars Pack	1.1.0
Simple Collectibles Pack	1.0.0
Standard Assets (for Unity 2018)	1.1.6
Super Pivot FREE Modifier	2.30.0
Toon Fox	1.0.0
Unity-Chan! Model	1.2.2

Simple Collectibles Pack
Version 1.0.0 - July 31, 2018 [asset store](#)
[3D Props](#)
Links
[View in the Asset Store](#)
[Publisher Website](#)
[Publisher Support](#)
Author
[Dillon Wallace](#)
Published Date
July 31, 2018
Set your players on a hunt with the Simple Collectibles Pack!
With over 30 Collectibles to choose from, ranging from [More...](#)
Supported Unity Versions
5.5.4 or higher
Package Size
Size: 451.11 KB (Number of files: 47)
Supporting Images

Last update Aug 2, 04:24 [Refresh](#) [Import](#) [Download](#)



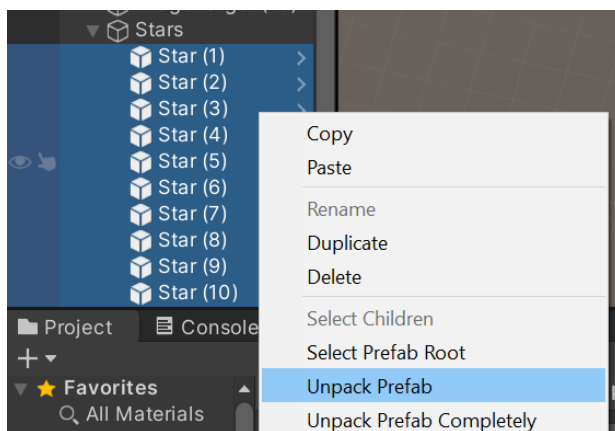
8. Buka Assets SimpleCollectiblePacks – Prefabs – Temukan Star – Drag ke atas image target IMStars



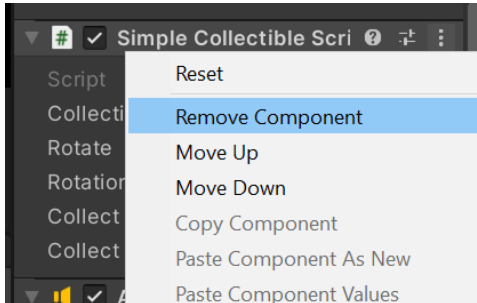
- Duplicate star sehingga di atas image target IMStars ada 10 bintang. Tekan Ctrl+D untuk Duplicate kemudian atur posisinya.



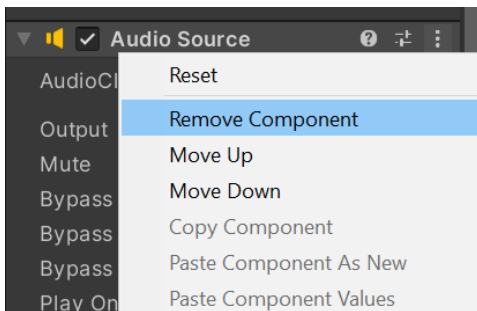
- Perhatikan bahwa setiap star memiliki Script dan Sound. Kita dapat membuang script dan sound dengan meng-unpack prefabs terlebih dahulu. Select / blok semua star – klik kanan – Unpack Prefab



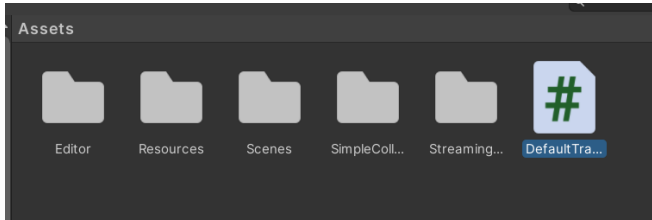
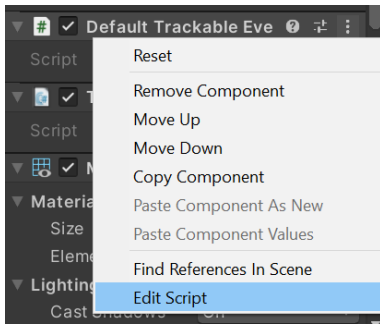
11. Untuk menghapus script pada kesepuluh star secara bersamaan, select semua star – pada inspector panel klik titik tiga pada Simple Collectible Script – Remove Component



