PEMROGRAMAN GAME
JOB SHEET MINGGU KE-3

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI SEMARANG
2022/2023

I. Tujuan Instruksional Khusus

Setelah melekukan praktikum ini Mahasiswa mampu memahami dan menerapkan konsep dasar game.

II. Dasar Teori

III. Alat dan Bahan

1. PC / Laptop

IV. Materi

Untuk memahami elemen dasar untuk membuat game, kali ini kita akan mempelajari aspek-aspek minimal pembuatan game TENIS MEJA.

A. Game Pukul Bola

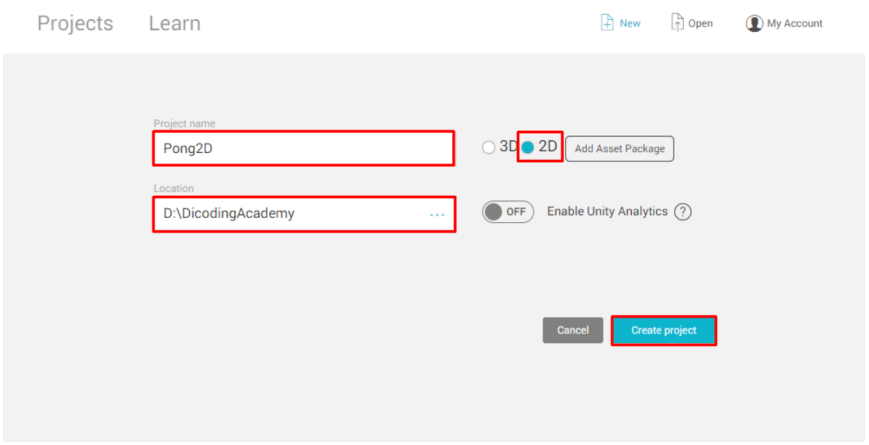
Game Pukul Bola adalah video game seperti tenis meja yang terdiri dari dua pemukul dan satu bola yang dapat dimainkan oleh dua orang pemain atau lebih dengan melawan komputer. Setiap pemain harus bisa mengembalikan setiap bola yang mengarah padanya. Setiap kali lawan gagal menerima bola, kita akan mendapatkan satu poin, dan sebaliknya. Setelah mencapai batas poin tertentu, kita atau lawan akan keluar sebagai pemenangnya.

Di modul ini kamu akan belajar tentang **UI**, **Physics 2D**, **Transform**, **Material Physics**, **Music dan Sound**.

Praktik: Membuat Proyek Baru

---

1. Buka Unity.
2. Klik **New.**
3. Masukkan nama proyek yang diinginkan. Mari kita gunakan nama "**Pong2D."**
4. Centang/pilih **2D.**
5. Pilih tempat di mana proyek tersebut akan disimpan.
6. Untuk tutorial ini, pilih **OFF** di sebelah tulisan "**Enable Unity Analytics.**"
7. Klik tombol "**Create project."**



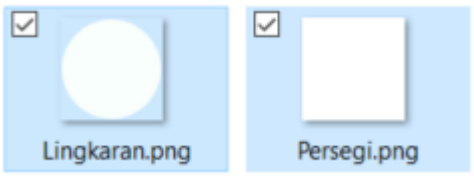
Setelah membuat berkas proyek, selanjutnya Anda akan mempelajari beberapa materi antara lain:

- **Memasukkan Assets.**
- **Membuat Area Game.**
- **Membuat Paddle (Pemukul).**
- **Membuat Bola.**
- **Menambahkan Audio.**
- **Membuat Score.**
- **Membuat Halaman Game Over dan Halaman Menu.**

Memasukkan Aset

Menyiapkan Asset

Kita memerlukan 2 gambar untuk mendukung permainan, yaitu lingkaran dan persegi.



Detailnya sebagai berikut:

Lingkaran

- Nama : Lingkaran.png
- Ukuran : 64x64 pixel
- Warna : Putih
- Keterangan : Lingkaran penuh dengan warna putih

Gambar ini akan digunakan sebagai bola

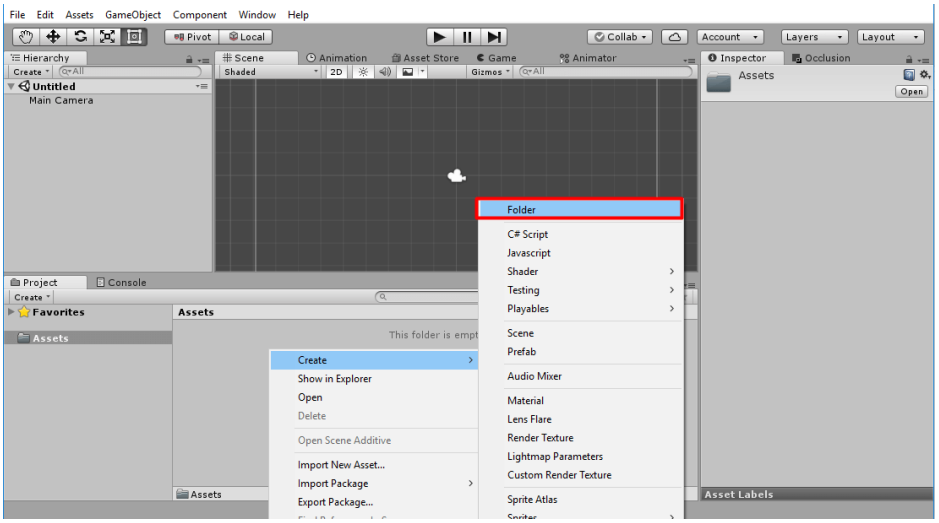
Persegi

- Nama : Persegi.png
- Ukuran : 64x64 pixel
- Warna : Putih
- Keterangan : Blok penuh dengan warna putih

Gambar ini akan digunakan sebagai Paddle dan Wall.

Praktik: Membuat Folder

Supaya penyusunannya rapi kita perlu mengelompokan jenis berkas yang sama dalam satu folder. Caranya dengan buka panel **Project**, kemudian klik kanan pilih **Create** > **Folder**. Beri nama **Images**.

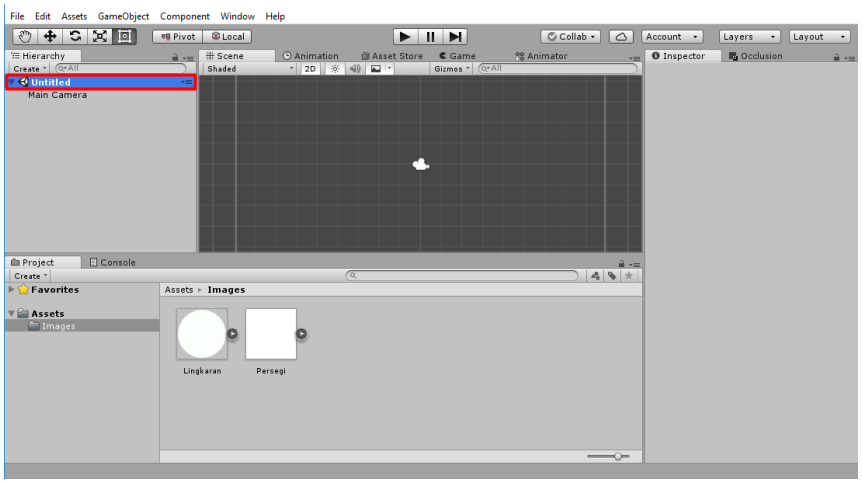


Anda juga dapat mengganti nama folder dengan klik 2 (dua) kali pada nama folder secara perlahan. Tunggu beberapa saat kemudian muncul edit teks pada nama folder. Cara lainnya, Anda dapat klik Folder kemudian tekan F12 pada keyboard.

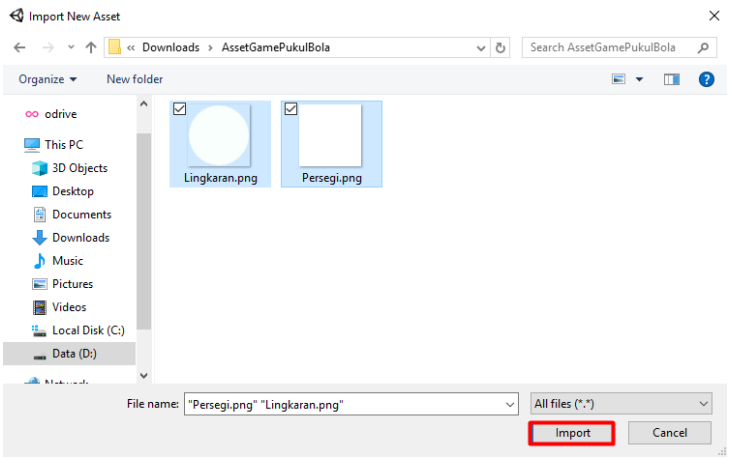
Praktik: Memasukkan Asset ke Proyek

Masukkan gambar yang telah kita buat/unduh sebagai aset proyek. Berikut ini langkah-langkahnya:

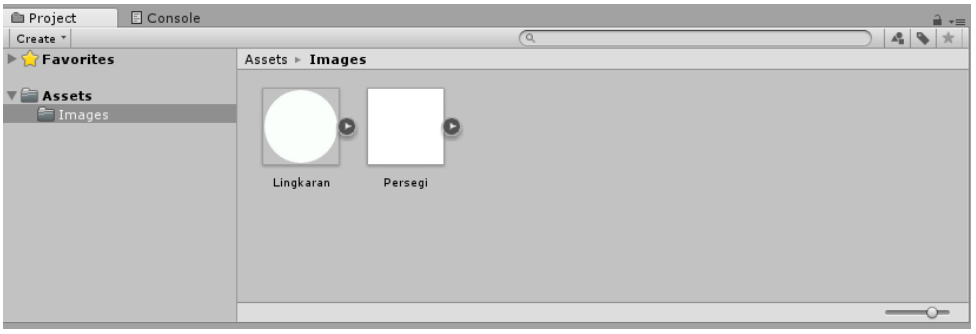
1. Masuk ke dalam folder Images yang telah dibuat dengan klik 2 kali kemudian klik kanan > **Import New Asset.**



2. Kemudian masukkan kedua gambar **Lingkaran** dan **Persegi** (Ctrl+A) dan klik **Import**.



3. Setelahnya muncul kedua gambar seperti di bawah ini



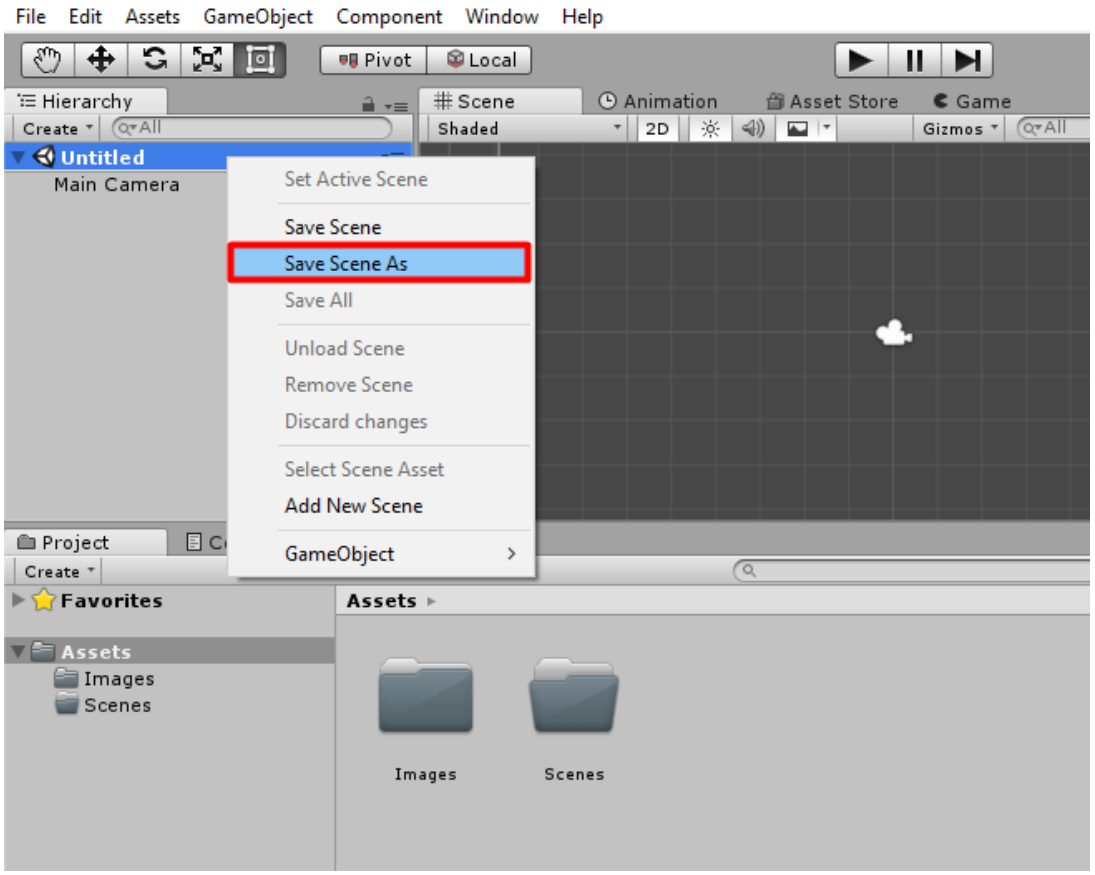
Praktik: Menyimpan Scene

Scene merupakan wadah yang menampung **GameObject** atau semua hal yang kita butuhkan. Satu Scene dapat dipandang sebagai satu level atau satu screen/tampilan dari sebuah objek.

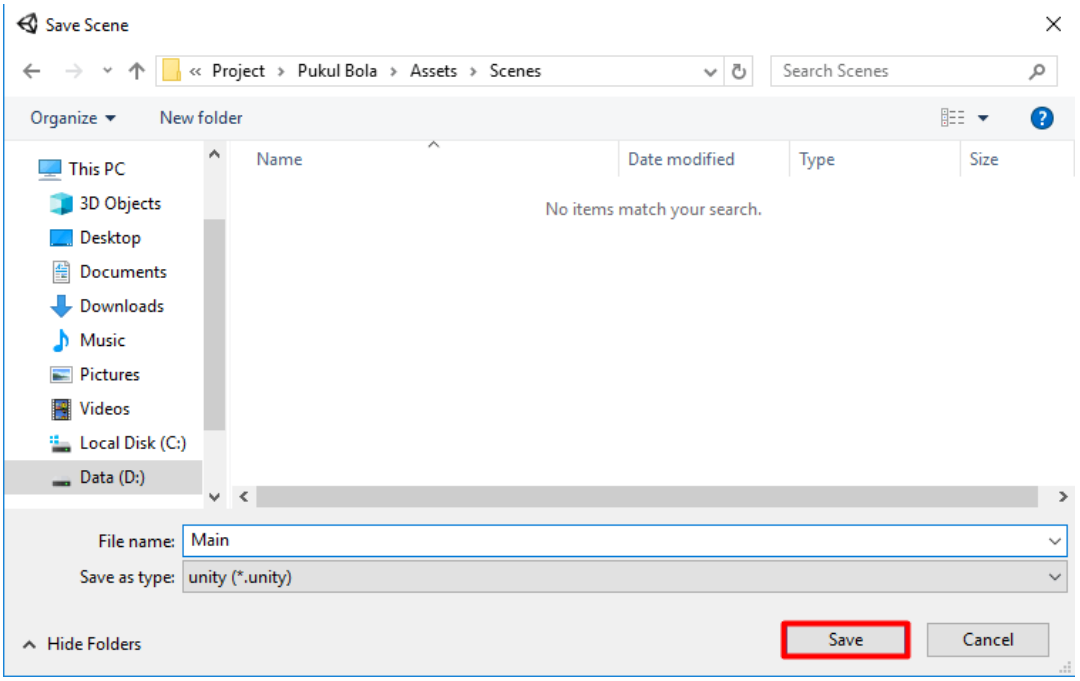
Setelah membuat proyek untuk pertama kali, pada panel Hierarchy terdapat sebuah **Scene (Untitled)** yang memiliki satu **Main Camera**.

Scene yang pertama kali dibuat ini belum tersimpan. Untuk menyimpannya, ikuti langkah-langkah berikut:

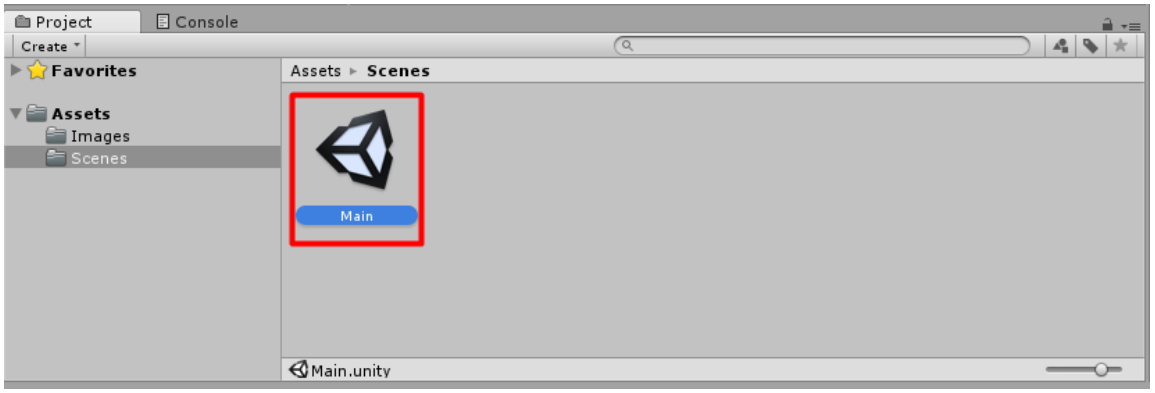
1. Buka Panel Project, kemudian buat folder baru dengan nama **Scenes.**
2. Untuk menyimpan dan memberi nama Scene untuk pertama kalinya, klik kanan pada Scene **Untitled**, lalu pilih **Save Scene As.**



3. Masuk ke folder **Scenes k**emudian beri nama **Main** dan tekan tombol **Save.**



4. Setelah berhasil, pada Folder Scenes muncul berkas baru dengan nama **Main.unity**.

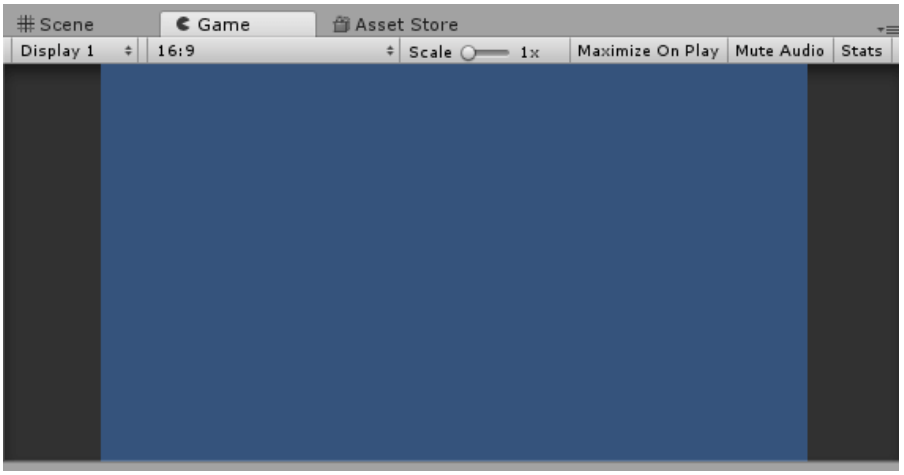


Membuat Background

Praktik: Menentukan Resolusi yang Digunakan

1. Sebelum menyusun sebuah tampilan, baiknya tentukan terlebih dahulu resolusi yang digunakan. Karena target platform adalah Android, maka kita gunakan resolusi 16:9. Buka panel Game kemudian pilih resolusi 16:9.Anda juga dapat menambahkan resolusi tertentu dengan menekan tombol plus (+).

