

**LAPORAN PRAKTIKUM  
GAME EDUKASI**

**MODUL 7  
GAME ENGINE UNITY 3D**

**Oleh:  
Zakky Septian N      A710210016**

**Program Studi Pendidikan Teknik Informatika  
Fakultas Keguruan dan Ilmu Pendidikan  
Universitas Muhammadiyah Surakarta  
2024**

Mata Kuliah	: Praktikum Game Edukasi	<b>Ttd</b>
Program Studi	: Pendidikan Teknik Informatika	
Nama	: Zakky Septian N	
NIM	: A710210016	
Submit pertama	:	
Accepted	:	

## MODUL 7 GAME ENGINE UNITY 3D

### A. DASAR TEORI

#### 1. Pengantar

Unity adalah sebuah platform permainan yang dapat digunakan di berbagai sistem operasi, dikembangkan oleh Unity Technologies dan diresmikan serta dirilis pertama kali pada Juni 2005 dalam acara Apple Worldwide Developers Conference sebagai platform permainan untuk sistem operasi Mac OS X. Sejak itu, platform ini telah mengalami perkembangan yang pesat untuk mendukung berbagai platform, termasuk desktop, perangkat seluler, konsol, dan realitas virtual. Khususnya, Unity sangat populer digunakan dalam pengembangan permainan untuk perangkat seluler seperti iOS dan Android, dikenal karena kemudahannya bagi pengembang pemula serta menjadi pilihan utama dalam pengembangan permainan indie.



Gambar 7.1 Logo Unity

Platform ini mampu digunakan untuk menciptakan berbagai jenis permainan dan simulasi interaktif, termasuk permainan tiga dimensi (3D) dan dua dimensi (2D). Selain itu, kemampuan Unity telah menarik minat dari industri lain di luar industri video game, seperti film, otomotif, arsitektur, teknik, konstruksi, dan bahkan Angkatan Bersenjata Amerika Serikat.

Unity menyediakan dukungan untuk berbagai bahasa pemrograman, termasuk C# dan JavaScript, yang memungkinkan pengembang untuk mengkustomisasi permainan mereka sesuai dengan keinginan. Dengan adanya dukungan untuk bahasa-bahasa ini, pengembang dapat dengan mudah menyesuaikan kode permainan, membuat logika permainan, mengimplementasikan fitur khusus, dan banyak lagi. Dengan demikian, Unity

memberikan fleksibilitas kepada pengembang dalam mengembangkan permainan sesuai dengan kebutuhan dan preferensi mereka.

## 2. Fitur Unity

Ada beberapa fitur yang sangat berguna dalam Unity, di antaranya:

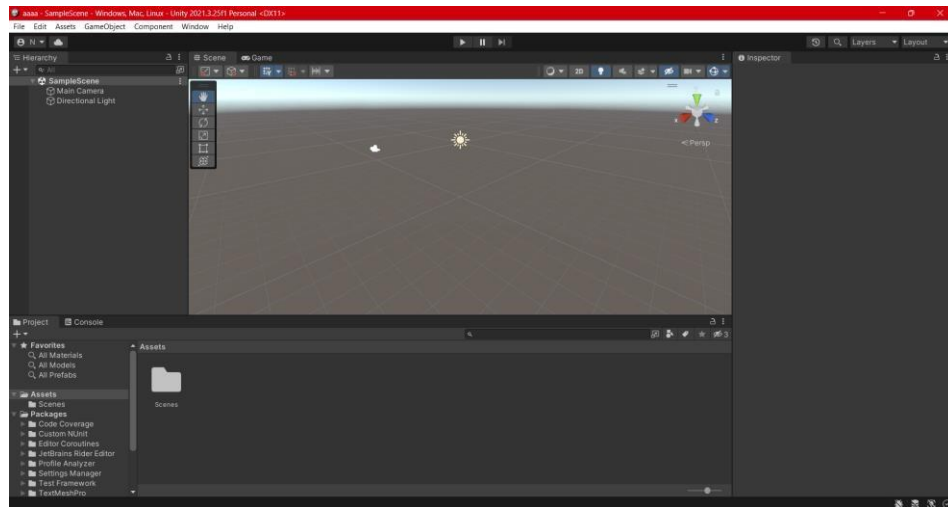
- a. High Definition Render Pipeline (HDRP): Pipa render yang menghasilkan grafis berkualitas tinggi dengan pencahayaan realistis dan efek visual yang lebih canggih.
- b. Universal Render Pipeline: Pipa render yang dapat digunakan untuk membuat grafis berkualitas tinggi pada berbagai platform, termasuk perangkat seluler.
- c. Data-oriented Technology Stack (DOTS): Teknologi yang memungkinkan pemrosesan data yang efisien untuk meningkatkan kinerja permainan.
- d. Prefabs: Fitur yang memungkinkan pengguna untuk membuat objek yang dapat digunakan kembali dan diinstansiasi ulang di berbagai bagian permainan.
- e. Cloud Diagnostics: Layanan yang membantu dalam memantau dan menganalisis performa permainan secara real-time di cloud.
- f. Shared Graph: Framework yang memungkinkan kolaborasi antar pengembang dalam mengembangkan permainan. Unity dan Havok Physics: Integrasi Unity dengan mesin fisika Havok untuk mensimulasikan interaksi fisik yang realistis dalam permainan.
- g. 2D Tools: Fitur-fitur khusus untuk pengembangan permainan 2D, termasuk animasi, fisika, dan pengolahan sprite.