12. Untuk menghapus sound secara bersamaan - select semua star – pada inspector panel klik titik tiga pada Audio Source – Remove Component



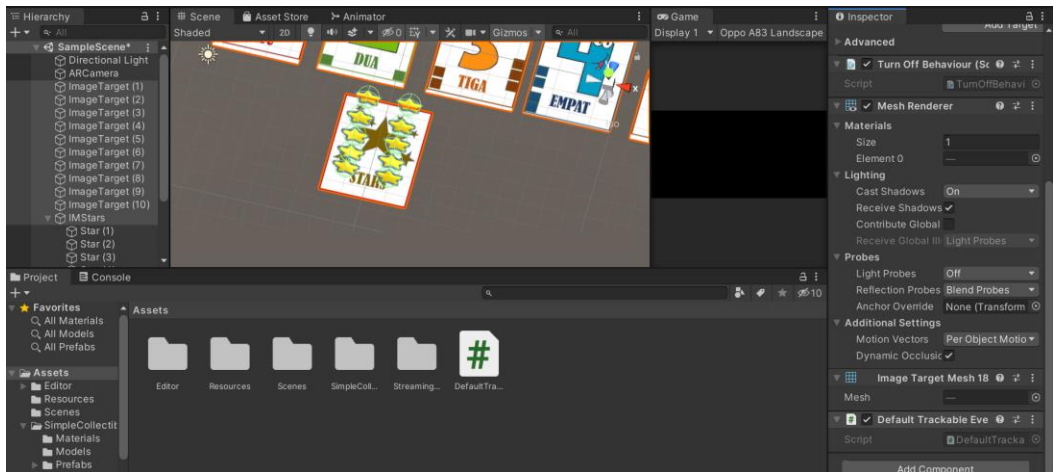
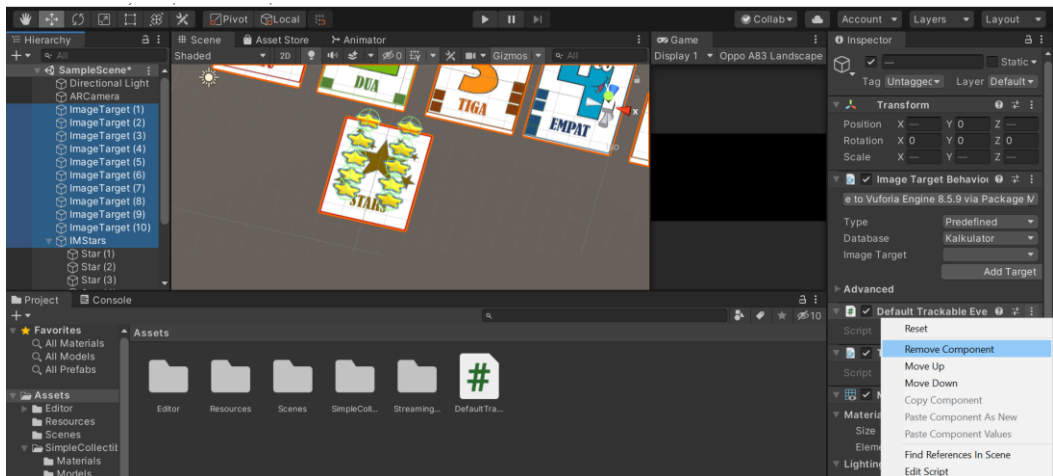
13. Pada aplikasi ini kita akan membuat Star pada IMStar muncul sejumlah angka yang ada. Misal jika kita arahkan kamera ke angka 2 dan gambar bintang, maka akan muncul 2 buah bintang di atas image target IMStar. Langkah yang perlu dilakukan adalah membuat duplicate dari Default Trackable Event Handler terlebih dahulu.



```
DefaultTrackableEventHandler.cs - X
Miscellaneous Files | DefaultTrackableEventHandler
1  /*=====
2  Copyright (c) 2017 PTC Inc. All Rights Reserved.
3
4  Copyright (c) 2010-2014 Qualcomm Connected Experiences, Inc.
5  All Rights Reserved.
6  Confidential and Proprietary - Protected under copyright and other laws.
7  =====*/
8
9  using UnityEngine;
10 using Vuforia;
11
12  /// <summary>
13  /// A custom handler that implements the ITrackableEventHandler interface.
14  ///
15  /// Changes made to this file could be overwritten when upgrading the Vuforia version.
16  /// When implementing custom event handler behavior, consider inheriting from this class instead.
17  /// </summary>
18  public class DefaultTrackableEventHandler : MonoBehaviour, ITrackableEventHandler
19  {
20  #region PROTECTED_MEMBER_VARIABLES
21
22  #endregion
23
24  #region UNITY_MONOBEHAVIOUR_METHODS
25
26  #endregion
27
28  #region PUBLIC_METHODS
29
30  #endregion
31
32  #region PROTECTED_METHODS
33
34  #endregion
35
36  }
125
```

```
DefaultTrackableEventHandler.cs - X | DefaultTrackableEventHandler.cs - X
Miscellaneous Files | D:\Unity Project\Kalkulator\Library\PackageCache\com.ptc.vuforia.engine@8.1.12\Vuforia\Scripts\DefaultTrackableEventHandler.cs
1  /*=====
2  Copyright (c) 2017 PTC Inc. All Rights Reserved.
```

```
DefaultTrackableEventHandler.cs - X | DefaultTrackableEventHandler.cs
Miscellaneous Files | D:\Unity Project\Kalkulator\Assets\DefaultTrackableEventHandler.cs
1  /*=====
```