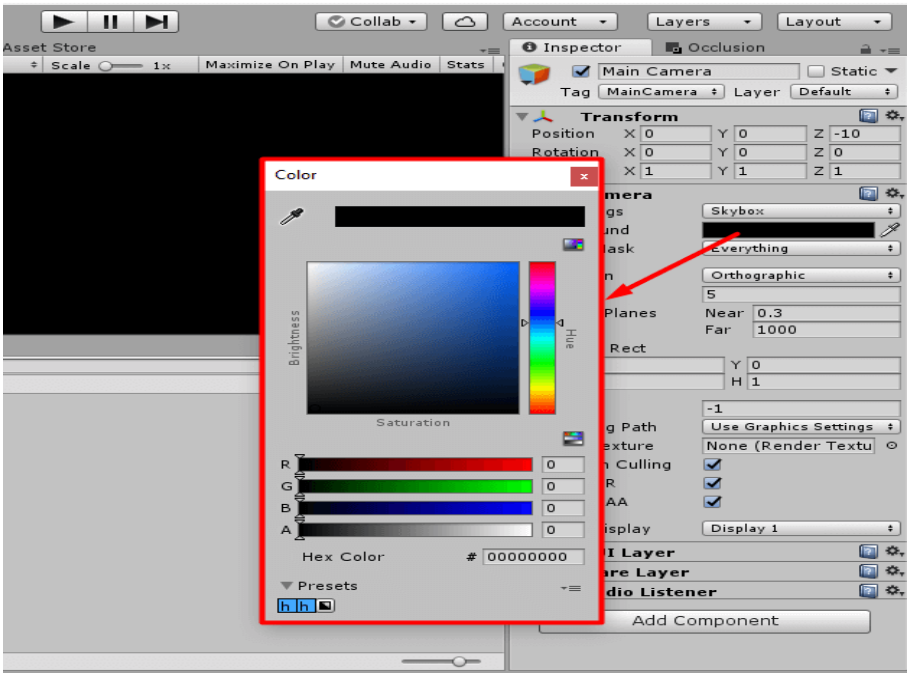
2. Sehingga tampilan menjadi seperti berikut ini:



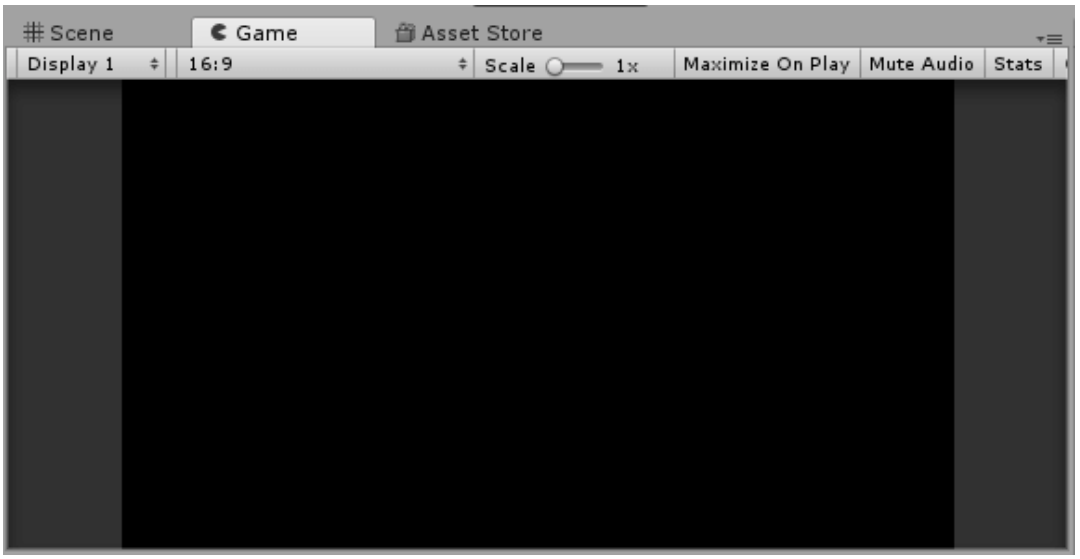
Praktik: Mengubah Warna Background

1. Kamu dapat melihat komponen-komponen dari sebuah GameObject dengan cara klik pada GameObject yang ingin dilihat. Lalu lihat pada **Inspector,** klik dua kali pada Main Camera dan lihat bagian Inspector dari Main Camera.

2. *Default* warna latar belakang adalah biru tua. Untuk mengubah warnanya, klik dua kali pada Background di komponen Camera. Kemudian akan terbuka color picker untuk memilih warna background yang diinginkan. Di pembelajaran ini kita menggunakan warna hitam sebagai background. Sehingga hasilnya seperti berikut ini:

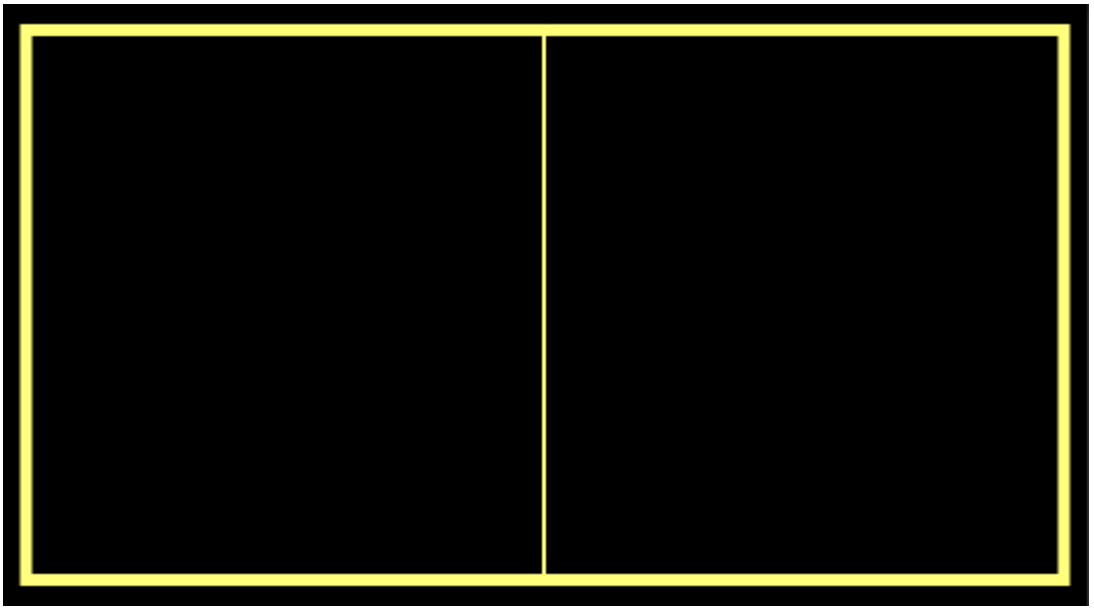


Dan



Membuat Area Game

Pertama mari rumuskan dulu gambar apa saja yang diperlukan dalam **Membuat Area Game Pukul Bola,** antara lain:



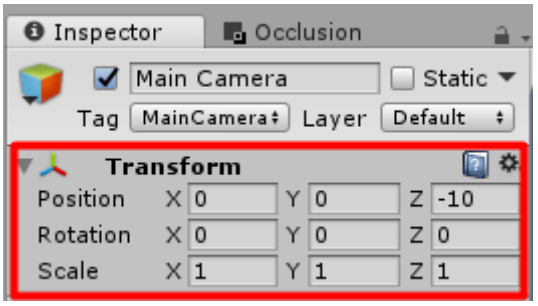
- **Garis tepi atas**
- **Garis tepi bawah**
- **Garis tepi kiri**
- **Garis tepi kanan**
- **Garis batas tengah**

Garis atas, bawah, kiri, kanan memiliki sifat **tidak dapat** ditembus oleh bola sedangkan untuk batas tengah **dapat** ditembus oleh bola.

Untuk semua gambar, kita akan menggunakan GameObject berupa **Sprite.** Sprite adalah sebuah object 2D yang digunakan untuk karakter, peluru, elemen, di permainan 2D. Untuk menampilkan gambar tersebut, dibutuhkan Komponen SpriteRenderer di sebuah GameObject.

Untuk menyusun Object, gunakan toolbox yang telah tersedia.

Dengan tool tersebut, Anda dapat melakukan menggerakkan, memutar atau mengatur besar kecil object. Anda juga dapat mengatur nilai pada Komponen Transform di Panel Inspector.

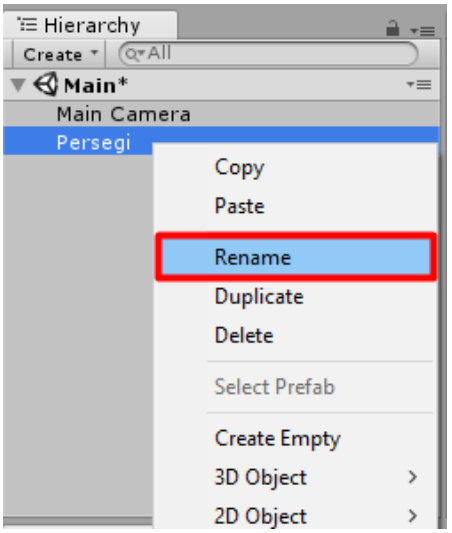


Semua tahapan akan dibahas di bawah ini.

Praktik: Membuat Tepi dan Garis Pembatas Tengah

Selanjutnya kita akan membuat tepi dan garis pembatas tengah sebagai area game. Langkah pertama kita akan membuat tepi atas terlebih dahulu.

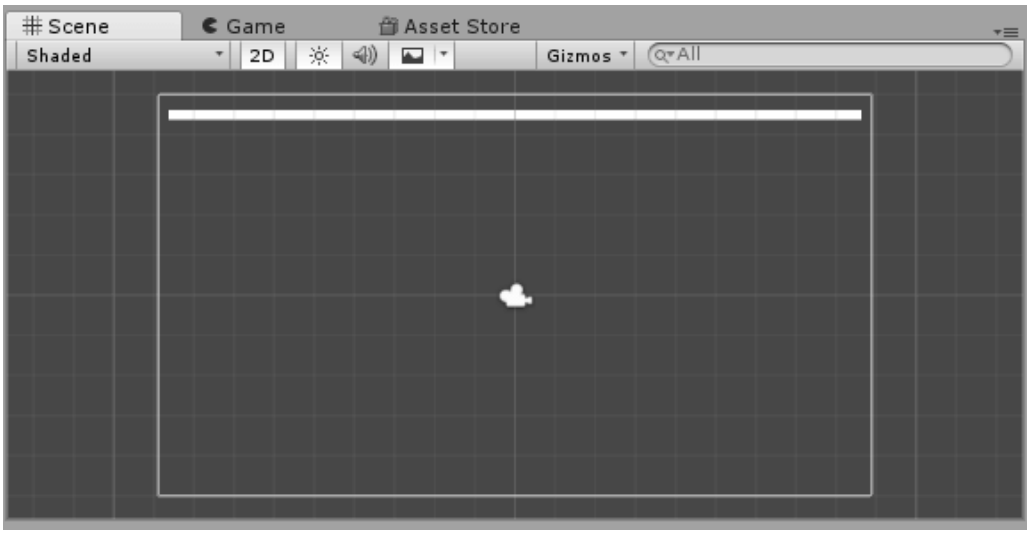
1. Drag gambar **Persegi** dan letakkan di Panel **Scene**.
2. Ubahlah nama Persegi menjadi TepiAtas dengan klik kanan pada GameObject Persegi dan pilih Rename.



3. Aturlah Posisi dan Scala sebagai berikut:

- Posisi X:**0**, Y:**4.5** dan Z:**0**
- Skala X:**26.8**, Y:**0.3** dan Z:**1**

4. Sehingga terlihat sebagai berikut:



5. Agar lebih menarik, Anda dapat mengubah warna pada Sprite Renderer dan pilih warna yang Anda suka, contohnya seperti gambar di bawah ini.
6. Kemudian buatlah hal yang sama seperti **TepiAtas** untuk **Tepi Bawah**, **Tepi-Kanan** , **Tepi Kiri** dan **BatasTengah**. Detailnya sebagai berikut:

    - **TepiBawah**

        - Posisi X:0, Y:-4.5 dan Z:0
        - Skala X:26.8, Y:0.3 dan Z:1

    - **TepiKanan**
        - Posisi X:8.5, Y:0 dan Z:0
        - Skala X:0.3, Y:14.37 dan Z:1

    - **TepiKiri**

        - Posisi X:-8.5, Y:0 dan Z:0
        - Skala X:0.3, Y:14.37 dan Z:1

    - **BatasTengah**

- Posisi X:0, Y:0 dan Z:0
- Skala X:0.1, Y:14.37 dan Z:1

7. Berikut ini cara untuk menambahkan **TepiBawah**:

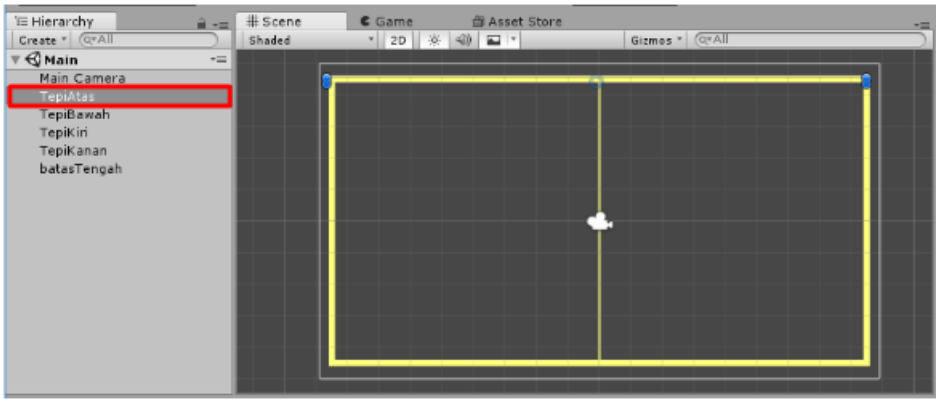
    - Klik kanan pada Tepi Atas
    - Pilih **Duplicate**
    - Rename **TepiAtas (1)** menjadi **TepiBawah**
    - Atur Pos Y menjadi **-4.5**

8. **Selanjutnya, tambahkan tepi kiri, tepi kanan, dan batas tengah dengan cara yang sama.** Atur ukuran dan posisi tepi kiri, tepi kanan, dan batas tengah. Setelah semua tepi dan batas ditambahkan, hasilnya sebagai berikut:

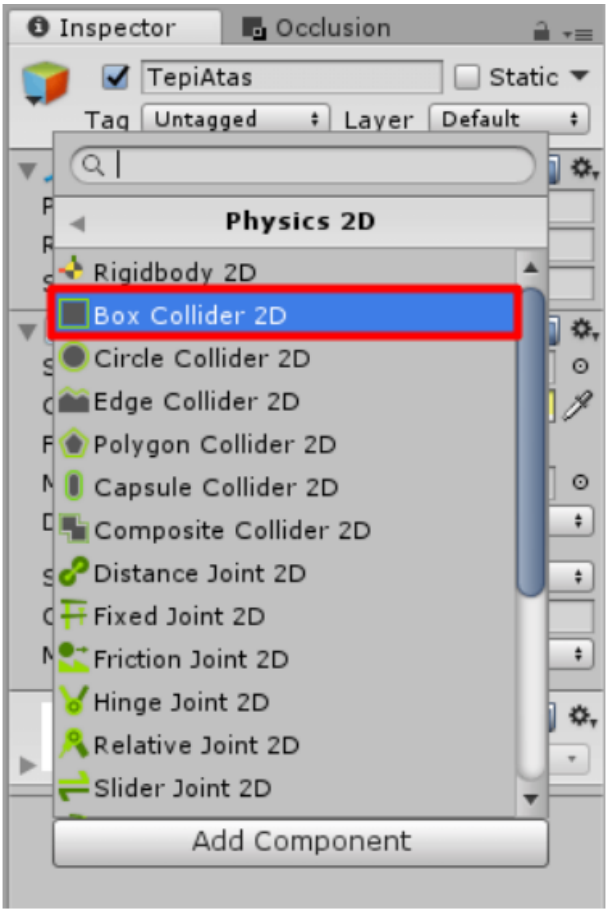
Praktik: Menambahkan Collision pada Tepi

Setelah selesai membuat tampilan Area Game, Anda akan menambahkan Collider sebagai pembatas area agar bola tidak dapat menembus keluar area. Berikut ini caranya:

1. Klik GameObject **TepiAtas** yang terdapat di **Hierarchy.**



2. Pada Panel Inspector, Tambahkan Komponen Collision dengan klik **Add Component** > **Physics 2D** > **Box Collider 2D.**



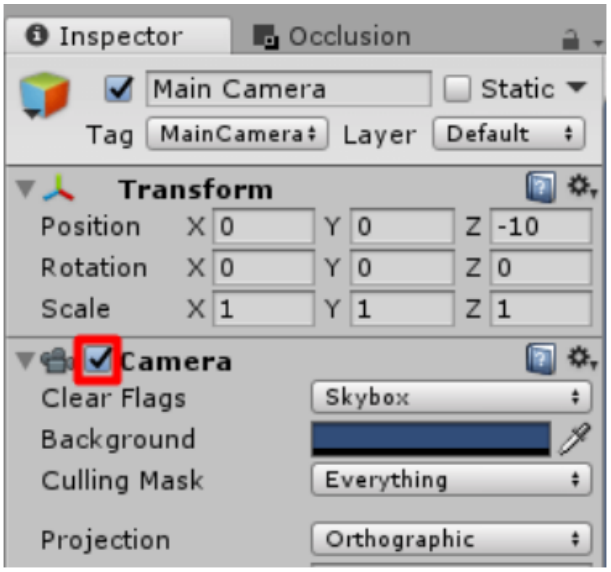
3. Lakukan hal yang sama seperti GameObject **TepiAtas** pada GameObject **TepiBawah**, **TepiKanan** dan **TepiKiri**. Anda dapat menambahkan Collider secara bersamaan dengan blok ketiga gameobject di Hierarchy. Terakhir, tambahkan Component Collider.

Membuat Paddle (Pemukul)

Segala hal yang berada pada Scene atau Hierarchy bertipe **GameObject**. GameObject dapat dipandang sebagai wadah dari berbagai **Components** (komponen). Fungsionalitas dari sebuah GameObject dibentuk dari komponen-komponennya. Secara default, GameObject memiliki sebuah komponen yaitu **Transform** yang memiliki beberapa atribut, yaitu:

- **Position:** menandakan posisi dalam sumbu X, Y, dan Z.
- **Rotation:** menandakan rotasi pada sumbu X, Y, dan Z dalam satuan derajat.
- **Scale:** menandakan skala ukuran pada sumbu X, Y, dan Z. Nilai 1 adalah ukuran asli.

Setiap **Component** memiliki centang di sebelah namanya untuk memungkinkan ia di-enable atau di-disable. **Component** yang di-disable dianggap tidak muncul atau tidak digunakan.

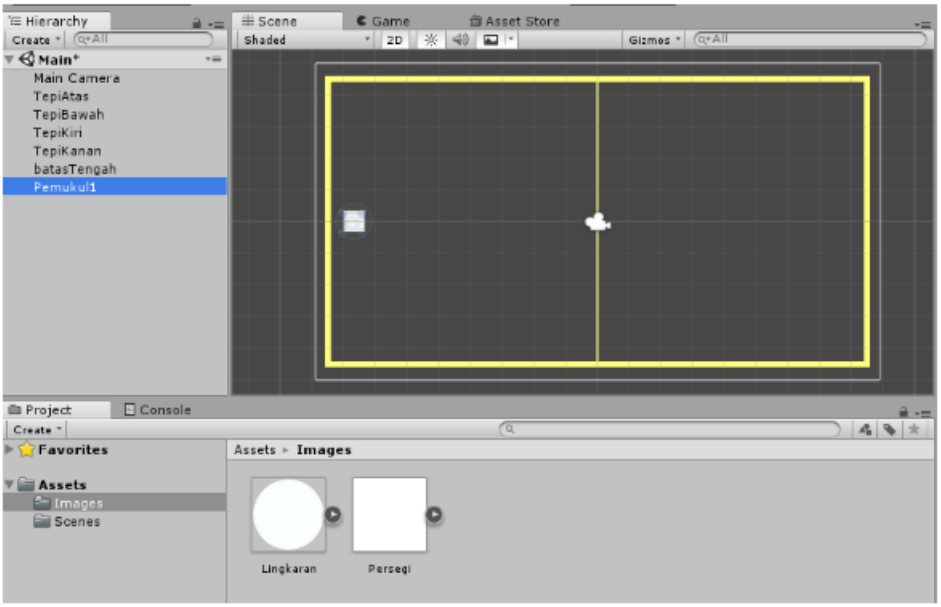


Di submodul ini kita akan membuat Paddle yang bergerak ke atas dan ke bawah dengan menekan tombol keyboard.