Selain fitur-fitur di atas, Unity juga menyediakan berbagai fitur yang mendukung operasional dan monetisasi, seperti:

- a. Multiplayer game servers: Server untuk mendukung pengembangan permainan multipemain. deltaDNA untuk keperluan personalized engagement: Layanan yang membantu dalam mengatur pengalaman permainan yang dipersonalisasi untuk pemain.
- b. Multiplayer games: Dukungan untuk pengembangan permainan multipemain.
- c. Voice dan chat services: Layanan untuk mendukung komunikasi suara dan chat antar pemain.
- d. Live Ops tools: Alat-alat yang membantu dalam mengelola dan memperbarui permainan secara langsung.
- e. User Acquisition (tercatat sudah terinstal di 1.5 Milyar device): Fitur yang membantu dalam mendapatkan pemain baru untuk permainan. Mobile business: Fitur-fitur yang mendukung pengembangan dan monetisasi permainan seluler.
- f. Unity Ads untuk monetisasi: Layanan iklan dalam permainan yang membantu dalam menghasilkan pendapatan.

## 3. User Interface Unity 3D

Secara umum tampilannya terbagi menjadi 6 window editor, antara lain: Project Window, Scene View, Hierarchy Window, Inspector Window, Toolbar, dan Game View.



Gambar 7.2 User Interface Unity 3D

a. Project Windows

Merupakan tempat menampung seluruh aset dari project yang sedang dibuka. Panel sebelah kiri menunjukkan struktur project secara hirarki. Saat folder dipilih, maka panel sebelah kanannya akan menampilkan isi dari folder tersebut. Kolom pencarian asset juga terdapat di dalamnya sehingga akan memudahkan ketika jumlah asetnya cukup banyak, bahkan pencarian pun dapat dilakukan berdasarkan tipe maupun label asset.

b. Scene View

Merupakan tempat melakukan editing dan manipulasi seluruh game object yang dibuat, mulai dari objek itu sendiri, hingga koordinat posisi masing-masing game object. Terdapat perspective sumbu koordinat x, y dan z yang memungkinkan pengguna untuk melihat dari berbagai sudut pandang yang berbeda.

c. Hierarchy Window

Berisi daftar game object yang dapat dilihat parenting system dari objek tersebut. Sehingga akan terlihat bahwa objek tersebut satu unit (induk) dengan objek yang mana. Untuk melakukan itu dapat melalui drag-and-drop, misalnya saja objek "main camera" akan disatukan dengan suatu objek untuk membuat game FPS (First Person Shooter), tinggal drag "main camera" tersebut dan drop pada objek yang diinginkan.

d. Inspector Window

Project game dengan Unity dibuat dengan macam-macam game object misalnya sound, Mesh, Script dan elemen grafik yang lain seperti cahaya. Melalui Inspector window, seluruh informasi yang terdapat pada suatu game object akan terlihat secara lebih jelas. Seluruh properties yang menempel dalam suatu game object tersebut akan ditampilkan.

e. Toolbar

Terdiri atas 7 kontrol dasar dimana semuanya merujuk pada masing-masing bagian editor yang berbeda, antara lain Transform Tools &

Transform Gizmo Toggles (digunakan untuk Scene View), Tombol Play/Pause/Stop (untuk game view), Cloud Button (membuka Unity Services Window), Account Drop-down, Layers Drop-down, dan Layout Drop-down.

f. Game View.