14. Kita tidak dapat mengubah isi dari Default Trackable Event Handler, namun kita bisa menambahkan perintah untuk menghantarkan sinyal. Jika kamera bertemu dengan masing-masing image target (1 – 10) maka akan menghantarkan sinyal 1, jika tidak menemui image target 1-10 maka sinyal yang dihantarkan bernilai nol.
15. Untuk itu pada Default Trackable Event Handler yang telah dicopy tadi kita tambahkan beberapa script. Yang pertama adalah variable untuk menampung sinyal

```

public class DefaultTrackableEventHandler : MonoBehaviour, ITrackableEventHandler
{
    #region PROTECTED_MEMBER_VARIABLES

    protected TrackableBehaviour mTrackableBehaviour;
    protected TrackableBehaviour.Status m_PreviousStatus;
    protected TrackableBehaviour.Status m_NewStatus;

    #endregion // PROTECTED_MEMBER_VARIABLES

    #region UNITY_MONOBEHAVIOUR_METHODS

    public static int a, b, c, d, e, f, g, h, i, j;
    
```

16. Selanjutnya pada On tracking Found (pada saat kamera bertemu dengan image target yang sesuai) maka Default Trackable Event Handler akan mengirimkan sinyal 1.

```

protected virtual void OnTrackingFound()
{
    var rendererComponents = GetComponentsInChildren<Renderer>(true);
    var colliderComponents = GetComponentsInChildren<Collider>(true);
    var canvasComponents = GetComponentsInChildren<Canvas>(true);

    // Enable rendering:
    foreach (var component in rendererComponents)
        component.enabled = true;

    // Enable colliders:
    foreach (var component in colliderComponents)
        component.enabled = true;

    // Enable canvas':
    foreach (var component in canvasComponents)
        component.enabled = true;

    if(mTrackableBehaviour.TrackableName == "SATU1")
    {
        a = 1;
    }
    if (mTrackableBehaviour.TrackableName == "Dua")
    {
        b = 1;
    }
    if (mTrackableBehaviour.TrackableName == "tiga")
    {
        c = 1;
    }
    if (mTrackableBehaviour.TrackableName == "empat")
    {
        d = 1;
    }
}