Praktik: Membuat GameObject Paddle (Pemukul)

Selanjutnya kita akan membuat Paddle sebagai player. Langkah-langkahnya sebagai berikut:

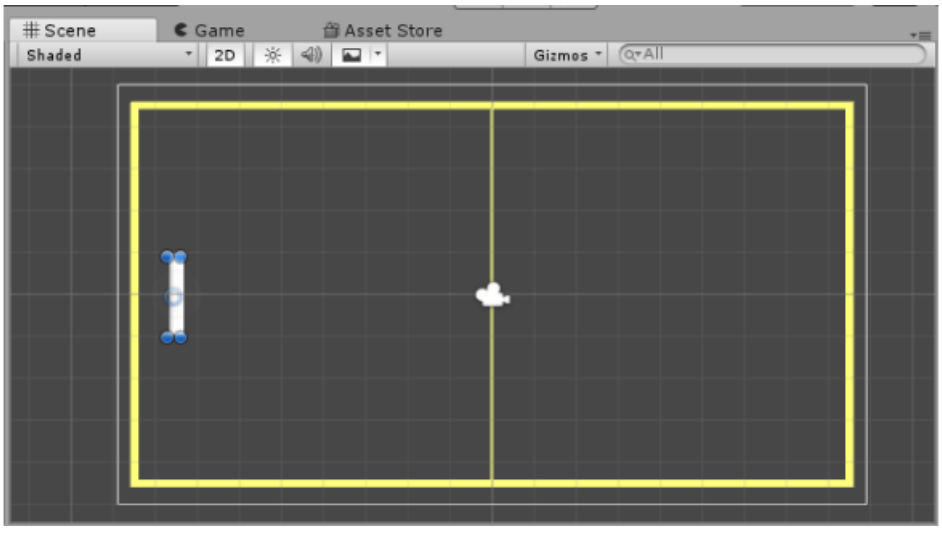
1. Masukkan **Persegi** ke **Scene** kemudian ubah Nama **Persegi** menjadi **Pemukul1.**



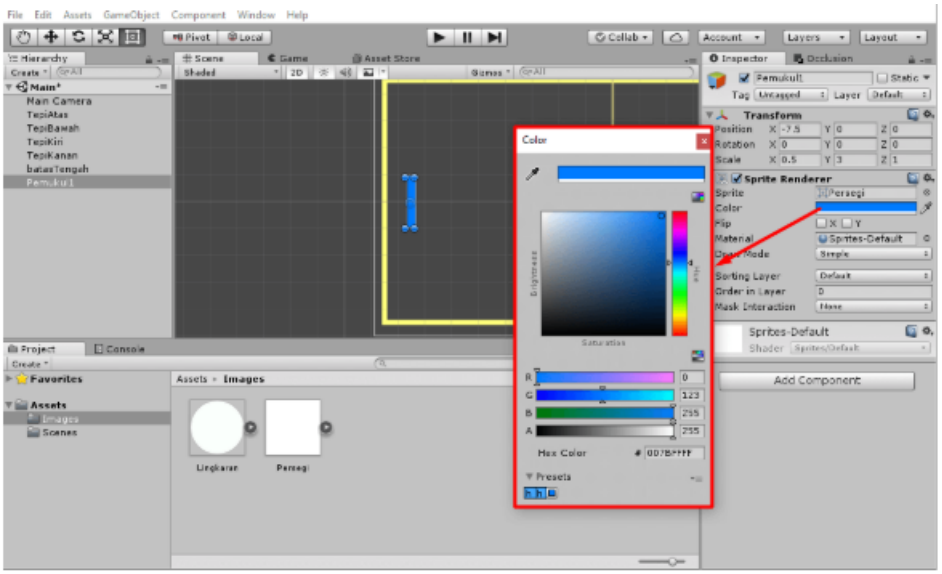
Kemudian Atur posisi dan Skala seperti ini:

- Position: X: **-7.5**, Y:**0** dan Z:**0**
- Scale : X: **0.5**, Y:**5** dan Z:**1**

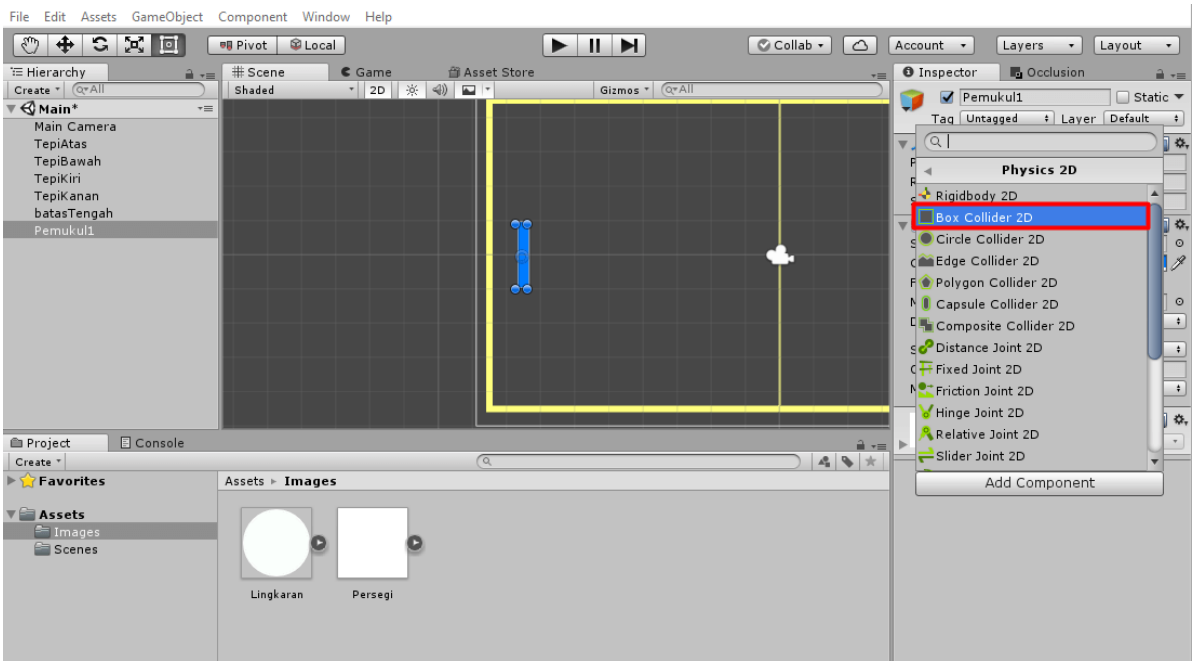
2. Sehingga hasilnya sebagai berikut:



3. Ubah warna sesuai keinginan, misalnya biru.



4. Karena Paddle merupakan objek yang tidak dapat ditembus oleh objek lain maka tambahkan Komponen Collision dengan cara Klik **Add Component** > **Physics 2D** > **Box Collider 2D.**

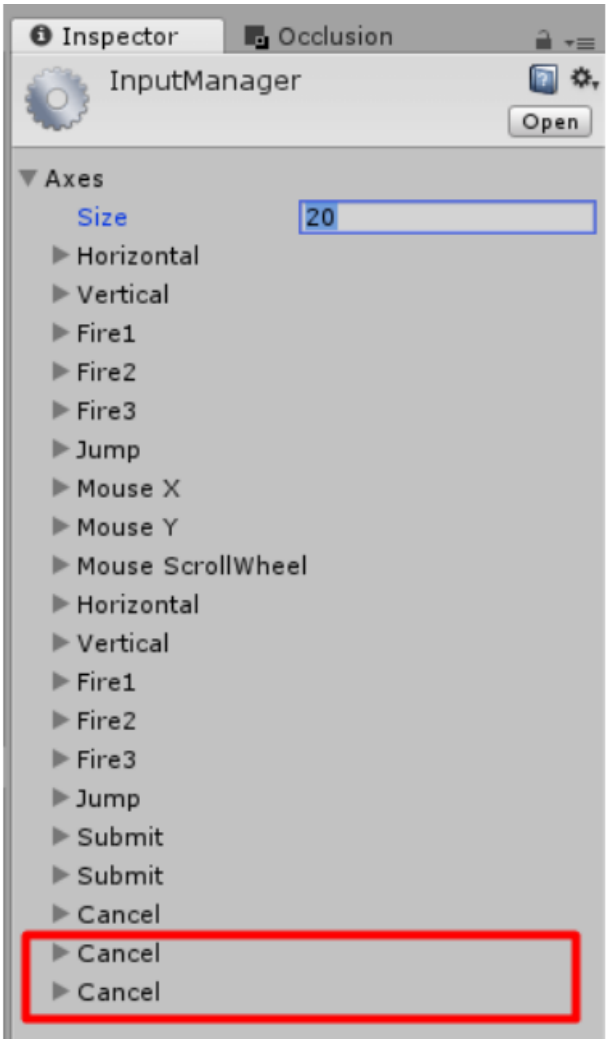


Objek paddle sudah siap. Selanjutnya kita akan membuatnya bergerak.

Praktik: Membuat Pemukul Bergerak

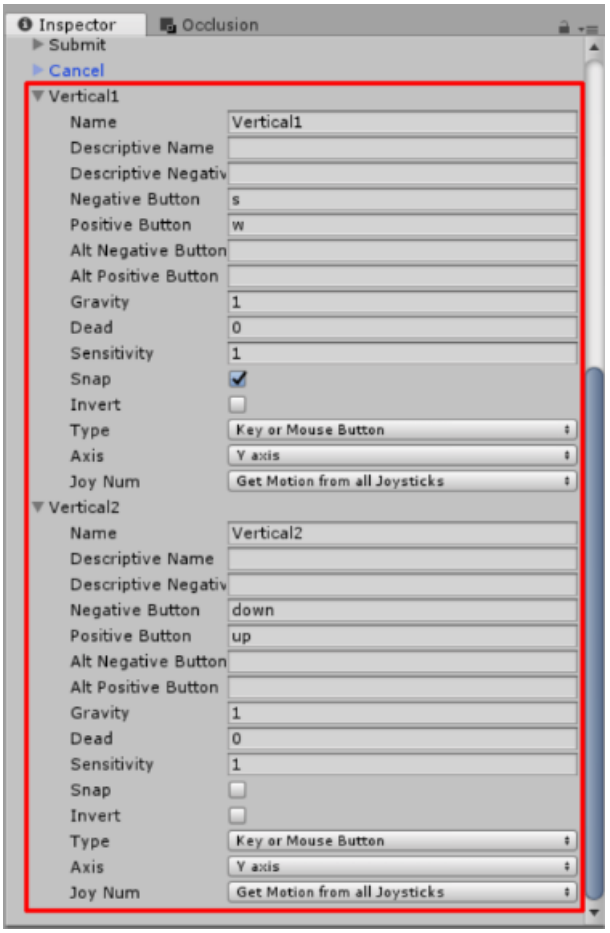
Dalam tutorial ini kedua paddle akan digerakkan dengan menggunakan keyboard. Pemukul pemain pertama (sebelah kiri) digerakkan dengan tombol **W** dan **S** sementara Pemukul pemain kedua (sebelah kanan) digerakkan dengan tombol ↑ dan ↓. Lakukan langkah berikut untuk setting input:

1. Buka **InputManager** (Klik **Edit** > **Project Settings** > **Input).** Karena kita akan menambahkan varian input baru, edit field **Size** pada **Axes** dari **18** menjadi **20.**



Setelah diubah sizenya maka muncul **2 tambahan baru** dengan nama **Cancel**.

2. Untuk dua varian input terbawah, ikuti setting pada screenshot berikut.



Catatan: di versi Unity yang terbaru, jika Anda mengedit yang sudah ada misal Vertical maka akan muncul error.

Dari sini kita memiliki 2 varian customized input.

- **Vertical 1**

  Ini akan digunakan untuk menggerakkan Pemukul dari player 1. Pemukul dari player 1 akan digerakkan ke bawah (negative button) dengan tombol **s** dan bergerak ke atas (positive button) dengan tombol **w**.

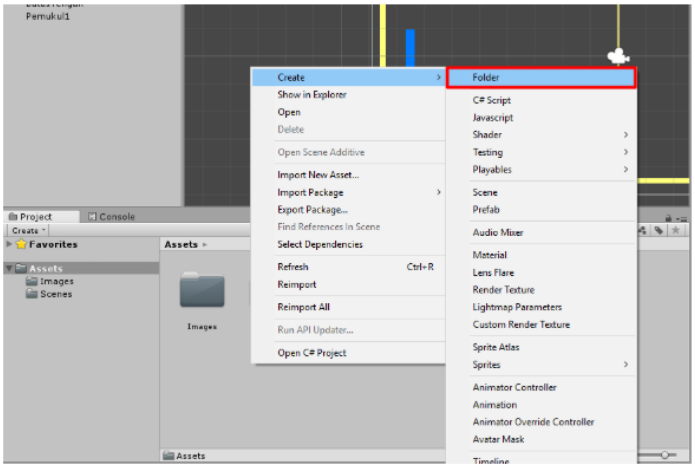
- **Vertical 2**

  Ini akan digunakan untuk menggerakkan Pemukul dari player 2. Pemukul dari player 2 akan bergerak ke bawah (negative button) dengan tombol (down) dan bergerak ke atas (positive button) dengan tombol ↑ (up).

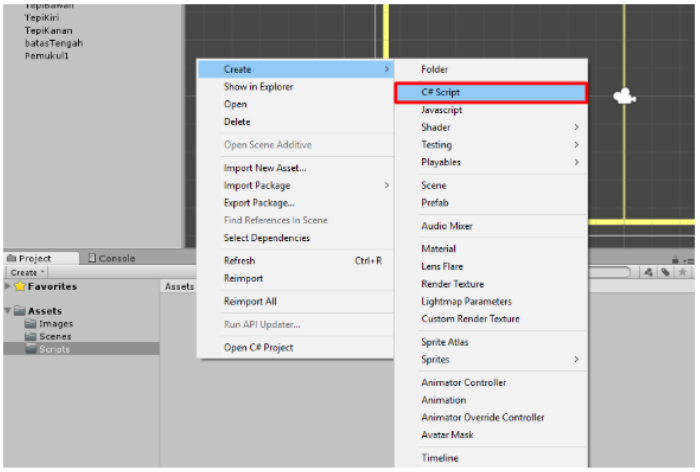
Praktik: Script dan Variable

Selanjutnya kita perlu membuat Script agar Pemukul dapat bergerak sesuai dengan masukan dari keyboard.

1. Pada folder **Assets**, tambahkan folder baru. Klik kanan pada panel Project, kemudian pilih **Create** > **Folder** dan beri nama **Scripts.**



2. Di dalam folder Scripts yang telah dibuat, tambahkan script baru dengan nama **PaddleController.cs**. (**Klik kanan** > **Create** > **C# Scripts**).



3. Pada script **PaddleController.cs**, Klik dua kali pada file " **PaddleController.cs**" untuk mulai menambahkan script. Setiap Script pada Unity merupakan turunan dari kelas **MonoBehaviour** sehingga semua fungsi dan prosedur yang ada pada kelas MonoBehaviour dapat dimanfaatkan pada script.
   Pada game Pong, script yang ditambahkan perlu mengatur:

   - kecepatan pergerakan **Paddle/Pemukul.**
   - varian input mana yang mengatur paddle tersebut: **Vertical 1** atau **Vertical 2.**
   - perubahan posisi pada **sumbu Y** sesuai dengan tombol yang ditekan.

4. **Berikut ini merupakan contoh script yang lengkap.**

   - Tambahkan variabel di dalam class Paddle Controller.

```
1. public float kecepatan;
2. public string axis;
```

   Dalam script di atas, terdapat dua variabel yang bersifat **public**, yaitu kecepatan (kecepatan pergerakan) dan axis (untuk mengatur input keyboard). Variabel yang bersifat public ini akan muncul sebagai field yang bisa diganti isinya tanpa harus mengubah kode.

   - Tambahkan kode di dalam function Update.

```
1. float gerak = Input.GetAxis(axis) * kecepatan * Time.deltaTime;
2. transform.Translate(0, gerak, 0);
```

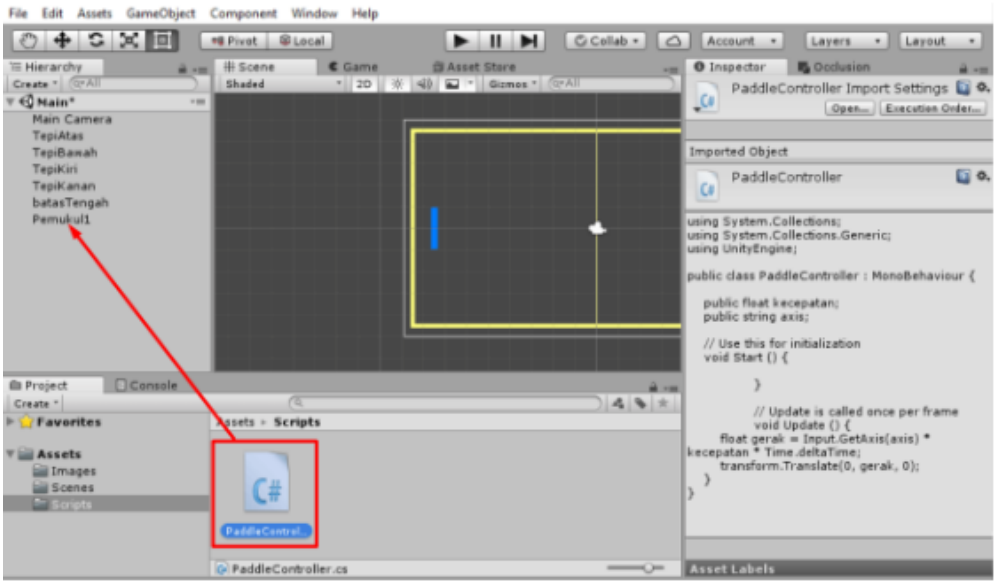
   float gerak = Input.GetAxis(axis) * kecepatan * Time.deltaTime; --> Untuk memperoleh seberapa besar nilai pergerakan paddle dari Input.GetAxis yang memperoleh nilai diantara -1 sampai 1 yang dikalikan kecepatan dan selisih waktu (Time.deltaTime). Tujuannya, supaya dapat berjalan dengan baik di berbagai perangkat.

   transform.Translate(0, gerak, 0); --> Implementasi perpindahan terhadap Paddle.

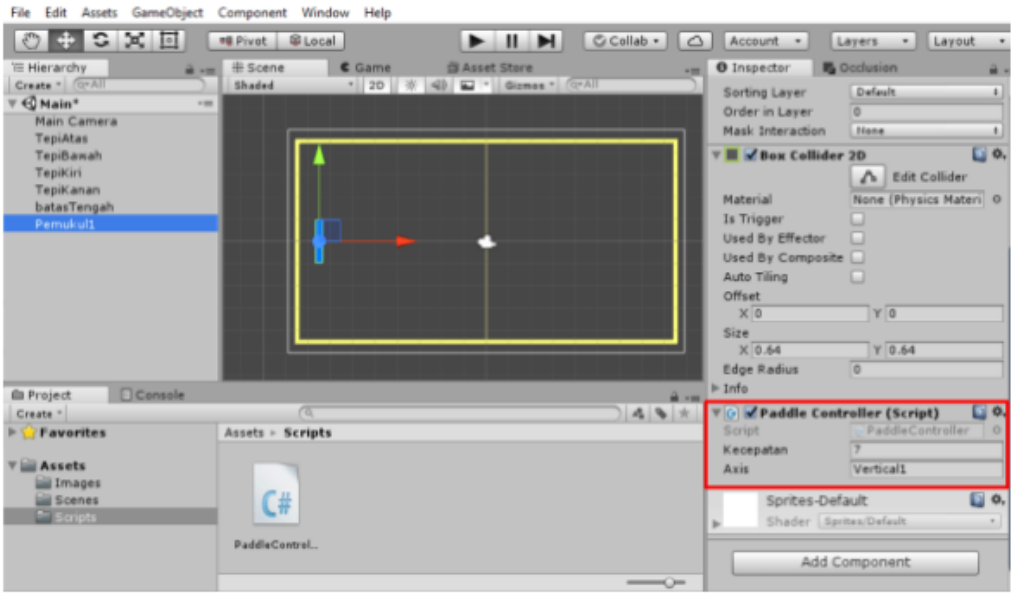
2. Simpan perubahan dengan Ctrl+S, sehingga seluruh kode menjadi seperti berikut:

```
using UnityEngine;
public class PaddleController : MonoBehaviour
{
    public float kecepatan;
    public string axis;
    // Use this for initialization
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        float gerak = Input.GetAxis(axis) * kecepatan * Time.deltaTime;
        transform.Translate(0, gerak, 0);
    }
}
```

2. Masukkan script **PaddleController.cs** yang telah dibuat ke dalam GameObject **Pemukul1**. Anda bisa melakukannya dengan cara drag dan drop.



3. Buka Inspector pada Pemukul1 kemudian pada field **Kecepatan** diisi nilai **7** dan pada **Axis** diisi "**Vertical1.**



"Dalam screenshot di atas, kecepatan diisi dengan 7, sementara Axis diisi dengan **Vertical1**. Hal ini berarti, kecepatan perpindahan adalah 7 dikali hasil pembacaan keyboard. Sementara untuk memindahkan pemukul akan digunakan varian input Vertical 1, yaitu **w** untuk perpindahan ke atas dan **s** untuk perpindahan ke bawah. Pada saat di-play, dengan menekan tombol w dan s, pemukul player 1 akan bergerak.

Praktik: Memberi Batasan Atas dan Bawah

Dalam pemrograman seringkali kita perlu mengatur sesuatu berdasarkan kondisi tertentu. Misalnya dalam kasus Pemukul, perpindahan posisinya perlu dibedakan berdasarkan state dari Pemukul tersebut. Berikut ini logika kondisi yang digunakan dalam pengaturan posisi Pemukul.

- Apabila nilai yang diinput melebihi batas atas maka nilai inputan menjadi 0 berarti GameObject pemukul tidak bergerak.
- Apabila nilai yang diinput kurang dari batas bawah maka nilai inputan menjadi 0 berarti GameObject pemukul tidak bergerak.
- Apabila pemukul berada dalam batas bawah dan batas atas maka pemukul bebas untuk berpindah ke mana pun.

Untuk menerapkannya,ikuti langkah-langkah ini:

1. Buka script **PaddleController.cs** dengan klik 2 kali kemudian tambahkan kode di bawah ini.

   - Tambahkan kode di dalam class **PaddleController.**

```
public float batasAtas;
public float batasBawah;
```

   - Tambahkan kode pengecekan di dalam function **Update().**

```
float nextPos = transform.position.y + gerak;
if (nextPos > batasAtas)
{
    gerak = 0;
}
if (nextPos < batasBawah)
{
    gerak = 0;
}
```

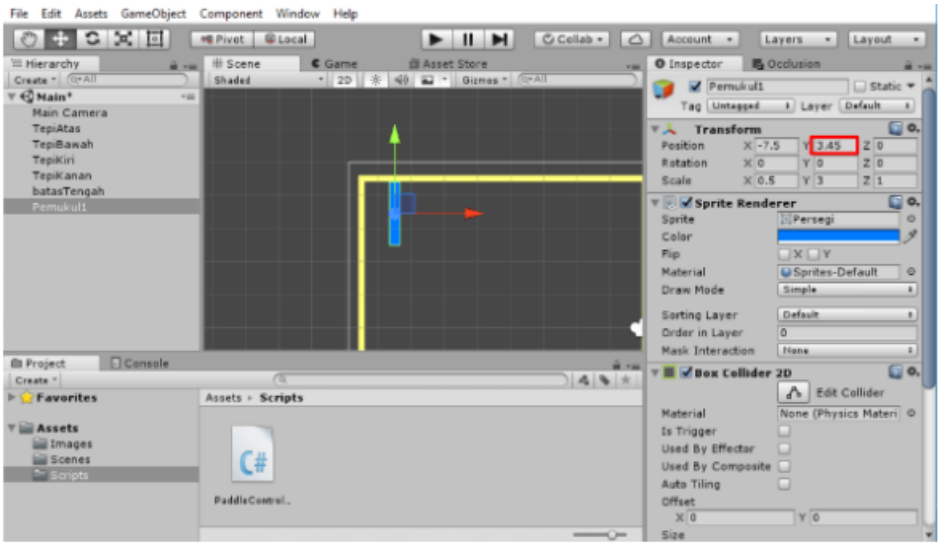
2. Simpan perubahan dengan tekan **Ctrl+S** sehingga seluruh script menjadi seperti ini:

```
using UnityEngine;
public class PaddleController : MonoBehaviour
{
    public float batasAtas;
    public float batasBawah;
    public float kecepatan;
    public string axis;
    // Use this for initialization
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        float gerak = Input.GetAxis(axis) * kecepatan * Time.deltaTime;
        float nextPos = transform.position.y + gerak;
        if (nextPos > batasAtas)
        {
            gerak = 0;
        }
        if (nextPos < batasBawah)
        {
            gerak = 0;
```

```
        }
        transform.Translate(0, gerak, 0);
    }
}
```

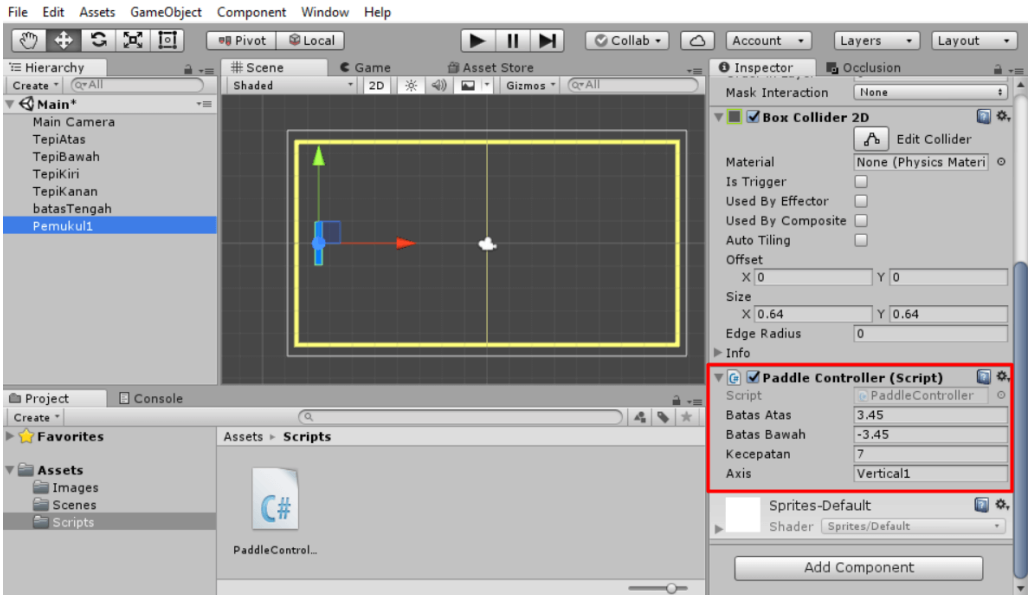
Memprediksi nilai posisi posisi selanjutnya apakah melewati batas atas atau batas bawah. Jika melewati maka paddle tidak ada pergerakan selanjutnya.