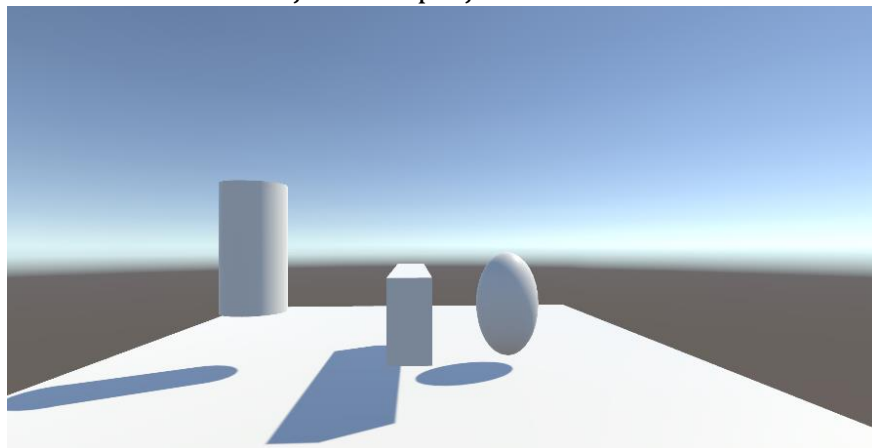
Merupakan tampilan hasil render dari kamera game yang merepresentasikan final game yang akan dipublish. Sehingga akan terlihat tampilan awal game yang sudah dibuat.

## B. TUJUAN PRAKTIKUM

1. Mahasiswa dapat mengenal game engine 3D yang bernama Unity.
2. Mahasiswa dapat melakukan instalasi Unity 3D.
3. Mahasiswa dapat memahami Graphical User Interface dari Unity 3D.

## C. HASIL DAN ANALISIS

1. Kegiatan 1 : Membuat dan menjalankan project baru

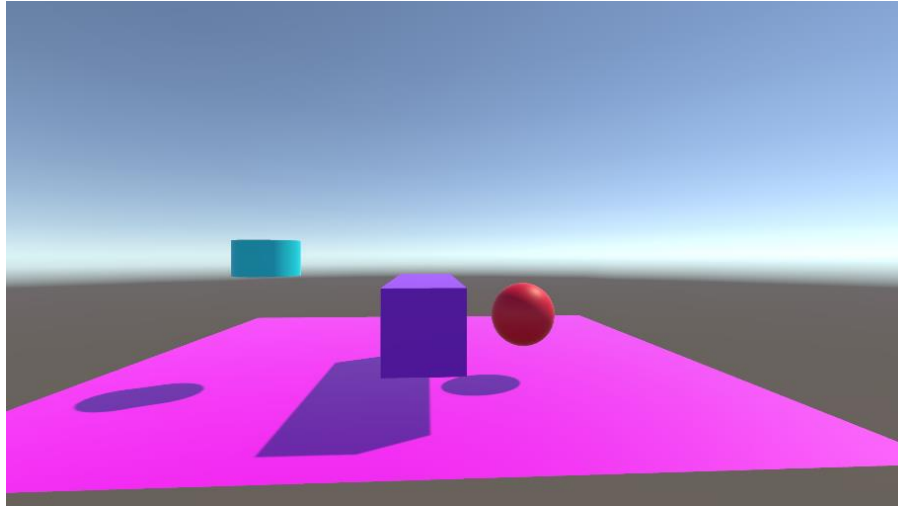


Gambar 7.3 Membuat object

### Analisis:

Dalam tugas pertama ini, kita diarahkan untuk mengkreasi beberapa objek. Objek-objek yang perlu dibuat mencakup Plane, Cube, Capsule, Cylinder, dan Sphere. Kita dapat menambahkannya ke dalam game dengan memilih opsi 'Game Object', kemudian '3D Object', dan setelah itu memilih jenis objek yang ingin ditambahkan, seperti Plane, Cube, Capsule, Cylinder, atau Sphere. Setelah menambahkan objek, kita bisa menjalankan proyek dengan menekan tombol 'play' yang terletak di atas Scene View. Hasilnya dapat dilihat pada Gambar 7.3 yang disediakan.