```

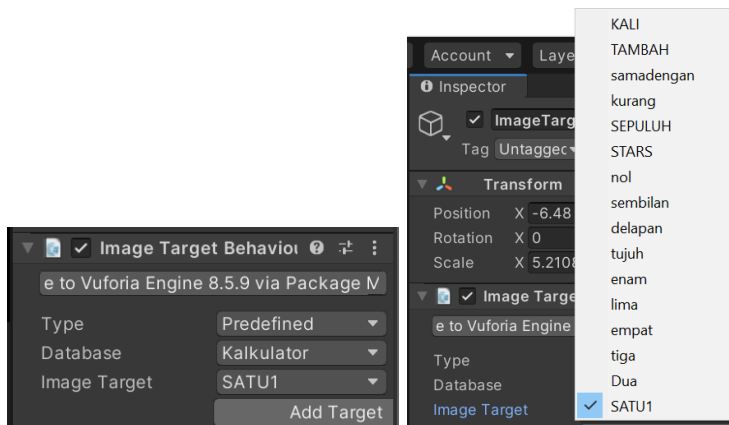


```

if (mTrackableBehaviour.TrackableName == "lima")
{
    e = 1;
}
if (mTrackableBehaviour.TrackableName == "enam")
{
    f = 1;
}
if (mTrackableBehaviour.TrackableName == "tujuh")
{
    g = 1;
}
if (mTrackableBehaviour.TrackableName == "delapan")
{
    h = 1;
}
if (mTrackableBehaviour.TrackableName == "sembilan")
{
    i = 1;
}
if (mTrackableBehaviour.TrackableName == "SEPULUH")
{
    j = 1;
}

```

17. Perlu diperhatikan, TrackableName harus sama dengan nama image target. Anda bisa mengecek nama image target pada bagian Image Target Behaviour