2. Untuk mengetahui batas atas dan batas bawah, Anda dapat menggeser paddle ke posisi paling atas. Anda dapat melihat nilai Y pada komponen Transform.

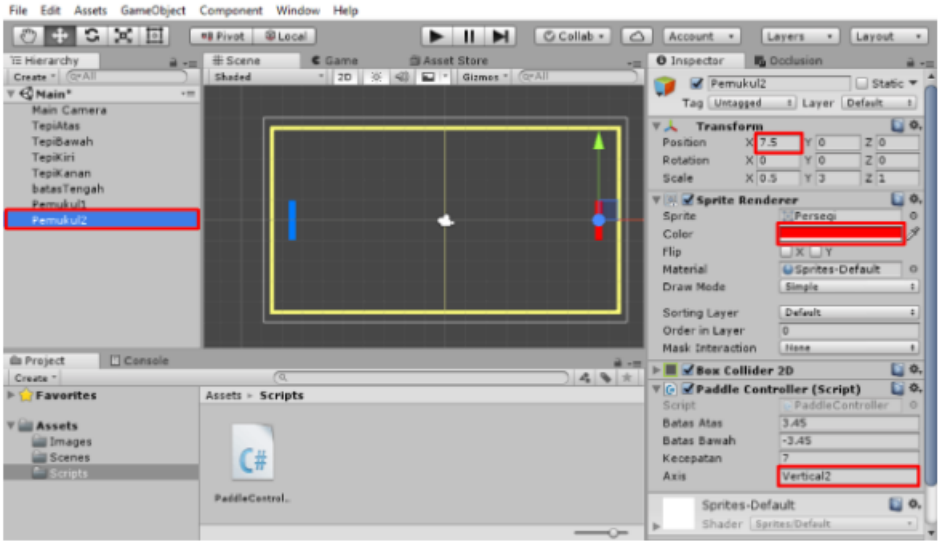


Kita telah mendapatkan nilai batas atas yaitu sebesar 3.45, sedangkan untuk nilai batas bawah adalah nilai negatif dari 3.45 yaitu -3.45. Setelah mendapatkan nilai tersebut, jangan lupa untuk mengembalikan posisi semula pada paddle dengan nilai y sama dengan 0.

3. Kemudian isikan kedua nilai tersebut pada field Batas Atas dan Batas Bawah.



4. Kemudian buatlah paddle lawan. Caranya, duplikasi GameObject **Pemukul1** di Hierarchy dengan nama **Pemukul2**. Lalu atur posisi x menjadi **7.5**. Ubahlah jadi warna yang berbeda. Atur juga nilai Axis menjadi **Vertical2 s**ehingga tampil seperti ini:



Sekarang Anda sudah dapat membuat paddle yang dapat bergerak ke atas dan ke bawah tanpa menembus dinding.

Membuat Bola

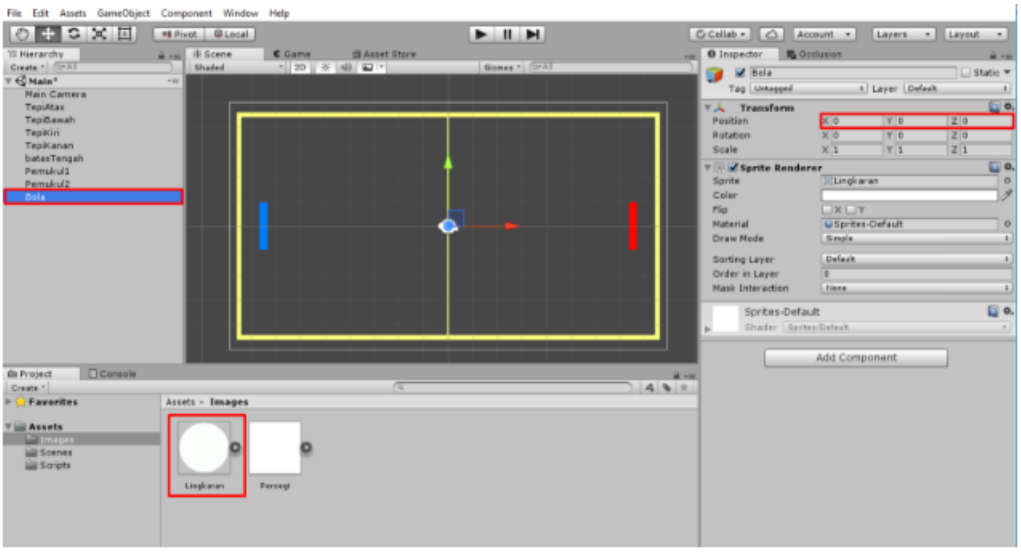
Di submodul ini, kita akan menerapkan 2D Physics ke Bola, antara lain sebagai berikut:

- **RigidBody 2D**, Sebuah komponen yang menerapan hukum fisika pada suatu gameobject 2D.
- **Collider 2D**, sebuah komponen yang menggambarkan bentuk sebuah fisik yang dapat digunakan untuk mendeteksi benturan dengan benda lain.
- **Physics Material 2D**, digunakan untuk mengatur nilai gesekan atau pantulan terhadap benturan benda lain.

Praktik: Membuat Objek Bola

Pertama kita membuat objek bola terlebih dahulu dengan cara sebagai berikut:

Masukkan gambar **Lingkaran** yang terdapat di panel **Project** ke panel **Hierarchy** kemudian ubah nama menjadi **Bola**. Atur posisi x, y dan z menjadi 0.

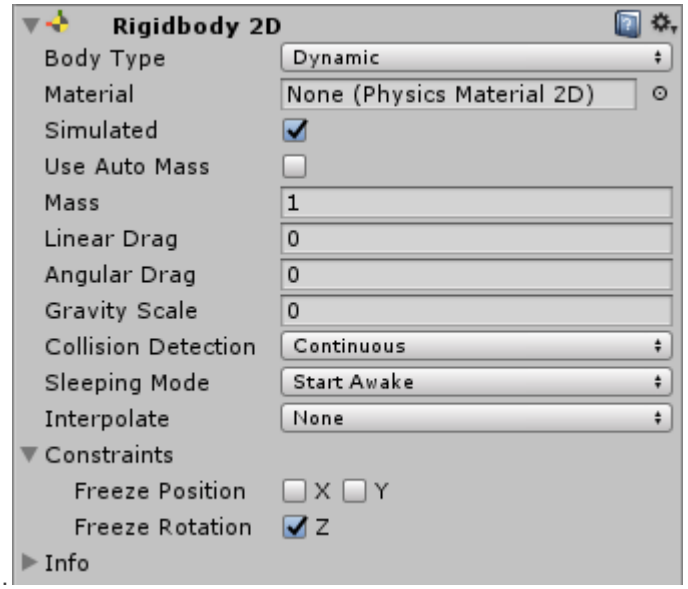


Praktik: Menambahkan Physics 2D pada Bola

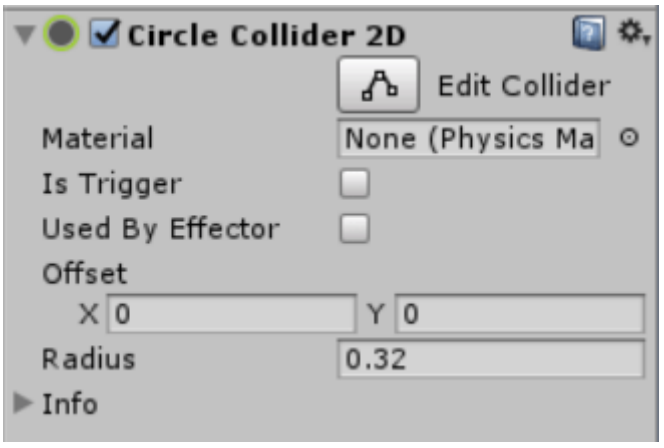
Setelah objek bola dibuat, tambahkan komponen bola.

1. Buka Inspector pada Bola, Lalu tambahkan komponen **RigidBody 2D** (Add Component > Physics 2D > Rigidbody 2D). Kemudian atur nilai **Linear**

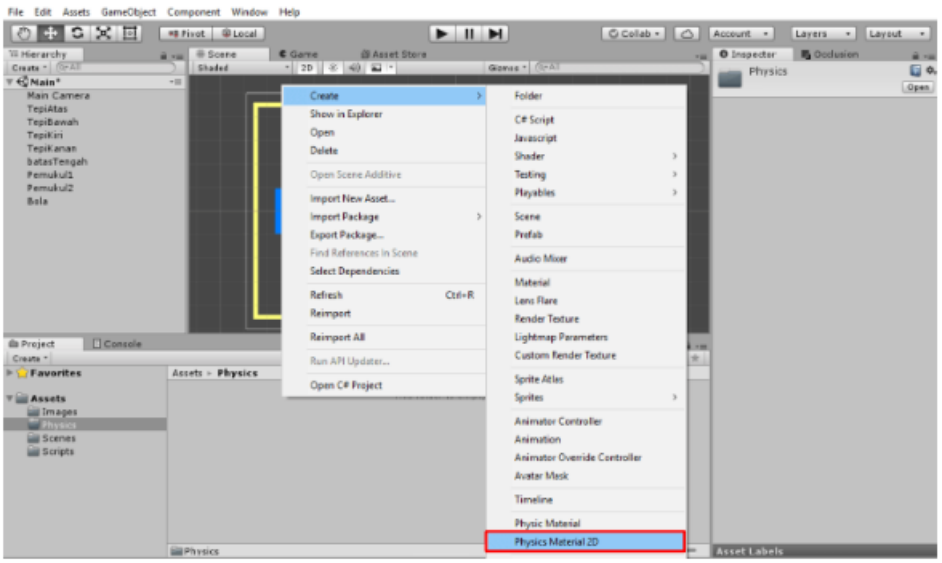
Drag, **Angular Drag**, **Gravity Scale** menjadi **0**. Jangan lupa untuk centang **Freeze Rotation Z**



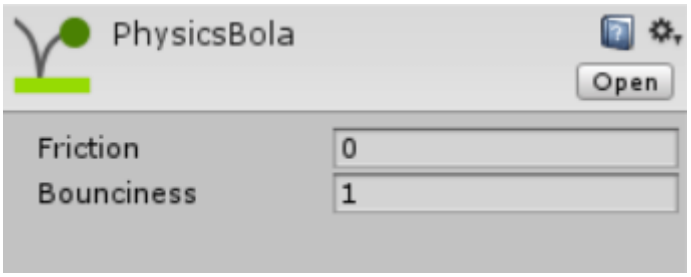
2. Tambahkan komponen **Circle Collider 2D** (**Add Component** > **Physics 2D** > **Circle Collider 2D**).



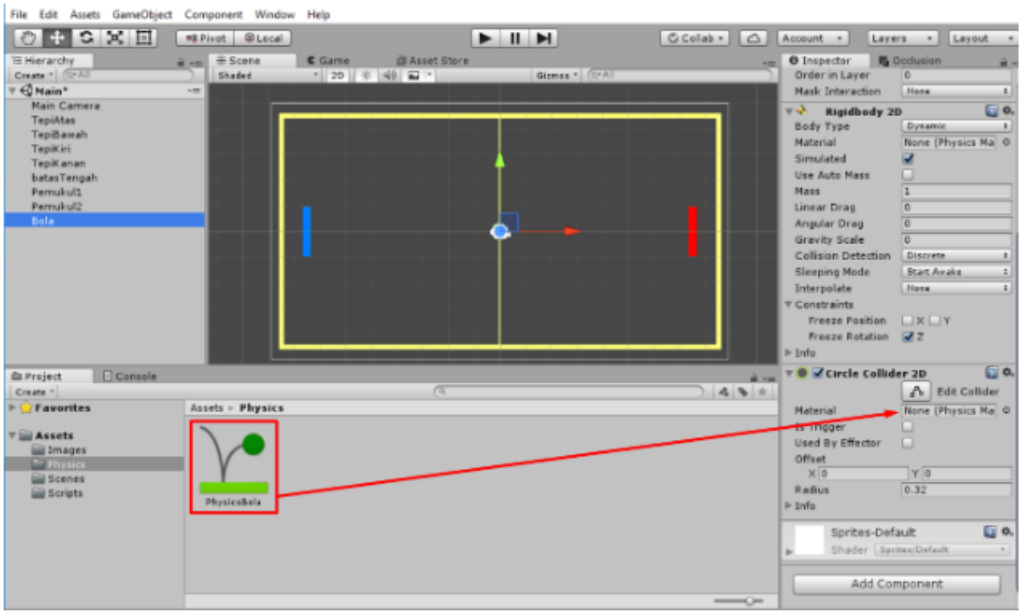
3. Buat folder baru dengan nama Physics. Kemudian buat **Physics Material 2D** dengan klik kanan pada panel **Project** > **Create** > **Physics Material 2D** dan beri nama **PhysicsBola**.



4. Klik **PhysicsBola** pada panel Project. Kemudian pada **Inspector** atur nilai sebagai berikut. Friction menjadi 0 dan Bounciness menjadi 1.



5. Terakhir, drag **PhysicsBola** dan drop ke field **Material** di Inspector **Bola**.



Praktik: Menambahkan Code

Kali ini kita akan mencoba menggerakkan Bola dengan menggunakan Force (tekanan). Seperti dalam hukum Fisika, kecepatan dan arah pergerakan Bola ditentukan oleh dua hal:

- **Arah** dan besaran **Force** (tekanan)
- **Massa bola**

Force dalam Unity ditambahkan pada Rigidbody 2D. Dalam tutorial ini, kita mencoba menambahkan Force sebagai pergerakan bola pertama kali. Berikut ini langkah-langkahnya.

1. Buat **Script C#** baru di dalam folder **Scripts** dengan nama **BallController.cs** kemudian tambahkan kode di bawah ini.
    - Tambahkan 2 variabel pada class BallController.

```
1. public int force;
2. Rigidbody2D rigid;
```

public int force; --> Dengan lewat inspector, nilai di atas untuk mengatur kecepatan gerak bola.

Rigidbody2D rigid; --> Menyimpan variable rigidbody yang dapat dipanggil sewaktu-waktu.

- Tambahkan kode ini di dalam function Start().

```
1. gid = GetComponent<Rigidbody2D>();
2. Vector2 arah = new Vector2(2, 0).normalized;
3. rigid.AddForce(arah * force);
```

rigid = GetComponent<Rigidbody2D>(); --> Memanggil komponen Rigidbody2D ke dalam script.

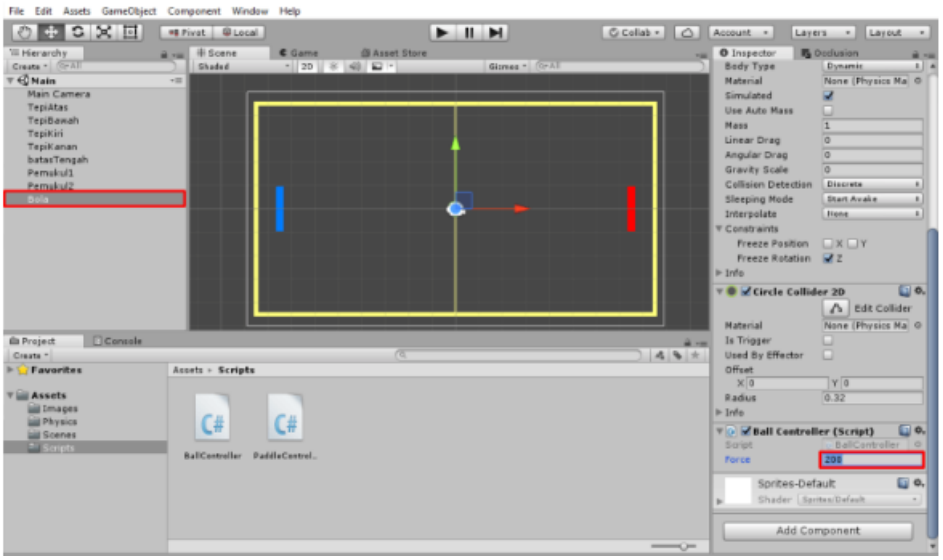
Vector2 arah = new Vector2(2, 0).normalized; --> Menyatakan arah dari Force, yaitu 2 satuan ke kanan dan 0 satuan ke atas.

rigid.AddForce(arah * force); --> Melontarkan bola sesuai dengan arah dan kekuatan.

2. Simpan dengan tekan **Ctrl+S sehingga seluruh kode menjadi seperti ini:**

```
using UnityEngine;
public class BallController : MonoBehaviour
{
    public int force;
    Rigidbody2D rigid;
    // Use this for initialization
    void Start()
    {
        rigid = GetComponent<Rigidbody2D>();
        Vector2 arah = new Vector2(2, 0).normalized;
        rigid.AddForce(arah * force);
    }
    // Update is called once per frame
    void Update()
    {
    }
}
```

3. Masukkan Script **BallController.cs** ke GameObject Bola yang terdapat di Hierarchy. Kemudian Atur nilai Force menjadi **200**.



Praktik: Menambahkan Reset Bola

Saat bola menyentuh tepi kanan dan tepi kiri, bola kembali ke posisi start. Jika bola mati di sebelah kanan maka bola akan mengarah ke kanan. Begitupun sebaliknya, jika bola mati di sebelah kiri maka bola akan mengarah ke kiri. Bagaimana cara mengaturnya? Ikuti langkah berikut ini:

1. Buka Script **BolaController.cs** dengan klik 2 kali. Lalu tambahkan kode sebagai berikut:

   - Tambahkan function di class **BolaController.**

```
1.  void ResetBall()
2.      {
3.          transform.localPosition = new Vector2(0, 0);
4.          rigid.velocity = new Vector2(0, 0);
5.      }
```

void ResetBall() --> Function ini dibuat supaya lebih ringkas.

transform.localPosition = new Vector2(0, 0); --> Memposisikan bola di tengah-tengah area.

rigid.velocity = new Vector2(0, 0); --> Tanpa memberi pergerakan pada bola.

   - Tambahkan function **OnCollisionEnter2D** di class **BolaController.**

```
private void OnCollisionEnter2D(Collision2D coll)
    {
        if (coll.gameObject.name == "TepiKanan")
        {
            ResetBall();
            Vector2 arah = new Vector2(2, 0).normalized;
            rigid.AddForce(arah * force);
        }
        if (coll.gameObject.name == "TepiKiri")
        {
            ResetBall();
            Vector2 arah = new Vector2(-2, 0).normalized;
            rigid.AddForce(arah * force);
        }
    }
```

private void OnCollisionEnter2D(Collision2D coll) --> Ketika bola mendeteksi suatu object.

if (coll.gameObject.name == "TepiKanan") --> Memastikan object tersebut adalah TepiKanan.

ResetBall(); --> Menjalakan seluruh code yang terdapat ResetBall yang meliputi mengatur ulang posisi bola dan pergerakan bola.

Vector2 arah = new Vector2(2, 0).normalized; --> Menentukan arah pergerakan bola. Nilai 2 berarti bola mengarah ke kanan.

rigid.AddForce(arah * force); --> Melontarkan bola berdasarkan arah dan kecepatan bola.

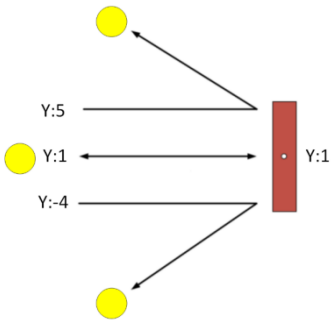
2. Simpan perubahan dengan tekan **Ctrl+S sehingga seluruh kode menjadi seperti ini:**

```
using UnityEngine;
public class BallController : MonoBehaviour
{
    public int force;
    Rigidbody2D rigid;
    // Use this for initialization
    void Start()
    {
```

```
9.          rigid = GetComponent<Rigidbody2D>();
10.         Vector2 arah = new Vector2(2, 0).normalized;
11.         rigid.AddForce(arah * force);
12.     }
13.     // Update is called once per frame
14.     void Update()
15.     {
16.     }
17.     private void OnCollisionEnter2D(Collision2D coll)
18.     {
19.         if (coll.gameObject.name == "TepiKanan")
20.         {
21.             ResetBall();
22.             Vector2 arah = new Vector2(2, 0).normalized;
23.             rigid.AddForce(arah * force);
24.         }
25.         if (coll.gameObject.name == "TepiKiri")
26.         {
27.             ResetBall();
28.             Vector2 arah = new Vector2(-2, 0).normalized;
29.             rigid.AddForce(arah * force);
30.         }
31.     }
32.     void ResetBall()
33.     {
34.         transform.localPosition = new Vector2(0, 0);
35.         rigid.velocity = new Vector2(0, 0);
36.     }
37. }
```

Praktik: Mengatur Pantulan Bola pada Paddle

Kita akan membuat sebuah pantulan berdasarkan posisi pantulan Bola terhadap posisi Paddle. Jika bola mengenai samping/pinggir Paddle, maka bola akan terpantul miring.