## 2. Kegiatan 2 : Manipulasi Object Game



Gambar 7.4 Memanipulasi Scale dan Warna object

### Analisis:

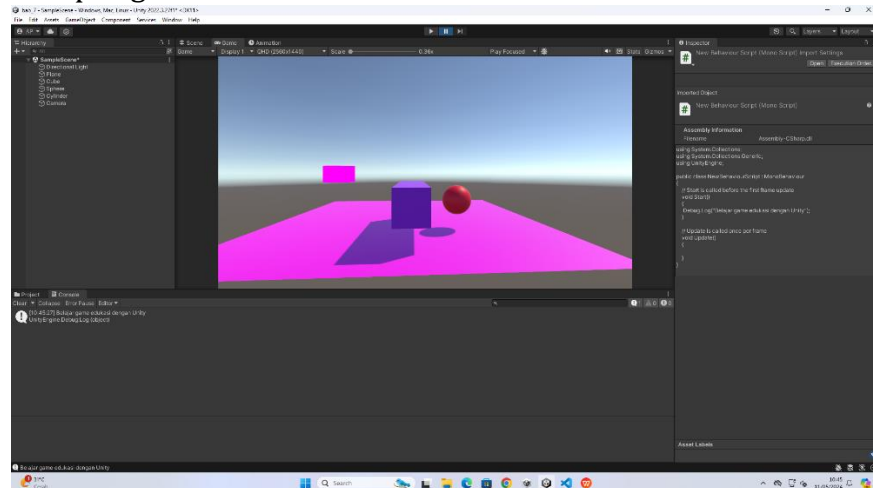
Dalam aktivitas kedua ini, kita akan melakukan manipulasi pada Objek Game yang meliputi dua bagian. Bagian pertama adalah mengubah Skala atau ukuran serta rotasi dari objek, dan bagian kedua adalah mengubah warna dari objek.

Untuk bagian pertama, mengubah skala atau ukuran objek dapat dilakukan dengan dua metode: pertama, menggunakan alat skala yang terletak di Toolbar, dan kedua, menyesuaikan melalui properti objek di jendela Inspector. Proses ini dan hasilnya dapat diperiksa pada Gambar 7.3.

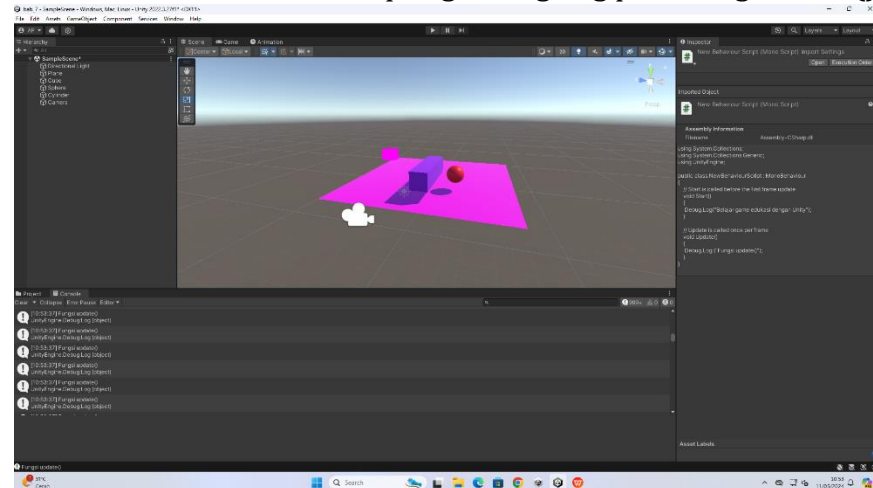
Pada bagian kedua, kita akan mengganti warna objek dengan langkah-langkah berikut:

- Buka menu Assets, kemudian pilih Create, lalu Material, dan beri nama baru seperti "hijau".
- Di jendela Inspector, ubah warna material menjadi hijau atau warna lain sesuai keinginan.
- Terapkan material baru ini ke objek Plane dengan cara memilih Plane, lalu di Mesh Renderer (jendela Inspector) pilih Materials, dan pilih material warna yang telah dibuat. Hasil dari manipulasi warna ini dapat dilihat pada Gambar di atas

### 3. Kegiatan 3 : Scripting Sederhana



Gambar 7.5 Menambahkan Scripting Debug Log pada fungsi Start ()



Gambar 7.6 Menambahkan Scripting Debug Log pada fungsi update ().

#### Analisis:

Dalam sesi ketiga ini, kita akan mempelajari dasar-dasar penulisan skrip dengan dua kegiatan terpisah. Kegiatan pertama adalah memasukkan skrip Debug.Log dalam fungsi Start(), dan kegiatan kedua adalah memasukkan skrip Debug.Log menggunakan fungsi Update().

Untuk kegiatan pertama, kita akan menambahkan skrip Debug.Log ke dalam fungsi Start() seperti berikut:

```
```csharp
void Start() {
    Debug.Log("Belajar game edukasi dengan Unity");
}
```
```

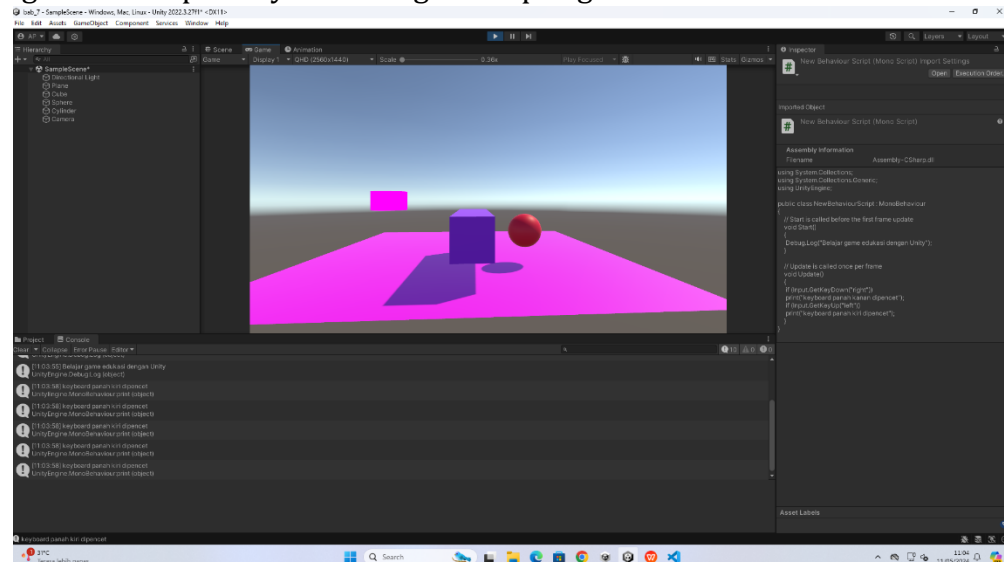
Skrip di atas akan menampilkan pesan Debug Log satu kali di konsol ketika game dimulai. Detailnya bisa dilihat pada Gambar 7.5.

Pada kegiatan kedua, kita akan menambahkan Debug.Log ke dalam fungsi Update() dengan skrip sebagai berikut:

```
``csharp
void Update() {
    Debug.Log("Belajar game edukasi dengan Unity");
}
``
```

Skrip ini akan menampilkan pesan Debug Log berulang kali di konsol selama game berjalan. Untuk detail lebih lanjut, lihat Gambar 7.6.