18. Selanjutnya pada On Tracking Lost (pada saat kamera tidak menemui Image Target) maka Default Trackable Event Handler akan mengirim sinyal nol

```

protected virtual void OnTrackingLost()
{
    var rendererComponents = GetComponentsInChildren<Renderer>(true);
    var colliderComponents = GetComponentsInChildren<Collider>(true);
    var canvasComponents = GetComponentsInChildren<Canvas>(true);

    // Disable rendering:
    foreach (var component in rendererComponents)
        component.enabled = false;

    // Disable colliders:
    foreach (var component in colliderComponents)
        component.enabled = false;

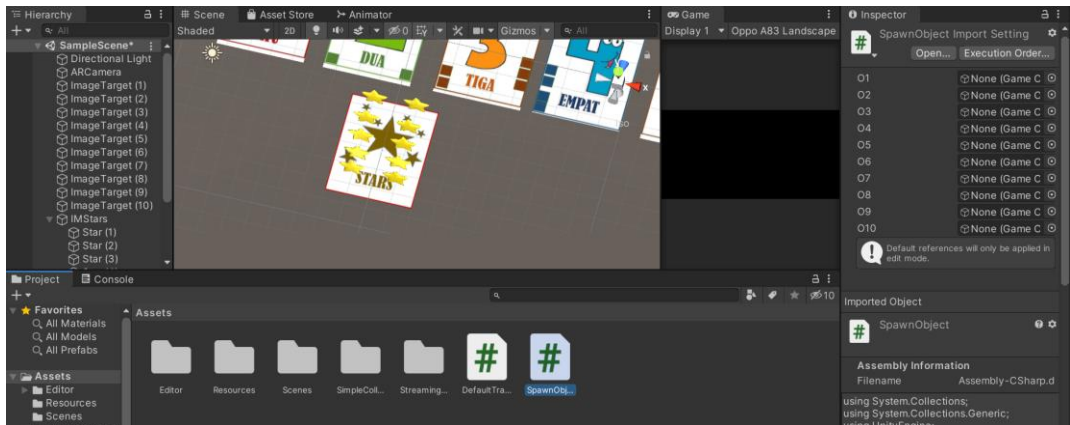
    // Disable canvas':
    foreach (var component in canvasComponents)
        component.enabled = false;

    if (mTrackableBehaviour.TrackableName == "SATU1")
    {
        a = 0;
    }
    if (mTrackableBehaviour.TrackableName == "Dua")
    {
        b = 0;
    }
    if (mTrackableBehaviour.TrackableName == "tiga")
    {
        c = 0;
    }

    if (mTrackableBehaviour.TrackableName == "empat")
    {
        d = 0;
    }
    if (mTrackableBehaviour.TrackableName == "lima")
    {
        e = 0;
    }
    if (mTrackableBehaviour.TrackableName == "enam")
    {
        f = 0;
    }
    if (mTrackableBehaviour.TrackableName == "tujuh")
    {
        g = 0;
    }
    if (mTrackableBehaviour.TrackableName == "delapan")
    {
        h = 0;
    }
    if (mTrackableBehaviour.TrackableName == "sembilan")
    {
        i = 0;
    }
    if (mTrackableBehaviour.TrackableName == "SEPULUH")
    {
        j = 0;
    }
}

```

19. Berikutnya untuk menampilkan object star yang sesuai dengan sinyal yang diberikan oleh Script Default Trackable Event Handler, perlu dibuat sebuah Script baru. Klik kanan pada assets – Create – C# Script – beri nama SpawnObject.



20. Tuliskan Script berikut. Pada void Start semua object di deactivated terlebih dahulu

```
SpawnObject.cs  DefaultTrackableEventHandler.cs
Miscellaneous Files
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class SpawnObject : MonoBehaviour
6  {
7      public GameObject o1, o2, o3, o4, o5, o6, o7, o8, o9, o10;
8      // Start is called before the first frame update
9      void Start()
10     {
11         o1.SetActive(false);
12         o2.SetActive(false);
13         o3.SetActive(false);
14         o4.SetActive(false);
15         o5.SetActive(false);
16         o6.SetActive(false);
17         o7.SetActive(false);
18         o8.SetActive(false);
19         o9.SetActive(false);
20         o10.SetActive(false);
21     }
22 }
```

21. Kemudian pada void update game object di activated bergantung dari signal dari Default Trackable Event Handler.

```

22
23 // Update is called once per frame
24 void Update()
25 {
26     if(DefaultTrackableEventHandler.a == 1)
27     {
28         o1.SetActive(true);
29         o2.SetActive(false);
30         o3.SetActive(false);
31         o4.SetActive(false);
32         o5.SetActive(false);
33         o6.SetActive(false);
34         o7.SetActive(false);
35         o8.SetActive(false);
36         o9.SetActive(false);
37         o10.SetActive(false);
38     }
39     if (DefaultTrackableEventHandler.b == 1)
40     {
41         o1.SetActive(true);
42         o2.SetActive(true);
43         o3.SetActive(false);
44         o4.SetActive(false);
45         o5.SetActive(false);
46         o6.SetActive(false);
47         o7.SetActive(false);
48         o8.SetActive(false);
49         o9.SetActive(false);
50         o10.SetActive(false);
51     }