Ada beberapa hal yang perlu diperhatikan.Pantulan berdasarkan posisi bola mengenai Paddle. Titik Paddle berfungsi sebagai acuan Pantulan bola. Jika bola mengenai bagian atas paddle maka nilainya lebih besar 0 sehingga memantul ke atas. Sebaliknya, jika bola mengenai bagian bawah Paddle maka nilainya kurang dari 0 sehingga memantul ke bawah.

Misalnya bola mengenai ujung atas Paddle yang mana posisi y pada bola senilai 5 dan posisi Y pada paddle senilai 1, maka dari kedua nilai tersebut terdapat perbedaan 5 - 1 = 4 (lebih besar nilai 0 maka bola dipantul ke atas). Sedangkan jika posisi Y pada bola tepat dengan posisi Paddle yaitu sama-sama 1 maka nilai selisih yang didapat adalah 0 (bola akan dipantulkan lurus).

Maka yang akan kita lakukan pada bola adalah:

- Ketika Bola mengenai Paddle maka perhatikan kedua posisi.
- Hitung selisih dari kedua posisi tersebut sebagai arah lontar bola.
- Sebelum melontarkan bola, bola tersebut harus dalam keadaan diam.
- Lontarkan Bola.

Langkah-langkahnya antara lain:

1. Buka Script **BallController.cs** dengan klik 2 kali kmudian tambahkan kode sebagai berikut:
2. Tambahkan kode ini di dalam function OnCollisionEnter2D(Collision2D coll).

```
1.          if (coll.gameObject.name == "Pemukul1" || coll.gameObject.name == "Pemukul2")
2.          {
3.              float sudut = (transform.position.y - coll.transform.position.y) * 5f;
4.              Vector2 arah = new Vector2(rigid.velocity.x, sudut).normalized;
5.              rigid.velocity = new Vector2(0, 0);
```

```
            rigid.AddForce(arah * force * 2);
        }
```

if (coll.gameObject.name == "Pemukul1" || coll.gameObject.name == "Pemukul2") --> Memastikan objek yang bersentuhan dengan bola adalah GameObject Pemukul1 dan Pemukul2.

float sudut = (transform.position.y - coll.transform.position.y) * 5f; --> Kemudian hitung seberapa kemiringan yang diberikan ke bola.

Vector2 arah = new Vector2(rigid.velocity.x, sudut).normalized; --> Menentukan arah bola yang akan dipantulkan.

rigid.velocity = new Vector2(0, 0); --> Reset gerak bola (dengan kode ini, bola akan diam).

rigid.AddForce(arah * force * 2); --> Implementasikan arah, kekuatan lontar dan kecepatan setelah bola menyentuh paddle. Di bagian ini Anda dapat memodifikasi agar bola semakin cepat dengan menambahkan kelipatan setiap menjalankan kode ini.

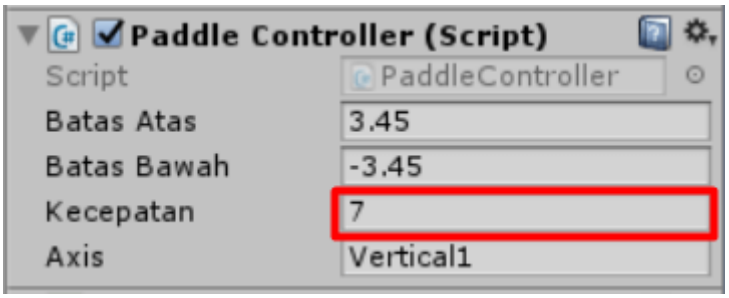
3. Simpan perubahan dengan tekan **Ctrl+S** sehingga seluruh kode menjadi seperti ini:

```
using UnityEngine;
public class BallController : MonoBehaviour
{
    public int force;
    Rigidbody2D rigid;
    // Use this for initialization
    void Start()
    {
        rigid = GetComponent<Rigidbody2D>();
        Vector2 arah = new Vector2(2, 0).normalized;
        rigid.AddForce(arah * force);
    }
    // Update is called once per frame
    void Update()
    {
    }
    private void OnCollisionEnter2D(Collision2D coll)
    {
        if (coll.gameObject.name == "TepiKanan")
```

```
        {
            ResetBall();
            Vector2 arah = new Vector2(2, 0).normalized;
            rigid.AddForce(arah * force);
        }
        if (coll.gameObject.name == "TepiKiri")
        {
            ResetBall();
            Vector2 arah = new Vector2(-2, 0).normalized;
            rigid.AddForce(arah * force);
        }
        if (coll.gameObject.name == "Pemukul1" || coll.gameObject.name == "Pemukul2")
        {
            float sudut = (transform.position.y - coll.transform.position.y) * 5f;
            Vector2 arah = new Vector2(rigid.velocity.x, sudut).normalized;
            rigid.velocity = new Vector2(0, 0);
            rigid.AddForce(arah * force * 2);
        }
    }
    void ResetBall()
    {
        transform.localPosition = new Vector2(0, 0);
        rigid.velocity = new Vector2(0, 0);
    }
}
```

Jika Anda masih kurang nyaman dengan pergerakan bola dan paddle, buka panel Inspector pada masing masing gameobject tersebut.

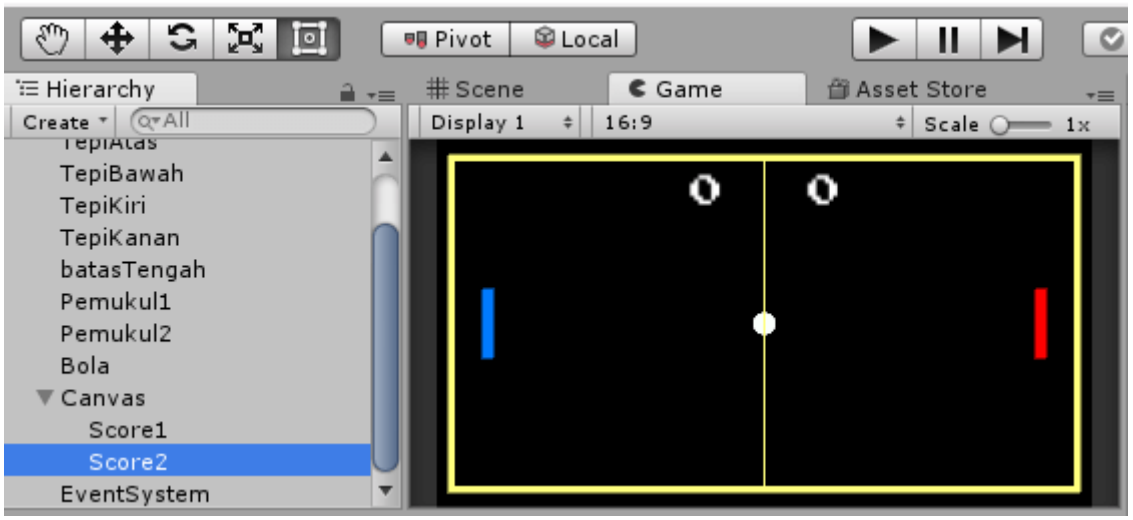
Atur kecepatan pada Paddle.



Atur force untuk kekuatan lontar pada Bola.

Membuat Skor

Setelah gameplay sederhana dapat dimainkan, selanjutnya kita akan membuat skor. Di bagian ini selain menghitung perolehan skor , kita juga akan belajar tentang UI, cara mengaturnya, dan implementasi Font. Anda dapat menggunakan Text, Font dan mengatur letak posisi tersebut pada nilai score yang ditampilkan. Hal tersebut merupakan contoh penggunaan komponen UI.



Praktik: Menambahkan Font

Penggunaan teks sangat berhubungan dengan font. Anda dapat menggunakan berbagai model font untuk menampilkan sebuah teks. Beberapa cara untuk mendapatkan sebuah font antara lain:

- Anda dapat mengunduh font dari berbagai situs dari yang gratis dan berbayar. Jangan lupa perhatikan lisensinya ya.
- Anda dapat menggunakan font dari Standard Asset Unity dengan mengimpor Utilities.
- Anda dapat membuat font sendiri dengan menggunakan tools vector seperti Adobe illustrator, dll.

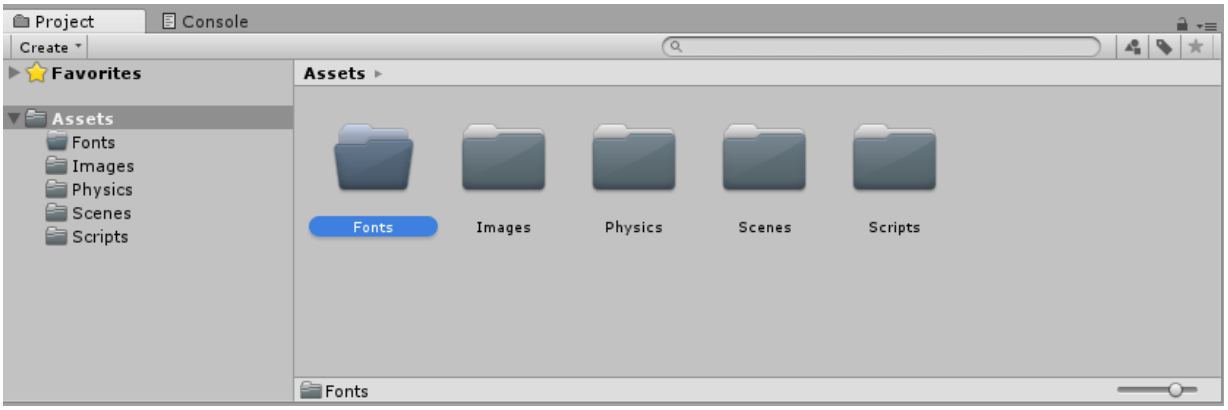
Pastikan style font selaras dengan style game Anda. Sebagai contoh, mengingat game pukul bola tergolong game klasik mari kita gunakan font pixel saja.

Beberapa referensi bisa dilihat di sini http://www.1001fonts.com/pixel-fonts.html

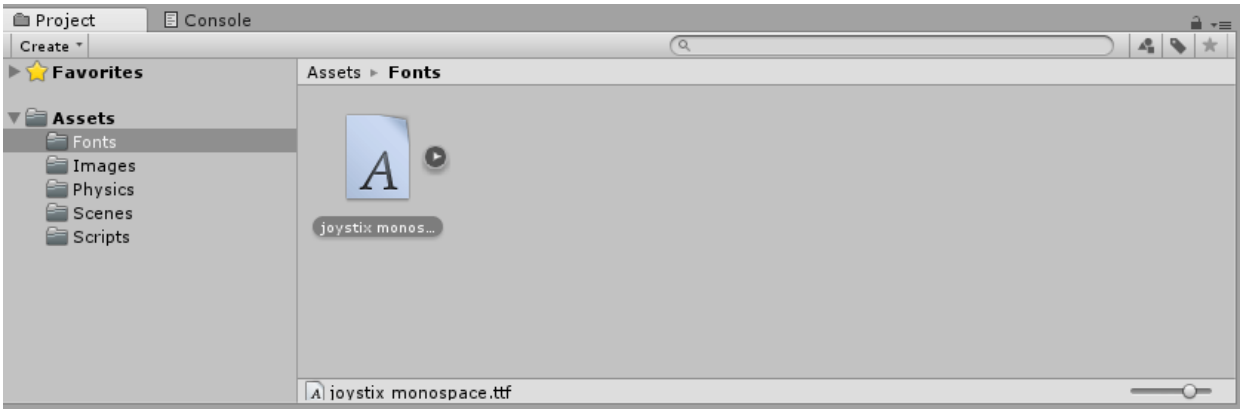
Pada kelas ini, kita menggunakan font berikut http://www.1001fonts.com/joystix-font.html

Langkah-langkahnya sebagai berikut:

1. Setelah Anda memiliki sebuah font buatlah folder baru dengan nama **Fonts.**



2. Tambahkan berkas font **"joystix monospace.ttf"** ke dalam folder Fonts. (Klik kanan > Import New Asset).

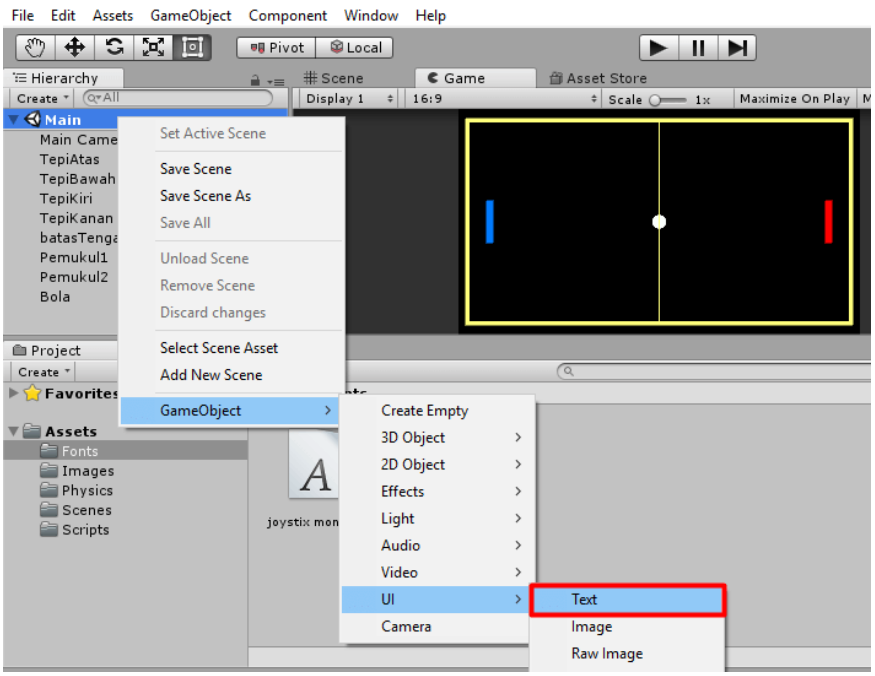


Sekarang Anda sudah memiliki font yang siap digunakan untuk UI Text.

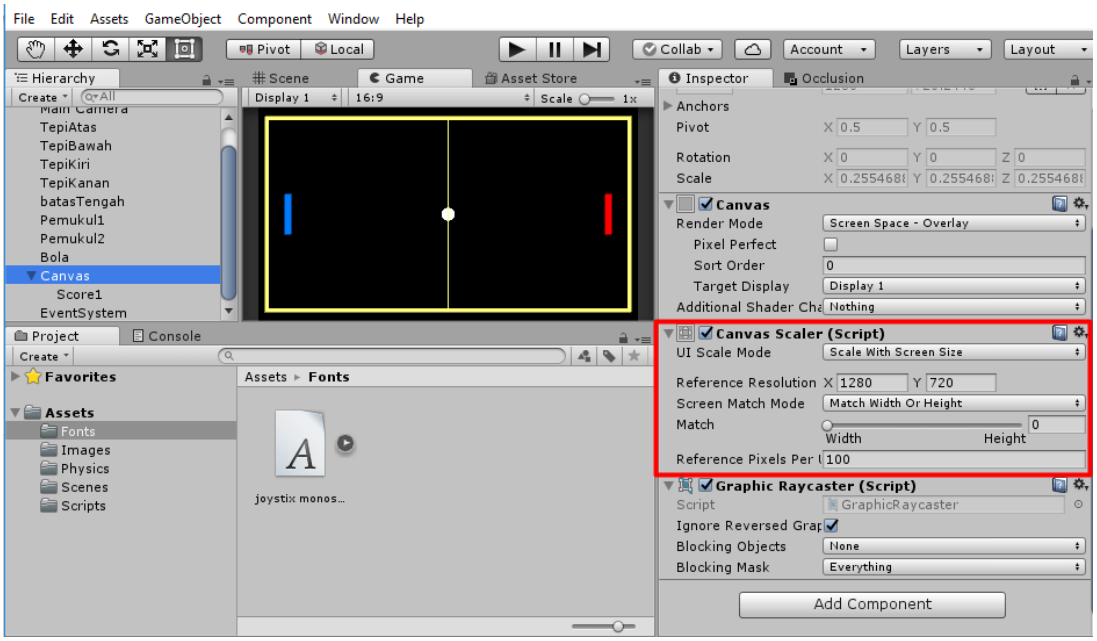
Praktik: Membuat UI Skor

Kita akan menampilkan skor yang telah diperoleh, kemudian menampilkan skor tersebut dengan UI Text. Langkah-langkahnya adalah sebagai berikut:

1. Tambahkan Teks baru dengan klik kanan pada Scene Main kemudian pilih GameObject > UI > Text dan ubah nama GameObject **Text** menjadi **Score1**.

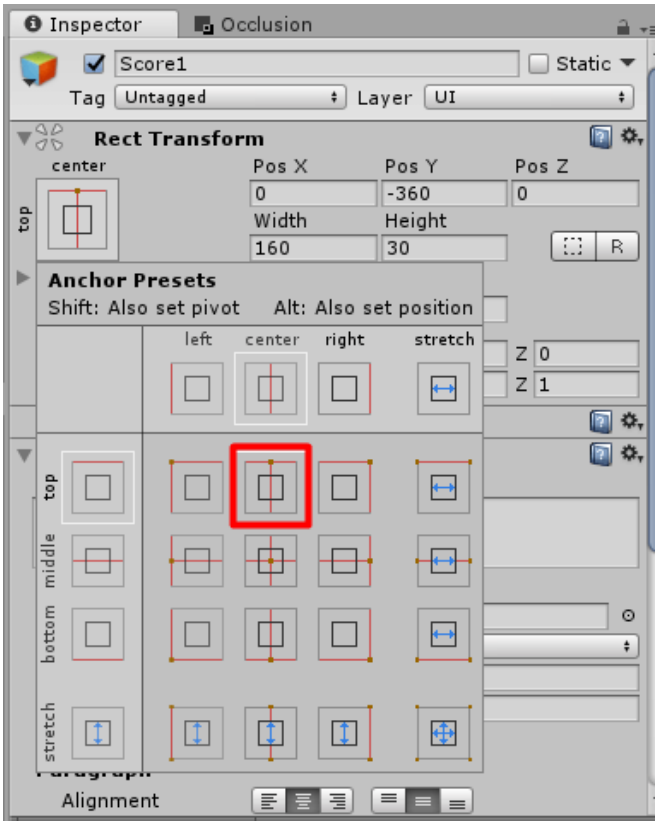


2. Klik Canvas. Pada Inspector atur **UI Scale Mode** di komponen Canvas Scaler menjadi **Scale With Screen.** Atur Nilai Reference Resolution menjadi X: **1280** dan Y: **720.**

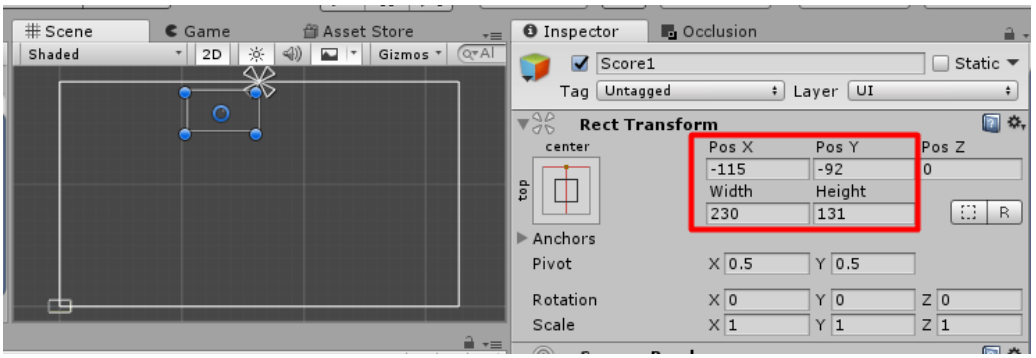


3. Klik Score1 kemudian pada Inspector:

    - Atur nilai posisi Rect Transform menjadi Top-Center.

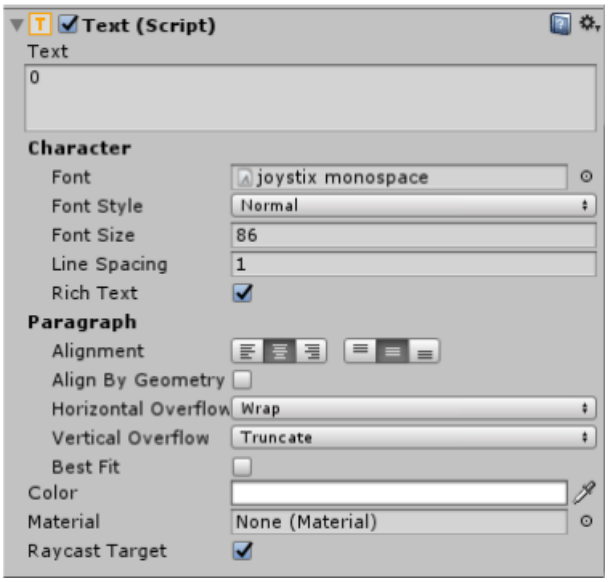


o Atur juga nilai Pos X: -115, Pos Y: -92, Width: 230 dan Height: 131.