#### 4. Kegiatan 4 : Input Keyboard dengan Scripting



Gambar 7.7 Menambahkan fungsi Input Keyboard dengan Scripting

Analisis:

Dalam kegiatan keempat ini, kita akan belajar bagaimana menambahkan skrip yang memungkinkan tindakan tertentu muncul di konsol sebagai respons terhadap input keyboard. Kita akan fokus pada input menggunakan tombol panah kanan. Ini dapat dilakukan dengan memasukkan kode berikut ke dalam fungsi Update():

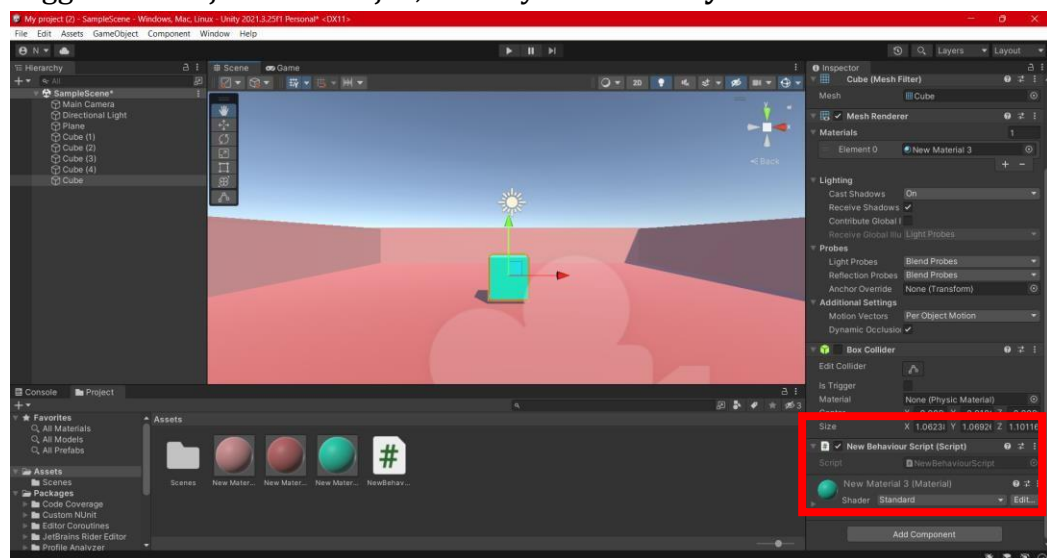
```
``csharp
void Update()
{
    if (Input.GetKeyDown("right")) print("tombol panah kanan telah ditekan");
}
``
```



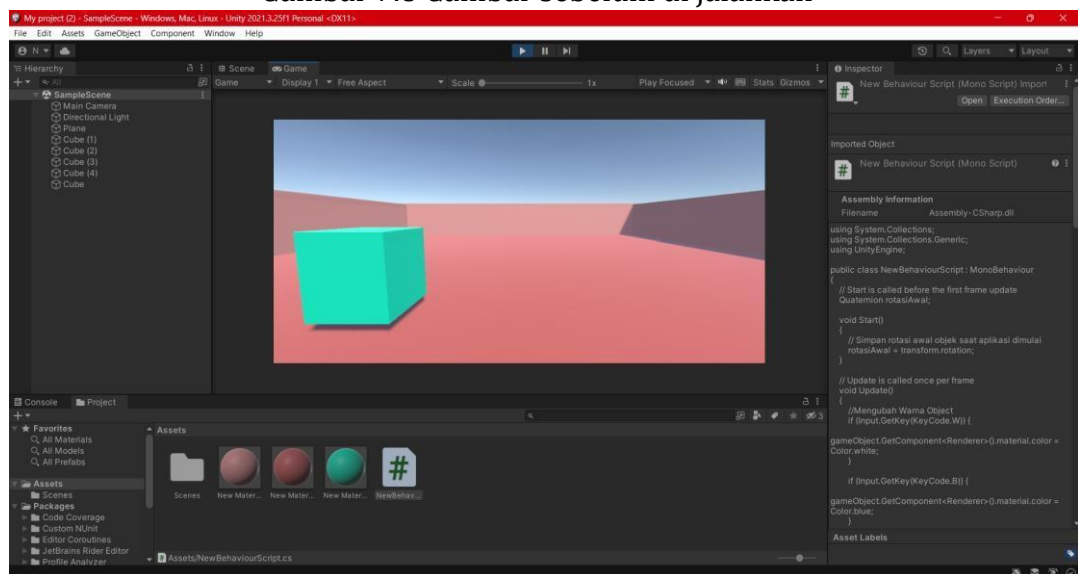
Skrip yang disediakan akan menghasilkan log debug yang menyatakan "tombol panah kanan telah ditekan" setiap kali tombol panah kanan ditekan. Detail lebih lanjut tentang proses ini dapat dilihat pada Gambar 7.7.