```

22. Pada void Update dituliskan semua kemungkinan sinyal yang diperoleh dari Default Trackable Event Handler.
23. Untuk membuat Stars menghilang jika kamera tidak menemui image target angka, tambahkan script berikut di void update

```

if (DefaultTrackableEventHandler.a == 0 &&
DefaultTrackableEventHandler.b == 0 &&
DefaultTrackableEventHandler.c == 0 &&
DefaultTrackableEventHandler.d == 0 &&
DefaultTrackableEventHandler.e == 0 &&
DefaultTrackableEventHandler.f == 0 &&
DefaultTrackableEventHandler.g == 0 &&
DefaultTrackableEventHandler.h == 0 &&
DefaultTrackableEventHandler.i == 0 &&
DefaultTrackableEventHandler.j == 0)

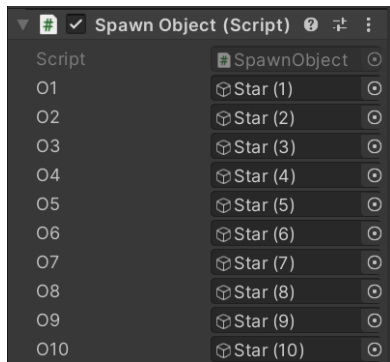
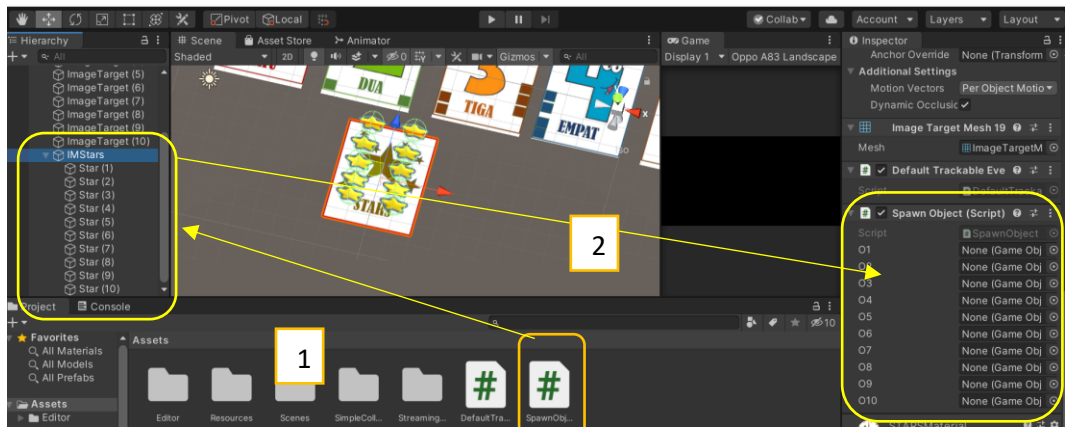
```

```

    {
        o1.SetActive(false);
        o2.SetActive(false);
        o3.SetActive(false);
        o4.SetActive(false);
        o5.SetActive(false);
        o6.SetActive(false);
        o7.SetActive(false);
        o8.SetActive(false);
        o9.SetActive(false);
        o10.SetActive(false);
    }
}

143 if (DefaultTrackableEventHandler.j == 1)
144 {
145     o1.SetActive(true);
146     o2.SetActive(true);
147     o3.SetActive(true);
148     o4.SetActive(true);
149     o5.SetActive(true);
150     o6.SetActive(true);
151     o7.SetActive(true);
152     o8.SetActive(true);
153     o9.SetActive(true);
154     o10.SetActive(true);
155 }
156 if (DefaultTrackableEventHandler.a == 0 && DefaultTrackableEventHandler.b == 0 &&
157 {
158     o1.SetActive(false);
159     o2.SetActive(false);
160     o3.SetActive(false);
161     o4.SetActive(false);
162     o5.SetActive(false);
163     o6.SetActive(false);
164     o7.SetActive(false);
165     o8.SetActive(false);
166     o9.SetActive(false);
167     o10.SetActive(false);
168 }
169 }
170 }
```