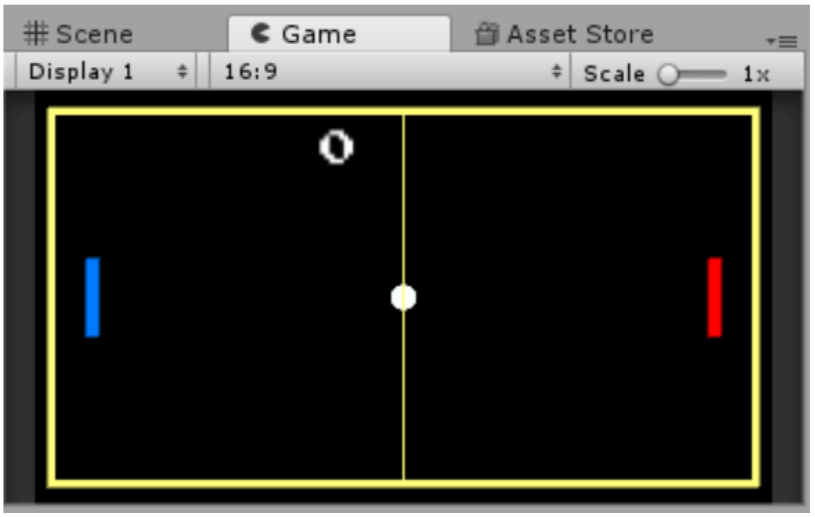


o Pada komponen text, aturlah dengan nilai sebagai berikut:

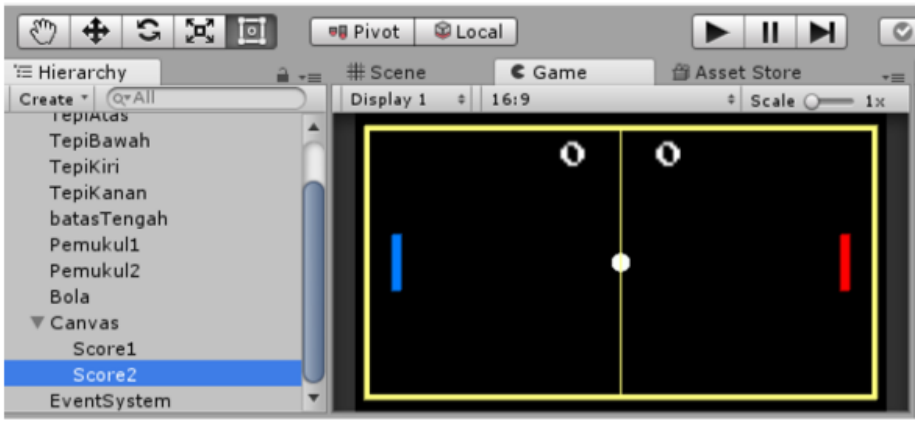
- Text: 0
- Font: joystix monospace
- Font size: 86
- Alignment: Center - Center
- Color: Putih



4. Sehingga tampilan pada panel Game menjadi seperti ini:



5. Duplikasi GameObject **Score1** dan ubah namanya dari **Score1 (1)** menjadi **Score2.** Pada Inspector atur **Pos X** menjadi **115**.



Praktik: Deklarasi dan Inisialisasi Variable Score

Selanjutnya kita akan masuk ke scripting. Pertama kita akan membuat variabel untuk menyimpan skor.

1. Buka Script **BallController.cs** kemudian tambahkan kode sebagai berikut:
   - Tambahkan variabel skor untuk **Pemukul1** dan **Pemukul2**.

```
int scoreP1;
int scoreP2;
```

   - Tambahkan inisialisasi variabel skor di function **Start().**

```
scoreP1 = 0;
scoreP2 = 0;
```

2. Simpan perubahan dengan tekan **Ctrl+S.**

Praktik: Penghitungan Score dan Reset Posisi Bola

Selanjutnya pada prosedur **OnCollisionEnter2D** kita perlu menambahkan kode apabila bola bertabrakan dengan **tepi kiri** atau **tepi kanan**. Kode tersebut akan mengatur penambahan **scoreP1** dan **scoreP2.**

1. Tambahkan isi prosedur **OnCollisionEnter** sebagai berikut:
   - Tambahkan kode ini ketika bola menyentuh Tepi Kanan.

```
scoreP1 += 1;
```

- Tambahkan kode ini ketika bola menyentuh Tepi Kiri.

```
2.  scoreP2 += 1;
```

2. Simpan perubahan dengan tekan **Ctrl+S sehingga seluruh kode menjadi seperti ini:**

```
1.  using System.Collections;
2.  using System.Collections.Generic;
3.  using UnityEngine;
4.  public class BallController : MonoBehaviour
5.  {
6.      public int force;
7.      Rigidbody2D rigid;
8.      int scoreP1;
9.      int scoreP2;
10.     // Use this for initialization
11.     void Start()
12.     {
13.         rigid = GetComponent<Rigidbody2D>();
14.         Vector2 arah = new Vector2(2, 0).normalized;
15.         rigid.AddForce(arah * force);
16.         scoreP1 = 0;
17.         scoreP2 = 0;
18.     }
19.     // Update is called once per frame
20.     void Update()
21.     {
22.     }
23.     private void OnCollisionEnter2D(Collision2D coll)
24.     {
25.         if (coll.gameObject.name == "TepiKanan")
26.         {
27.             scoreP1 += 1;
28.             ResetBall();
29.             Vector2 arah = new Vector2(2, 0).normalized;
30.             rigid.AddForce(arah * force);
31.         }
32.         if (coll.gameObject.name == "TepiKiri")
33.         {
34.             scoreP2 += 1;
35.             ResetBall();
36.             Vector2 arah = new Vector2(-2, 0).normalized;
37.             rigid.AddForce(arah * force);
38.         }
39.         if (coll.gameObject.name == "Pemukul1" || coll.gameObject.name == "Pemukul2")
```

```
        {
            float sudut = (transform.position.y - coll.transform.position.y) * 5f;
            Vector2 arah = new Vector2(rigid.velocity.x, sudut).normalized;
            rigid.velocity = new Vector2(0, 0);
            rigid.AddForce(arah * force * 2);
        }
    }
    void ResetBall()
    {
        transform.localPosition = new Vector2(0, 0);
        rigid.velocity = new Vector2(0, 0);
    }
}
```

Praktik: Menuliskan Score ke Layar

Setelah menambahkan text (Score1 dan Score2 pada Canvas) kita perlu menampilkan skor yang diperhitungan ke layar. Penambahan skor diatur dalam Script BallController.cs. Oleh karena itu, penulisan skor juga akan ditangani dalam script ini dengan cara sebagai berikut:

1. Buka Script BallController.cs dengan klik 2 kali kemudian tambahkan kode sebagai berikut:

   - Tambahkan library **UnityEngine.UI.**

     ```
     Using UnityEngine.UI;
     ```

     Digunakan untuk mengakses komponen-komponen untuk UI.

   - Tambahkan variabel di dalam class **BallController.**

     ```
     Text scoreUIP1;
     Text scoreUIP2;
     ```

     Digunakan untuk menyimpan gameobject text.

- Tambahkan inisialisasi variabel **scoreUIP1** dan **scroreUIP2.**

```
scoreUIP1 = GameObject.Find ("Score1").GetComponent<Text> ();
scoreUIP2 = GameObject.Find ("Score2").GetComponent<Text> ();
```

Fungsinya untuk mengakses GameObject yang memiliki Nama Score1 dan Score 2. Kemudian dari GameObject tersebut dicari komponen Text yang ada di dalamnya. Komponent Text ini kemudian disimpan ke scoreUIP1 dan Score UIP2.

- Tambahkan function di dalam class **BallController.**

```
void TampilkanScore()
{
    Debug.Log("Score P1: " + scoreP1 + " Score P2: " + scoreP2);
    scoreUIP1.text = scoreP1 + "";
    scoreUIP2.text = scoreP2 + "";
}
```

Digunakan untuk mengimplementasikan penampilan skor dengan memanggil function TampilanScore().

- Tambahkan kode ini ketika scoreP1 dan scoreP2 ditambahkan.

```
TampilkanScore();
```

2. Simpan perubahan dengan tekan **Ctrl+S** sehingga seluruh kode menjadi seperti berikut:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class BallController : MonoBehaviour
{
    public int force;
    Rigidbody2D rigid;
    int scoreP1;
    int scoreP2;
    Text scoreUIP1;
```

```
    Text scoreUIP2;
    // Use this for initialization
    void Start()
    {
        rigid = GetComponent<Rigidbody2D>();
        Vector2 arah = new Vector2(2, 0).normalized;
        rigid.AddForce(arah * force);
        scoreP1 = 0;
        scoreP2 = 0;
        scoreUIP1 = GameObject.Find("Score1").GetComponent<Text>();
        scoreUIP2 = GameObject.Find("Score2").GetComponent<Text>();
    }
    // Update is called once per frame
    void Update()
    {
    }
    private void OnCollisionEnter2D(Collision2D coll)
    {
        if (coll.gameObject.name == "TepiKanan")
        {
            scoreP1 += 1;
            TampilkanScore();
            ResetBall();
            Vector2 arah = new Vector2(2, 0).normalized;
            rigid.AddForce(arah * force);
        }
        if (coll.gameObject.name == "TepiKiri")
        {
            scoreP2 += 1;
            TampilkanScore();
            ResetBall();
            Vector2 arah = new Vector2(-2, 0).normalized;
            rigid.AddForce(arah * force);
        }
        if (coll.gameObject.name == "Pemukul1" || coll.gameObject.name == "Pemukul2")
        {
            float sudut = (transform.position.y - coll.transform.position.y) * 5f;
            Vector2 arah = new Vector2(rigid.velocity.x, sudut).normalized;
            rigid.velocity = new Vector2(0, 0);
            rigid.AddForce(arah * force * 2);
        }
    }
    void ResetBall()
    {
```

```
56.         transform.localPosition = new Vector2(0, 0);
57.         rigid.velocity = new Vector2(0, 0);
58.     }
59.     void TampilkanScore()
60.     {
61.         Debug.Log("Score P1: " + scoreP1 + " Score P2: " + scoreP2);
62.         scoreUIP1.text = scoreP1 + "";
63.         scoreUIP2.text = scoreP2 + "";
64.     }
65. }
```

Menampilkan Halaman Selesai

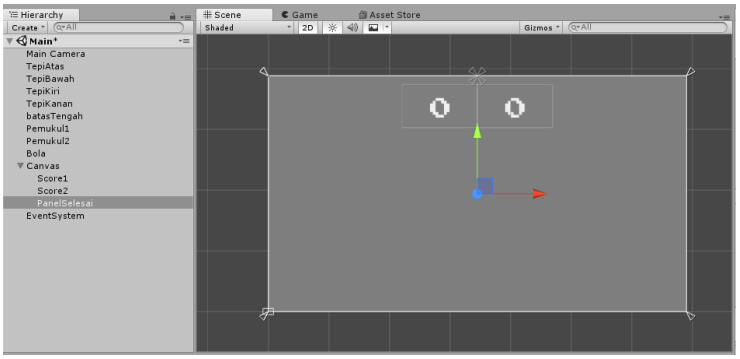
Halaman Selesai/Halaman Gameover wajib ada di setiap akhir kita menyelesaikan sebuah game. Di submodul ini kita akan membahas tentang cara menampilkan halaman selesai sesaat setelah salah satu dari player mencapai target skor.

Seperti ini ketika dijalankan:

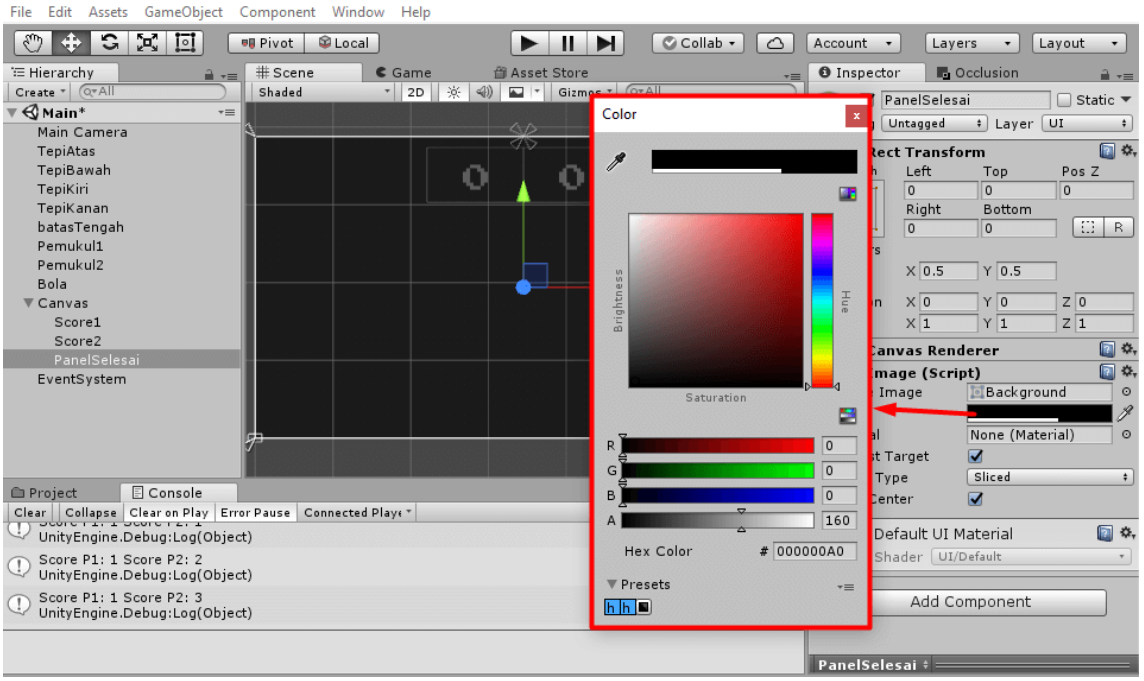
Praktik: Membuat UI GameOver

---

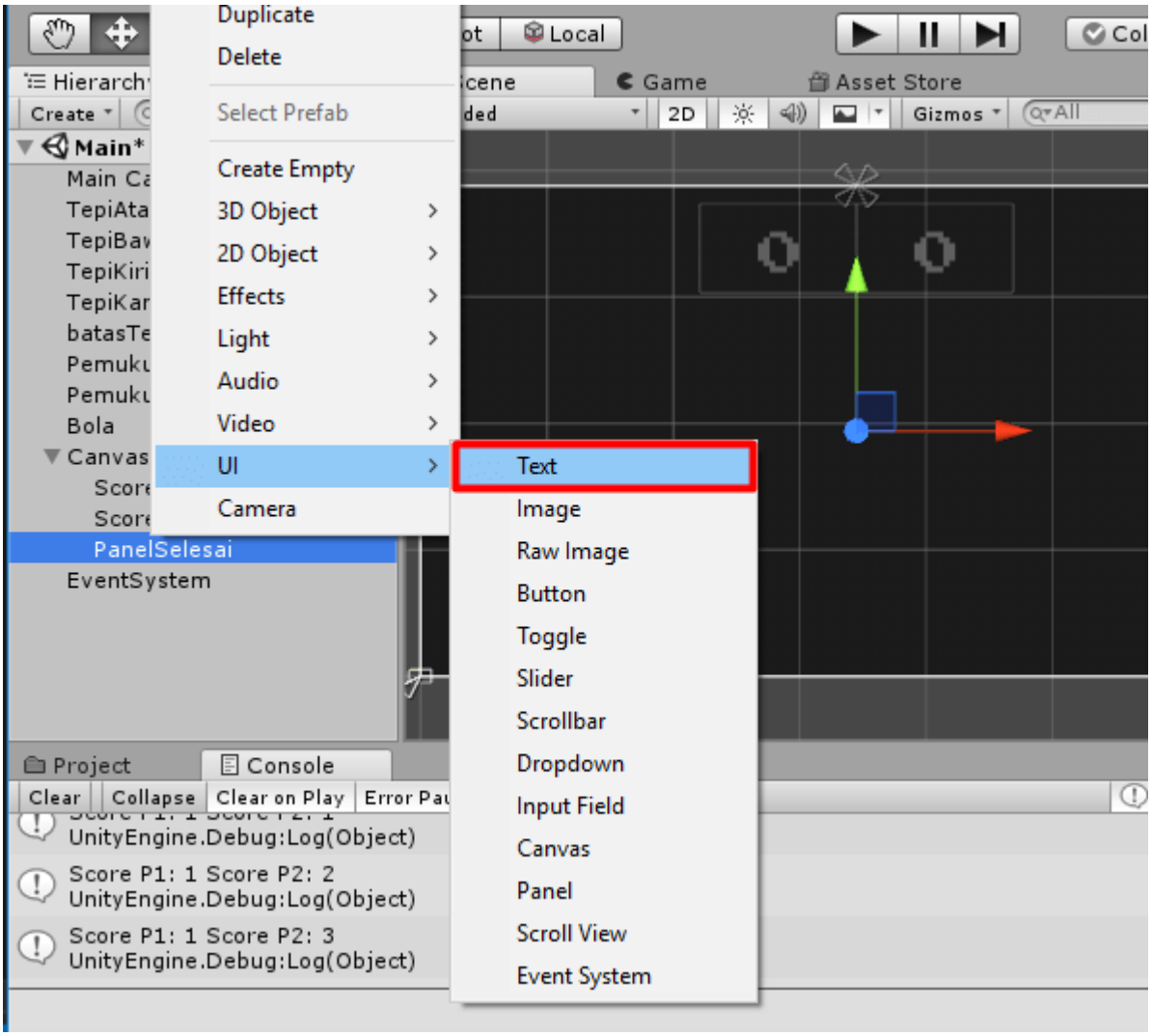
1. Tambahkan UI Panel pada Canvas dengan cara klik kanan pada Canvas kemudian pilih **UI** > **Panel**. Ubah Nama dari **Panel** menjadi **PanelSelesai**.



2. Atur warna pada PanelSelesai dengan warna yang lebih gelap agar teks yang di atasnya dapat lebih terbaca.



3. Tambahkan 2 Text di dalam Panel (Klik kanan pada PanelSelesai, Pilih **UI** > **Text**).

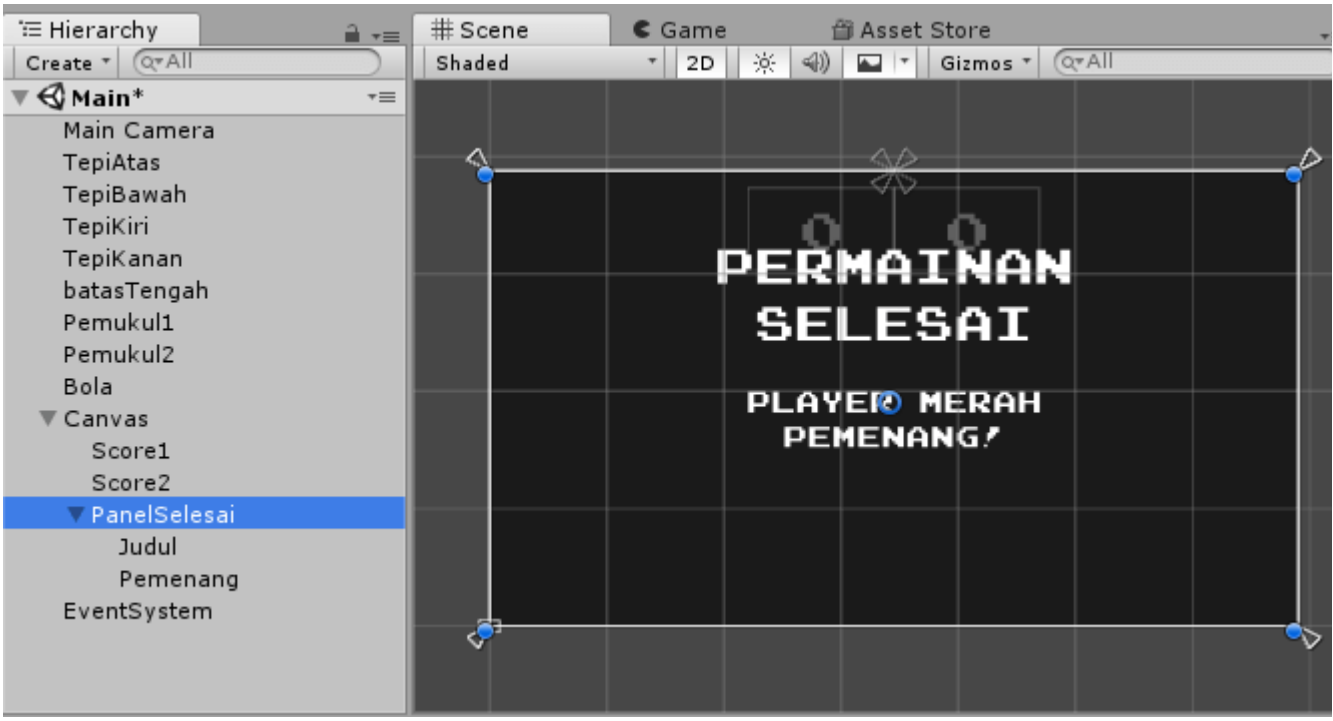


Aturlah kedua Text sebagai berikut:

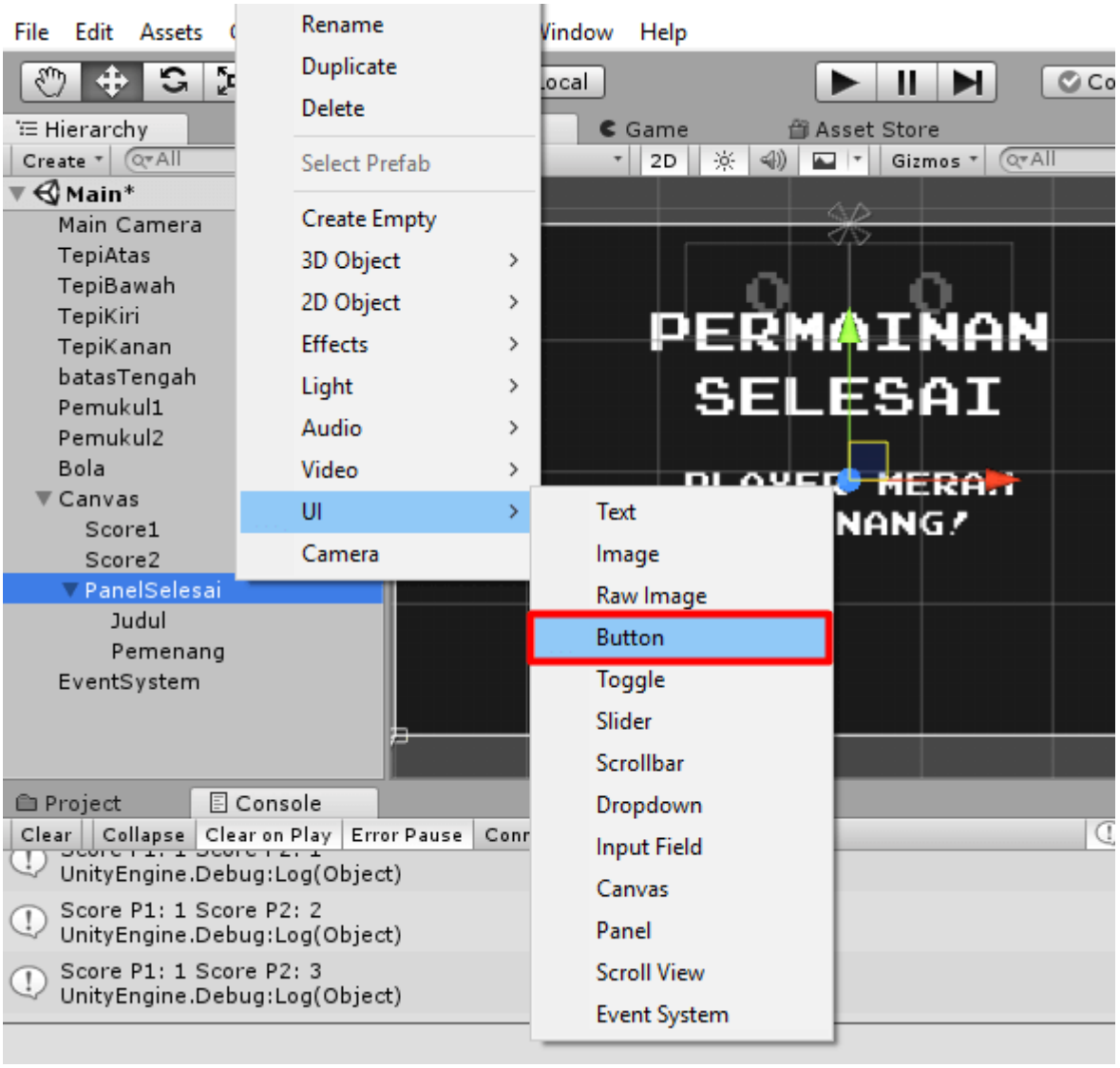
1. **Text 1**
    1. Nama Object: Judul
    2. Text: Permainan Selesai
    3. Font: joystix monospace
    4. Font Size: 77
    5. Alignment: Center-Center
    6. Color: Putih
2. **Text 2**

    1. Nama Object: Pemenang
    2. Text: Player Merah Pemenang!
    3. Font: joystix monospace
    4. Font Size: 47
    5. Alignment: Center-Center
    6. Color: Putih

Jangan lupa untuk mengatur posisi dan width dan height pada Rect Transform, Sehingga nampak seperti ini:



4. Tambahkan 2 Button di dalam PanelSelesai (Klik kanan pada **Panel selesai** > **UI** > **Button**).



Aturlah kedua Text sebagai berikut:

- **Button 1**
  - Nama Object: BtnMenu
  - Width: > 271
  - Height: > 92
  - Source Image: Persegi
  - Normal Color: Putih
  - Pressed Color: Oranye
- **Text pada Botton 1**
  - Text: Kembali Ke Main Manu
  - Font: joystix monospace
  - Font Size: 26
  - Alignment: Center-Center
  - Color: Hitam
- **Button 2**
  - Nama Object: BtnUlangi

- Width: > 271
- Height: > 92
- Source Image: Persegi
- Normal Color: Putih
- Pressed Color: Oranye

- **Text pada Botton 2**

  - Text: Ulangi Permainan
  - Font: joystix monospace
  - Font Size: 26
  - Alignment: Center-Center
  - Color: Hitam
  - Seperti ini ketika dijalankan:

Praktik: Membuat Kondisi untuk Menampilkan Halaman Selesai

Setelah halaman selesai , berikutnya kita membuat kondisi bahwa jika salah satu player mencapai skor 5 maka dialah pemenangnya. Seperti ini alurnya:

- Tampilkan skor.
- Cek skor apakah sudah mencapai nilai 5.
- Jika iya, tampilkan halaman selesai dan hentikan permainan (menghilangkan bola).

Langkah-langkahnya antara lain:

1. Buka Script **BallController.cs** dan tambahkan kode sebagai berikut:

   - Tambahkan variabel di dalam class **BallController.**