#### D. TUGAS

1. Buatlah scene baru, atau menggunakan scene pada kegiatan sebelumnya kemudian buat script baru atau modifikasi script kegiatan diatas untuk menggerakkan objek suatu objek, misalnya Cube. **Penyelesaian:**



Gambar 7.8 Gambar sebelum di jalankan



Gambar 7.9 Gambar setelah di jalankan dan dipindahkan dengan keyboard Analisis:

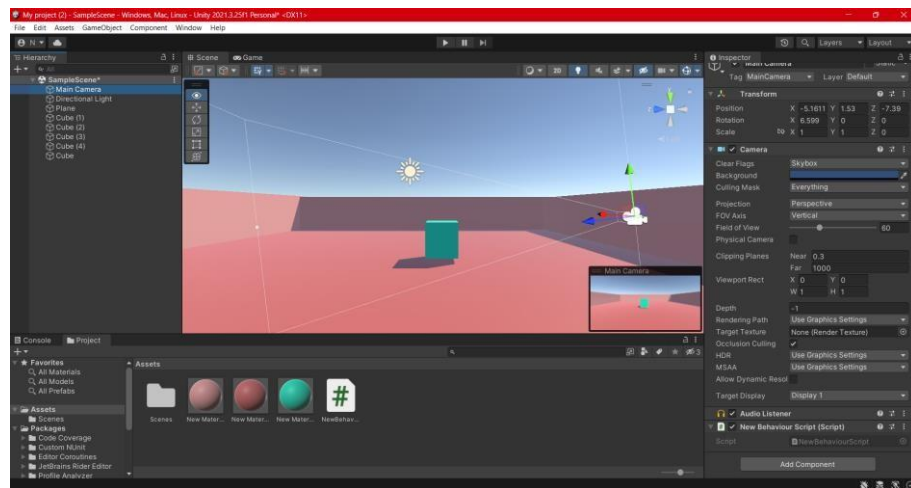
Pada tugas pertama ini, saya membuat sebuah cube yang bisa digerakkan. Disini saya memakai 2 GameObject yaitu cube dan plane. Setelah itu saya menambahkan script dan menuliskan kode berikut pada fungsi Update (): void Update()

```
{
    if (Input.GetKey(KeyCode.UpArrow)) {
transform.Translate(new Vector3(0, 0, 5) * Time.deltaTime);
    }
    if (Input.GetKey(KeyCode.LeftArrow)) {
transform.Translate(new Vector3(-5, 0, 0) * Time.deltaTime);
    }
    if (Input.GetKey(KeyCode.DownArrow)) {
transform.Translate(new Vector3(0, 0, -5) * Time.deltaTime);
    }
    if (Input.GetKey(KeyCode.RightArrow)) {
transform.Translate(new Vector3(5, 0, 0) * Time.deltaTime);
    }

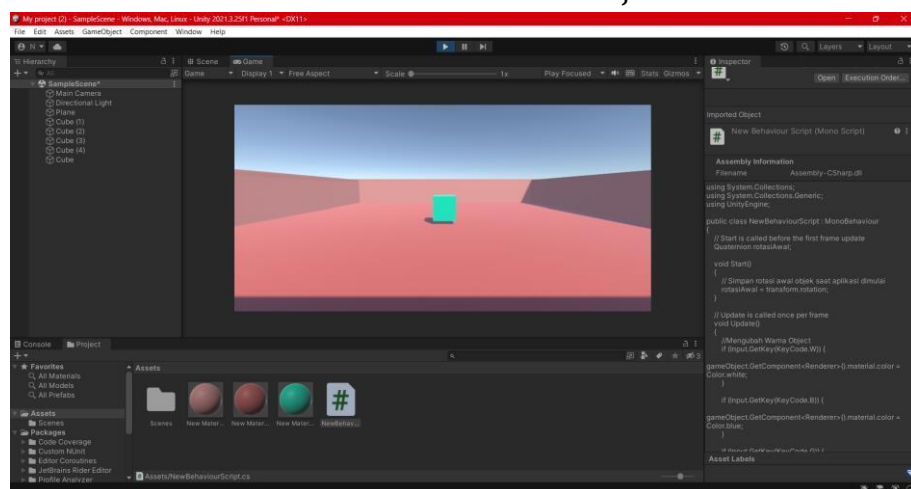
    //Menggeser Naik Turun
    if (Input.GetKey(KeyCode.O)) {
transform.Translate(new Vector3(0, 5, 0) * Time.deltaTime);
    }
    if (Input.GetKey(KeyCode.P)) {
transform.Translate(new Vector3(0, -5, 0) * Time.deltaTime);
    }
}
```