24. Drag SpawnScript ke IMStars. Kemudian drag setiap star ke Spawn Object Script pada Inspector



25. Klik Play untuk mencoba

DAFTAR PUSTAKA

- Arifitama, B. (2015). *Panduan Mudah Membuat Augmented Reality*. Tangerang: Andi Offset.
- Azuma, R. T. (1997). A Survey of Augmented Reality. *Presence : Teleoperators and. Virtual Environments* 6(4), 355-385.
- Dhiyatmika, I. D., Putra, I. K., & Mandenni, N. M. (2015). Aplikasi Augmented Reality Magic Book Pengenalan Binatang untuk Siswa TK. *LONTAR KOMPUTER VOL. 6, NO. 2, AGUSTUS 2015* , 120-127.
- Roedavan, R. (2015). *Unity Tutorial Game Engine*. Bandung: Informatika.
- Sukanto, R. A., & Shalahuddin, M. (2013). *Rekayasa Perangkat Lunak*. Bandung: Informatika.

PROVINSI JAWA TENGAH
KABUPATEN SEMARANG

NIK : 3322136608870004

Nama : AURORA NUR AINI
Tempat/Tgl Lahir : KAB. KEBUMEN, 26-08-1987
Jenis kelamin : PEREMPUAN Gol. Darah : O
Alamat : DLIWANG
RT/RW : 005/003
Kel/Desa : UNGARAN
Kecamatan : UNGARAN BARAT
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : DOSEN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP



SEMARANG
04-01-2018

Aurora

PROVINSI JAWA TENGAH
KOTA SEMARANG

NIK : 3318087012840001

Nama : DINA PRASETYOWATI
Tempat/Tgl Lahir : JEMBER, 30-12-1984
Jenis kelamin : PEREMPUAN Gol. Darah : -
Alamat : PERUM GRAHA MUTIARA RESIDEN
D-10
RT/RW : 009/009
Kel/Desa : TLOGOMULYO
Kecamatan : PEDURUNGAN
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : DOSEN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP



KOTA SEMARANG
29-02-2016

Dina

PROVINSI JAWA TENGAH
KABUPATEN BATANG

NIK : 3327062250860007

Nama : MUHAMMAD PRAYITO
Tempat/Tgl Lahir : PE MALANG, 25-02-1986
Jenis kelamin : LAKI-LAKI Gol. Darah :-
Alamat : LUWUNG
RT/RW : 004/001
Kel/Desa : LUWUNG
Kecamatan : BANYUPUTIH
Agama : ISLAM
Status Perkawinan: KAWIN
Pekerjaan : DOSEN
Kewarganegaraan: WNI
Berlaku Hingga : SEUMUR HIDUP



BATANG
29-12-2017

Muhammad Prayito

PROVINSI JAWA TENGAH
KABUPATEN BREBES

NIK : 3329141803840001

Nama : ARYO ANDRI NUGROHO
Tempat/Tgl Lahir : BREBES, 18-03-1984
Jenis kelamin : LAKI-LAKI Gol. Darah :-
Alamat : KARANGSARI
RT/RW : 007/001
Kel/Desa : KARANGSARI
Kecamatan : BULAKAMBA
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : DOSEN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP



BREBES
20-04-2016

Aryo Andri Nugroho

PROVINSI JAWA TENGAH
KOTA SEMARANG

NIK : 3307064301870002

Nama : IDA DWIJAYANTI
Tempat/Tgl Lahir : WONOSOBO, 03-01-1987
Jenis kelamin : PEREMPUAN Gol. Darah : B
Alamat : PERUM GASEM WULUNG KAV 02
RT/RW : 004/004
Kel/Desa : TLOGOSARI WETAN
Kecamatan : PEDURUNGAN
Agama : ISLAM
Status Perkawinan : KAWIN
Pekerjaan : DOSEN
Kewarganegaraan : WNI
Berlaku Hingga : SEUMUR HIDUP



KOTA SEMARANG
07-10-2015

[Handwritten signature]