```
GameObject panelSelesai;
Text txPemenang;
```

     Digunakan untuk menghandle UI Selesai.

- Tambahkan inisialisasi variable panelSelesai tersebut di function **Start().**

```
3.    panelSelesai = GameObject.Find ("PanelSelesai");
4.    panelSelesai.SetActive (false);
```

Kode di atas memanggil GameObject PanelSelesai yang terdapat di Hierarchy lalu memastikannya dalam keadaaan non aktif.

- Tambahkan kode ini setelah menampilkan skor saat bola menyentuh tepi kanan.

```
1.  if (scoreP1 == 5)
2.  {
3.      panelSelesai.SetActive (true);
4.      txPemenang = GameObject.Find ("Pemenang").GetComponent<Text> ();
5.      txPemenang.text = "Player Biru Pemenang!";
6.      Destroy (gameObject);
7.      return;
8.  }
```

if (scoreP1 == 5) --> Jika skor pemukul 1 lebih dari 5.

panelSelesai.SetActive(true); --> Maka halaman selesai ditampilkan.

txPemenang = GameObject.Find("Pemenang").GetComponent<Text>(); --> Cari Komponen UI Text pada GameObject Pemenang.

txPemenang.text = "Player Biru Pemenang!"; --> Tampilkan text Player Biru Pemenang.

Destroy(gameObject); --> Hilangkan bola.

Return; --> Kode selesai dan tidak lanjut membaca kode berikutnya.

- Tambahkan kode ini setelah menampilkan skor saat bola menyentuh tepi kiri.

```
1.  if (scoreP2 == 5)
2.  {
3.      panelSelesai.SetActive (true);
4.      txPemenang = GameObject.Find ("Pemenang").GetComponent<Text> ();
5.      txPemenang.text = "Player Merah Pemenang!";
6.      Destroy (gameObject);
7.      return;
8.  }
```

2. Simpan dengan tekan **Ctrl+S** sehingga seluruh kode menjadi sebagai berikut:

```
1.  using System.Collections;
```

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


public class BallController : MonoBehaviour
{


    public int force;
    Rigidbody2D rigid;


    int scoreP1;
    int scoreP2;


    Text scoreUIP1;
    Text scoreUIP2;


    GameObject panelSelesai;
    Text txPemenang;


    // Use this for initialization
    void Start ()
    {
        rigid = GetComponent<Rigidbody2D> ();
        Vector2 arah = new Vector2 (2, 0).normalized;
        rigid.AddForce (arah * force);


        scoreP1 = 0;
        scoreP2 = 0;


        scoreUIP1 = GameObject.Find ("Score1").GetComponent<Text> ();
        scoreUIP2 = GameObject.Find ("Score2").GetComponent<Text> ();


        panelSelesai = GameObject.Find ("PanelSelesai");
        panelSelesai.SetActive (false);
    }
```

```
46.
47.
48.     // Update is called once per frame
49.     void Update ()
50.     {
51.
52.
53.     }
54.
55.
56.     private void OnCollisionEnter2D (Collision2D coll)
57.     {
58.         if (coll.gameObject.name == "TepiKanan") {
59.             scoreP1 += 1;
60.             TampilkanScore ();
61.             if (scoreP1 == 5) {
62.                 panelSelesai.SetActive (true);
63.                 txPemenang = GameObject.Find ("Pemenang").GetComponent<Text> ();
64.                 txPemenang.text = "Player Biru Pemenang!";
65.                 Destroy (gameObject);
66.                 return;
67.             }
68.             ResetBall ();
69.             Vector2 arah = new Vector2 (2, 0).normalized;
70.             rigid.AddForce (arah * force);
71.         }
72.
73.
74.         if (coll.gameObject.name == "TepiKiri") {
75.             scoreP2 += 1;
76.             TampilkanScore ();
77.             if (scoreP2 == 5) {
78.                 panelSelesai.SetActive (true);
79.                 txPemenang = GameObject.Find ("Pemenang").GetComponent<Text> ();
80.                 txPemenang.text = "Player Merah Pemenang!";
81.                 Destroy (gameObject);
82.                 return;
83.             }
84.             ResetBall ();
85.             Vector2 arah = new Vector2 (-2, 0).normalized;
86.             rigid.AddForce (arah * force);
87.         }
88.
89.
```

```
90.         if (coll.gameObject.name == "Pemukul1" || coll.gameObject.name == "Pemukul2") {
91.             float sudut = (transform.position.y - coll.transform.position.y) * 5f;
92.             Vector2 arah = new Vector2 (rigid.velocity.x, sudut).normalized;
93.             rigid.velocity = new Vector2 (0, 0);
94.             rigid.AddForce (arah * force * 2);
95.         }
96.
97.
98.     }
99.
100.
101.         void ResetBall ()
102.         {
103.             transform.localPosition = new Vector2 (0, 0);
104.             rigid.velocity = new Vector2 (0, 0);
105.         }
106.
107.
108.         void TampilkanScore ()
109.         {
110.             Debug.Log ("Score P1: " + scoreP1 + " Score P2: " + scoreP2);
111.             scoreUIP1.text = scoreP1 + "";
112.             scoreUIP2.text = scoreP2 + "";
113.         }
114.     }
```

Membuat dan Mengelola Halaman

Halaman utama/menu merupakan titik awal player memulai game.

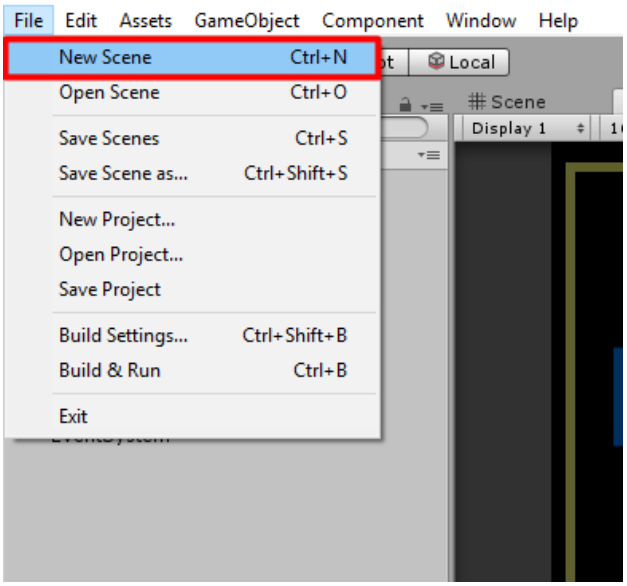
Biasanya di halaman utama/menu terdapat beberapa halaman navigasi berupa tombol yang berfungsi untuk mengakses antara lain:

- permainan
- halaman kredit
- keluar
- tutorial

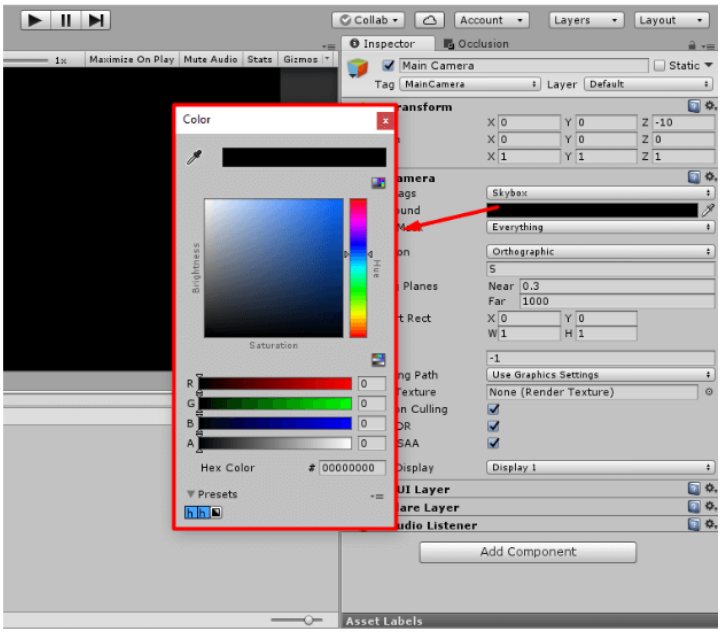
Di pembelajaran ini kita hanya menggunakan satu tombol untuk masuk ke permainan. Selain membuat halaman utama, tombol juga mengatur perpindahan halaman baik di halaman utama maupun di halaman selesai.

Praktik: Membuat Halaman Menu

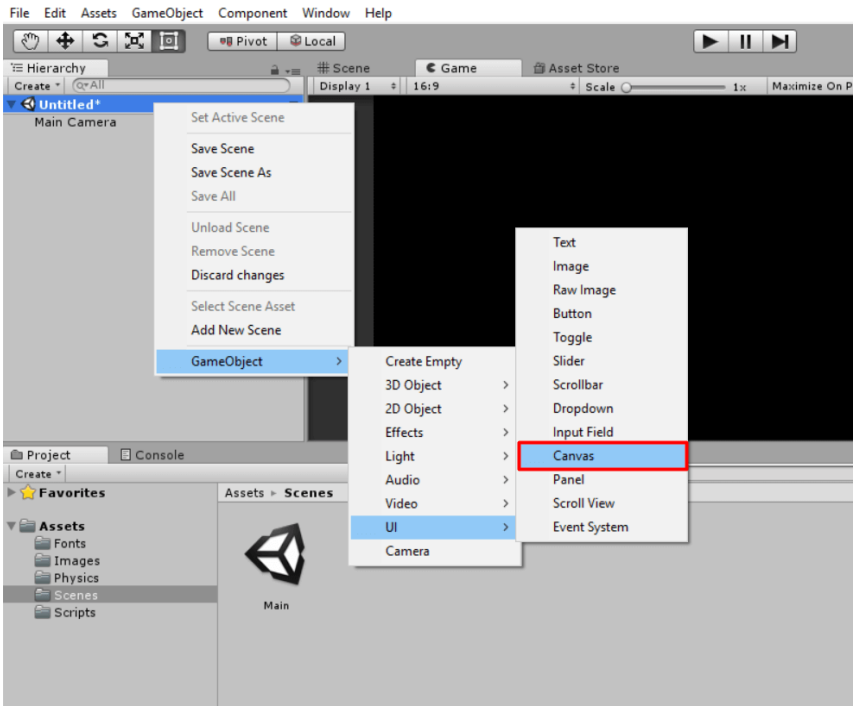
1. Tambahkan Scene baru dengan klik menu **File** > **New Scene**.



2. Ubah warna background pada **Inspector Camera.**



3. Tambahkan UI Canvas (klik kanan pada scene (**Untitled***) pilih **GameObject** > **UI** > **Canvas**).



Pada Inspector Canvas aturlah **UI Scale Mode** menjadi **Scale With Screen Size** dan atur **Reference Resolution** X: **1280** dan Y: **720**

4. Susunlah Home dengan tampilan sebagai berikut:

Tambahkan satu **Text** dan satu **Button** (Pastikan ditambah di dalam Canvas). Detailnya sebagai berikut:

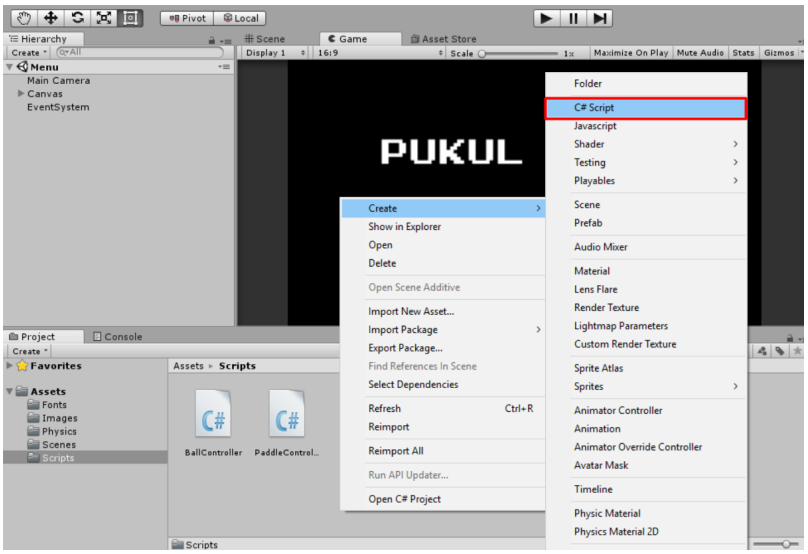
- **Text 1**
    - Nama Object: Judul
    - Text: **Pukul Bola**
    - Font: joystix monospace
    - Font size: 96
    - Alignment: Center - Center
    - Color: Putih
- **Button 1**
    - Nama Object: btnMulai
    - Width: > **400**
    - Height: > **150**
    - Source Image: Persegi
    - Normal Color: Putih
    - Pressed Color: Oranye
- **Text pada Botton 1**
    - Text: **Mulai!**
    - Font: joystix monospace
    - Font Size: 60
    - Alignment: Center-Center
    - Color: Hitam

5. Simpan Scene dengan klik **Ctrl+S** . Cara lainnya, klik Menu **File** > **Save As d**i dalam folder **Scenes** dan beri nama **Menu**.

Praktik: Mengatur Perpindahan Halaman

---

1. Buka Folder **Scripts** kemudian buat Script baru dengan nama **HalamanManager.cs**.



2. Buka Script HalamanManager.cs dengan klik 2 kali lalu tambahkan script berikut:

   - Tambahkan library baru untuk mengelola Scene

```
using UnityEngine.SceneManagement;
```

   Digunakan untuk mengakses method mengelola Scene.

   - Tambahkan variabel di dalam class HalamanManager

```
 public bool isEscapeToExit;
```

   Digunakan untuk menentukan fungsi tombol Escape untuk kembali ke Menu atau ke Main (Gameplay).

   - Tambahkan 2 function ke dalam class HalamanManager.

```
public void MulaiPermainan()
{
    SceneManager.LoadScene("Main");
}
public void KembaliKeMenu()
{
    SceneManager.LoadScene("Menu");
}
```

   Digunakan untuk berpindah ke halaman Main dan ke halaman Menu.

   - Tambahkan kode ini di dalam function **Update.**

```
if (Input.GetKeyUp(KeyCode.Escape))
{
    if (isEscapeToExit)
```

```
        {
            Application.Quit();
        }
        else
        {
            KembaliKeMenu();
        }
    }
```

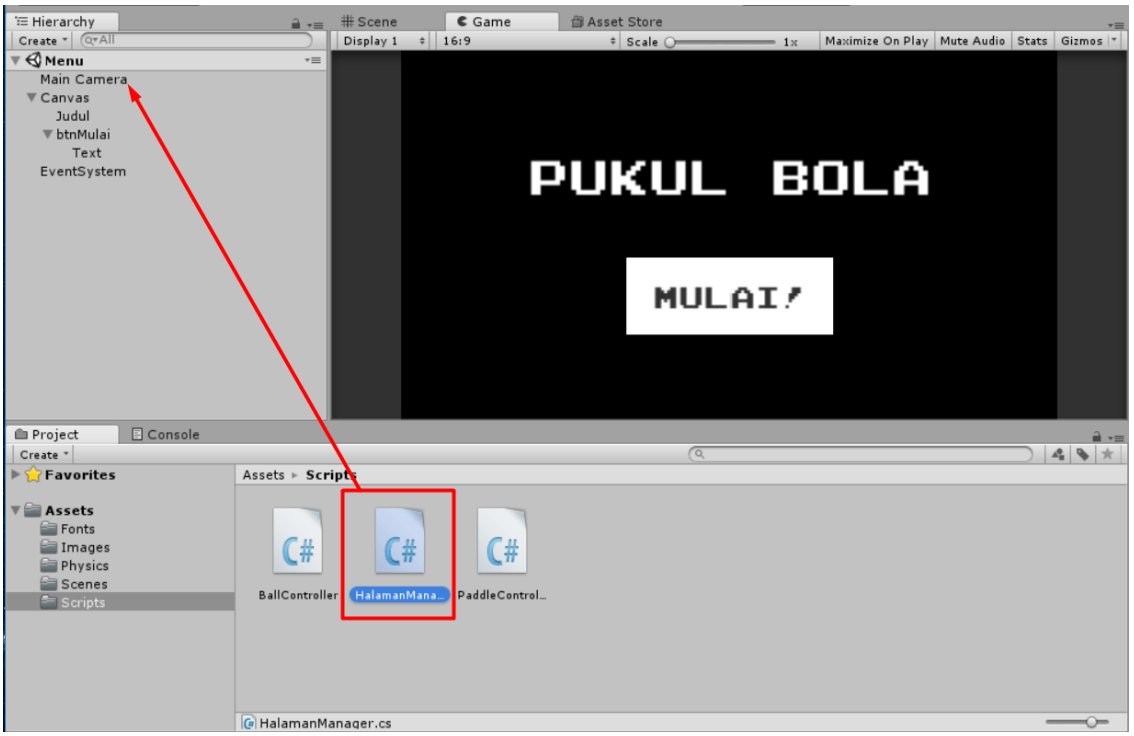
Ketika menekan tombol Escape, jika nilai **isEscapeToExit** bernilai benar maka akan keluar dari Game. Sebaliknya, jika bernilai salah maka akan kembali ke Menu.