Disini saya memakai fungsi IF (Input.GetKey()) untuk memasukkan inputan perintah, dan disini saya juga memakai perintah transform.Translate(new Vector(X,Y,Z)\*Time.deltaTime) yang digunakan untuk mengubah posisi. Setelah selesai menuliskan script, kemudian kita akan menambahkan script tersebut kedalam game object cube supaya cube tersebut bisa digerakkan. Untuk lebih jelasnya dapat dilihat pada gambar 7.8 dan gambar 7.9 diatas.

2. Lakukan hal yang sama seperti pada tugas 1 tetapi untuk menggerakkan kamera utama menggunakan input keyboard selain yang sudah digunakan pada tugas 1.
- Penyelesaian:**



Gambar 7.10 Gambar sebelum dijalankan



Gambar 7.11 Gambar setelah di jalankan dan dimundurkan dengan keyboard  
**Analisis:**

Pada tugas kedua ini kita diminta untuk melakukan hal yang sama seperti tugas 1, tetapi untuk menggerakkan kamera utama menggunakan input keyboard selain yang sudah digunakan pada tugas 1. Disini saya membuat script baru dengan fungsi yang sama akan tetapi dengan input keyboard yang berbeda yaitu dengan key K dan L. setelah script ditulis, saya memasukkan script tersebut kedalam camera. Untuk hasil dapat dilihat pada gambar 7.10 dan gambar 7.11 diatas.

## E. KESIMPULAN

1. Unity mampu digunakan untuk menciptakan berbagai jenis permainan dan simulasi interaktif, termasuk permainan tiga dimensi (3D) dan dua dimensi (2D).
2. Ada beberapa fitur yang sangat berguna dalam Unity, di antaranya:
  - a. High Definition Render Pipeline (HDRP),
  - b. Universal Render Pipeline,
  - c. Data-oriented Technology Stack (DOTS),
  - d. Prefabs, Cloud Diagnostics,
  - e. Shared Graph,
  - f. 2D Tools.

3. User Interface dari Uity terbagi menjadi 6 window editor, antara lain:
  - a. Project Window,
  - b. Scene View,
  - c. Hierarchy Window,
  - d. Inspector Window,
  - e. Toolbar,
  - f. dan Game View.

#### **F. DAFTAR PUSTAKA**

- Andi Taru Nnw. 2020. Sejarah dan Perkembangan Unity Game Engine Tahun 2020. <https://www.gamelab.id/news/250-sejarah-dan-perkembangan-unitygame-engine-tahun-2020> (Diakses pada tanggal 03 Juni 2024 Pukul 03.05)
- Griyawebste.com. 2022. Penjelasan UI dan Fitur Unity 3D. <https://www.griyawebste.com/penjelasan-ui-dan-fitur-unity-3d/> (Diakses pada tanggal 03 Juni 2024 Pukul 03.06)
- Wikipedia.com . 2023. Unity Technologies. [https://id.wikipedia.org/wiki/Unity\\_Technologies](https://id.wikipedia.org/wiki/Unity_Technologies) . (Diakses Pada tanggal 03 Juni 2024 Pukul 03.07)