2. Simpan perubahan dengan **Ctrl+S sehingga seluruh kode sebagai berikut:**

```
using UnityEngine;
using UnityEngine.SceneManagement;
public class HalamanManager : MonoBehaviour
{
    public bool isEscapeToExit;
    // Use this for initialization
    void Start()
    {
    }
    // Update is called once per frame
    void Update()
    {
        if (Input.GetKeyUp(KeyCode.Escape))
        {
            if (isEscapeToExit)
            {
                Application.Quit();
            }
            else
            {
                KembaliKeMenu();
            }
        }
    }
    public void MulaiPermainan()
    {
        SceneManager.LoadScene("Main");
    }
    public void KembaliKeMenu()
    {
        SceneManager.LoadScene("Menu");
    }
```

```
}
```

3. Masukkan script **HalamanManager.cs** ke GameObject **Main Camera**.

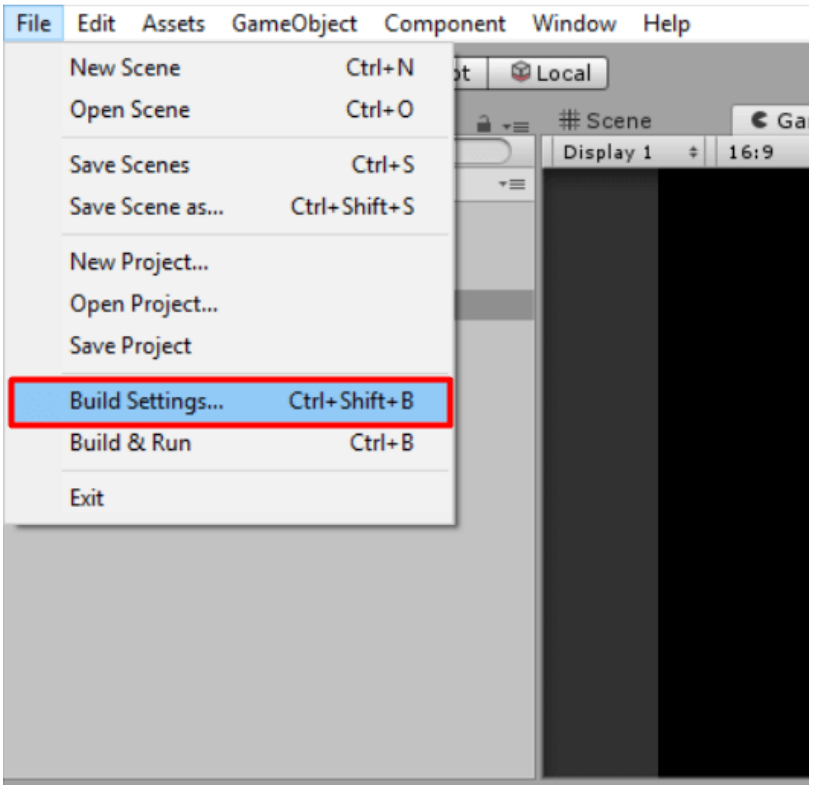


Setelah Anda memasukkan script ke gameobject pada Inspector, centang “Is Escape to Exit.” Alhasil, ketika user menekan tombol escape ia akan keluar dari aplikasi

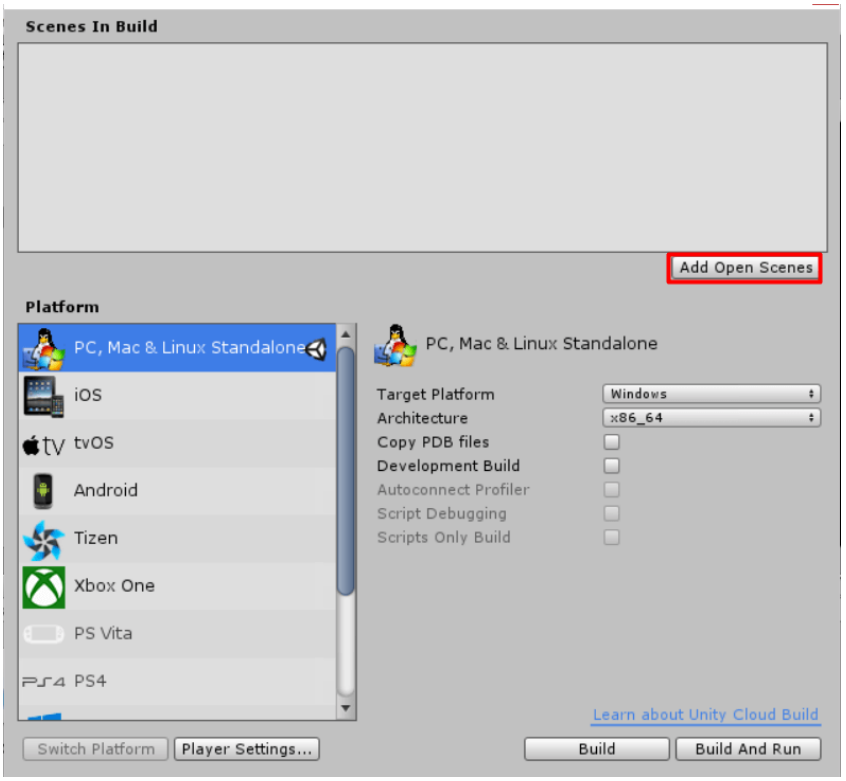
Praktik: Daftarkan Semua Scene ke Scenes In Build

Scene yang telah dibuat tidak dapat langsung diakses. Mengingat kita sudah memiliki 2 Scene, keduanya harus dimasukkan terlebih dulu ke Scenes In Build.

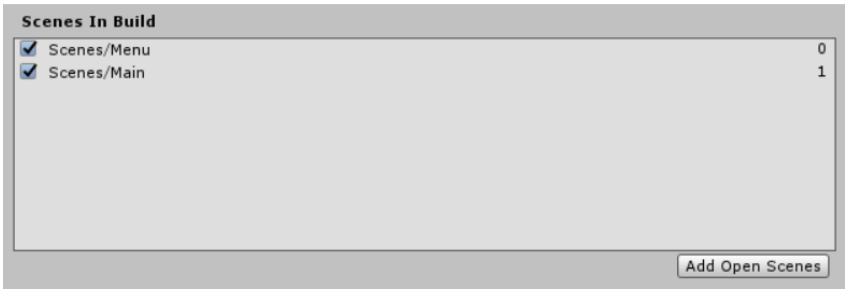
1. Buka **Build Settings...** (menu **File** > **Build Settings...**)



2. Tambahkan scene yang telah dibuka dengan klik **Add Open Scenes**.



Jika Anda memiliki lebih dari 1 scene, maka Anda harus buka scene tersebut kemudian klik **Add Open Scenes**. Sehingga hasilnya seperti berikut:

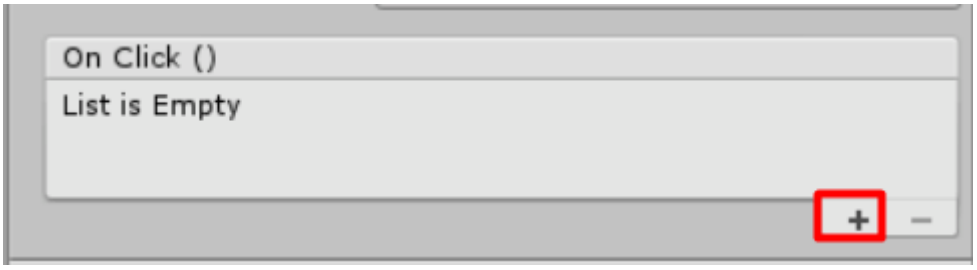


3. Sekarang Scene yang telah dibuat dapat dipanggil melalui script.

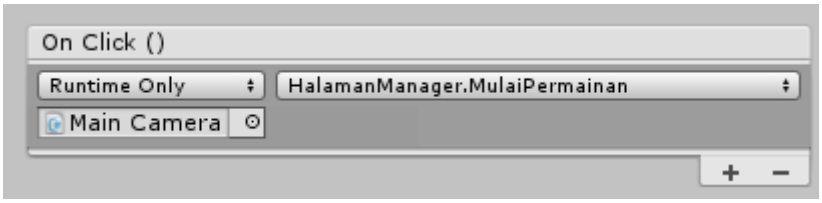
Praktik: Memberi Fungsi pada Button

Setelah script sudah dibuat dan scene sudah siap, selanjutnya hubungkan script yang telah dibuat ke Button (btnMulai). Langkah-langkahnya antara lain:

1. Klik **btnMulai**, kemudian pada komponen **Button** di panel **Inspector**. Klik icon **+** pada On Click().



2. Pada script HalamanManager.cs masukkan gameObject **Main Camera** ke list baru On Click () kemudian masukkan method HalamanManager > MulaiPermainan().



Menambahkan Audio

Menuju bagian akhir dari modul ini, kali ini kita akan belajar menambahkan audio agar game buatan kita lebih hidup. Dalam tutorial ini audio akan digunakan sebagai:

- Background music
- Sound effect ketika bola terpantul

Salah satu tempat yang menyediakan audio gratis adalah **Audio Library – No Copyright Music** (https://www.youtube.com/channel/UCht8qITGkBvXKsR1Byln-wA). Sebagai sampel, sound berikut ini yang akan kita gunakan.

- "Chiptune Shopping Adventures" untuk *background music* (https://www.youtube.com/watch?v=CDEbHETrg9Q).
- "How Now Brown Cow?" untuk *background music* (https://www.youtube.com/watch?v=_3No0WiZE0w).
- "Glass and Metal Collision" untuk *sound effect* (https://www.youtube.com/watch?v=1yr897epApM).
- "Golf Ball Hit" untuk *sound effect* (https://www.youtube.com/watch?v=kMr9BbC69PE).

Silakan gunakan audio lain yang Anda inginkan, tetapi perhatikan lisensinya ya. Jika ingin konversi dari mp4 ke mp3, gunakan Online Video Converter (https://www.onlinevideoconverter.com/mp3-converter).

Setelah mengunduh audio pilihan, buat folder **Audios** serta drag and drop berkas audio tersebut ke folder tersebut.

Pada Unity terdapat beberapa istilah yang berkaitan dengan Audio. Beberapa istilah penting antara lain:

- **Audio Listener**

  Audio Listener merupakan Component yang bertindak seperti microphone. Component ini

menerima semua input dari Audio Source pada sebuah Scene dan memainkan musik melalui speaker komputer.

- **Audio Source**

  Audio Source merupakan Component yang memutar sebuah Audio Clip pada Scene. Efek audio bisa ditambahkan pada Audio Source.

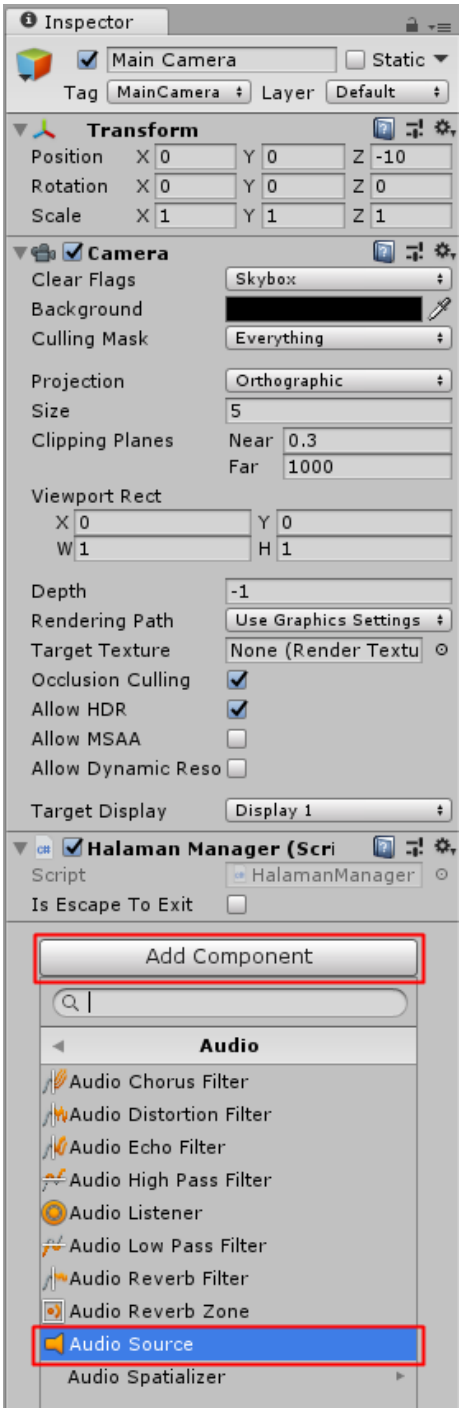
- **Audio Clip**

  Audio Clip berisi data audio yang akan digunakan oleh Audio Source. Format audio yang ada dalam Unity yaitu: .aif, .wav, .mp3, and .ogg.

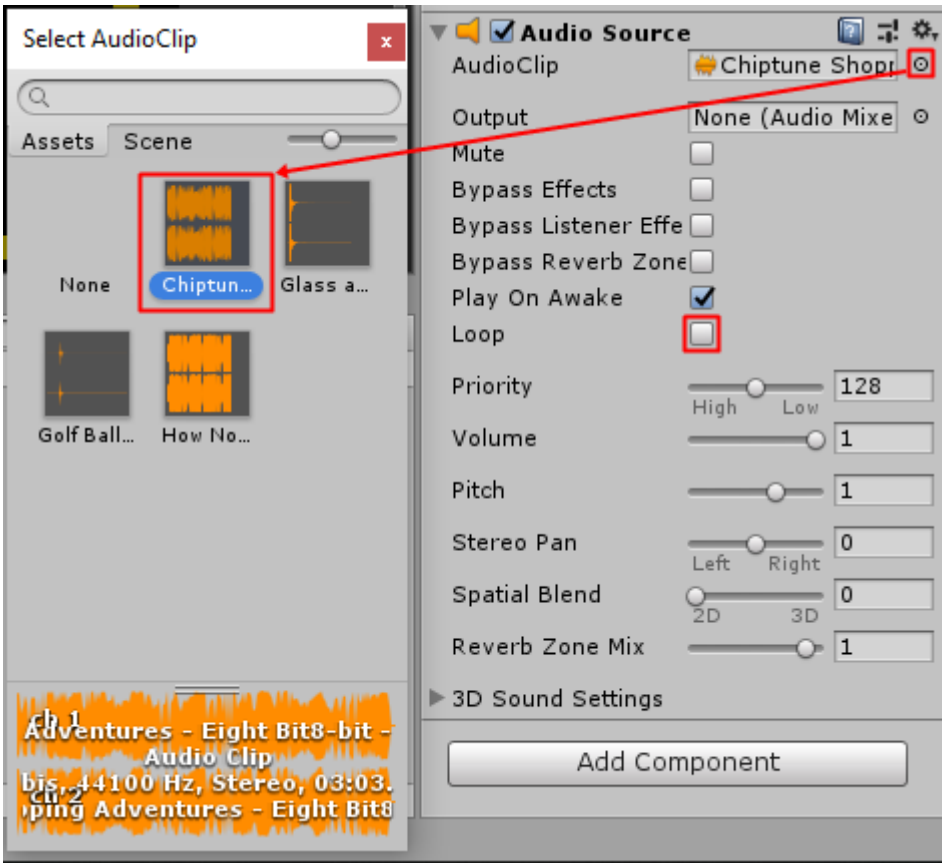
Praktik: Menambahkan Background Music

Untuk menambahkan background music, lakukan tahapan berikut:

1. Buka Scene Main lalu klik **Main Camera** di Hierarchy. Pada Inspector tambahkan **Audio Source** (**Add Component** > **Audio** > **Audio Source**).


Inspector
Main Camera
Static
Tag MainCamera
Layer Default
Transform
Position X 0 Y 0 Z -10
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1
Camera
Clear Flags Skybox
Background
Culling Mask Everything
Projection Orthographic
Size 5
Clipping Planes Near 0.3
Far 1000
Viewport Rect
X 0 Y 0
W 1 H 1
Depth -1
Rendering Path Use Graphics Settings
Target Texture None (Render Textu
Occlusion Culling
Allow HDR
Allow MSAA
Allow Dynamic Reso
Target Display Display 1
Halaman Manager (Scri
Script HalamanManager
Is Escape To Exit
Add Component
Audio
Audio Chorus Filter
Audio Distortion Filter
Audio Echo Filter
Audio High Pass Filter
Audio Listener
Audio Low Pass Filter
Audio Reverb Filter
Audio Reverb Zone
Audio Source
Audio Spatializer

2. Masukkan berkas "**Chiptune Shopping Adventures.mp3**" ke field **AudioClip.** Jangan lupa untuk centang **Loop**.

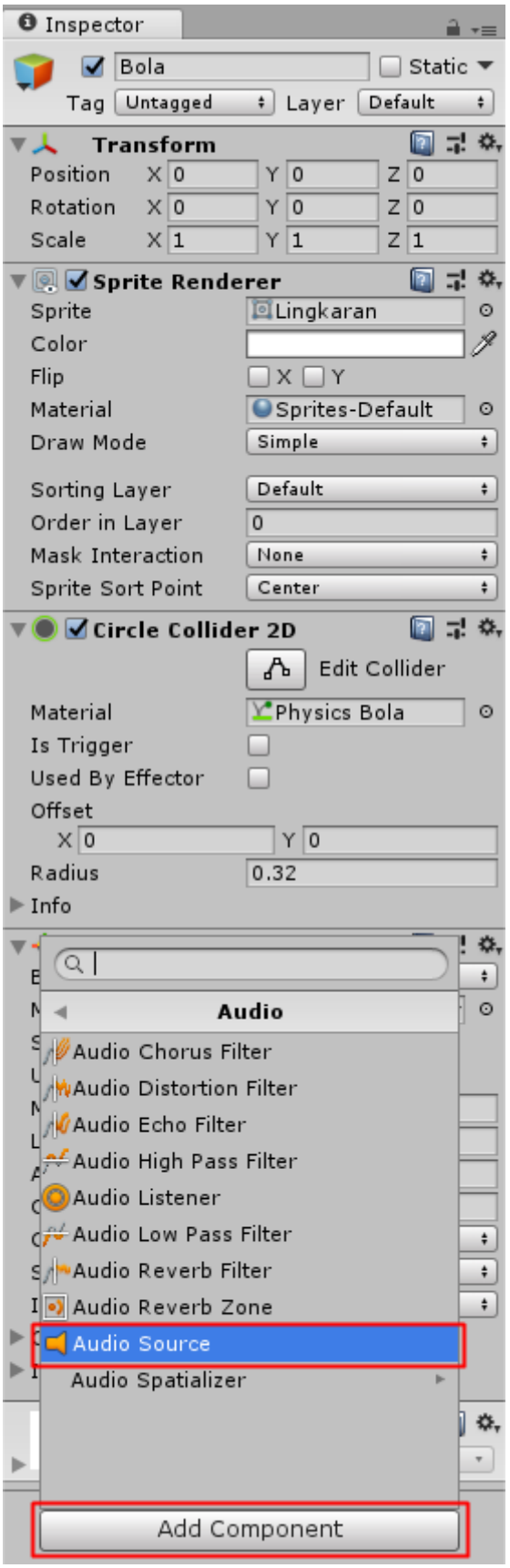


Ketika dimainkan, musik langsung jalan dan secara otomatis mengulang ketika selesai. Anda juga dapat mengatur Volume untuk menyesuaikan kenyamanan dalam game yang Anda buat.

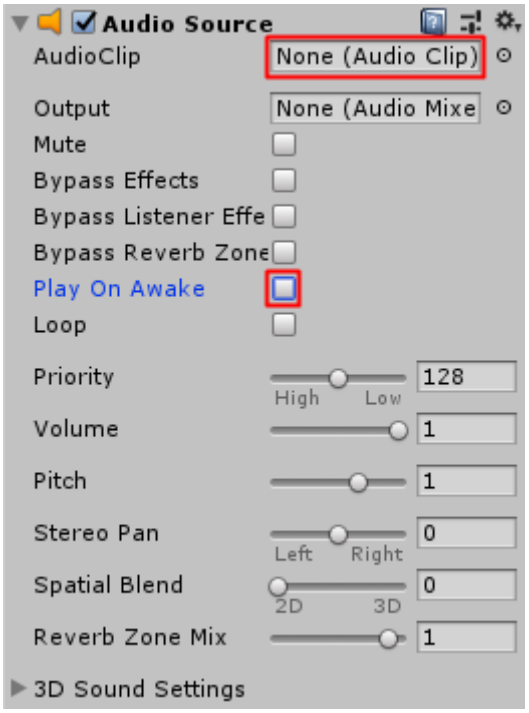
Praktik: Menambahkan Sound Effect

Penambahan sound effect akan dicontohkan dengan momen ketika Bola menabrak benda lain. Langkah-langkah yang diperlukan cukup mirip dengan penambahan *background music*, yaitu:

1. Klik GameObject **Bola** di Hierarchy. Pada Inspector tambahkan **Audio Source (Add Component > Audio > Audio Source)**.



Biarkan field **AudioClip** tetap kosong**.** Lalu hilangkan centang pada **Play On Awake.**



2. Selanjutnya, agar sound effect dapat diputar pada waktu yang sesuai, buka script **BolaController.cs** dan tambahkan potongan kode berikut.

   - Pada inisialisasi **variabel** tambahkan:

```
1. AudioSource audio;
2. public AudioClip hitSound;
```

AudioSource audio; --> Sebagai kontroller audio.

public AudioClip hitSound; --> Menyimpan berkas audio.

   - Pada prosedur **Start()** tambahkan:

```
1. audio = GetComponent<AudioSource>();
```

Untuk "mengambil" komponen Audio Source.

   - Pada prosedur **OnCollisionEnter2D** tambahkan:

```
1. audio.PlayOneShot(hitSound);
```

Menjalankan sebuah audio.

2. Simpan perubahan dengan tekan **Ctrl+S** sehingga seluruh kode seperti in**i:**

```
2. using UnityEngine;
```

```
3.  using UnityEngine.UI;
4.
5.  public class BallController : MonoBehaviour
6.  {
7.      public int force;
8.      Rigidbody2D rigid;
9.      int scoreP1;
10.     int scoreP2;
11.     Text scoreUIP1;
12.     Text scoreUIP2;
13.     GameObject panelSelesai;
14.     Text txPemenang;
15.     AudioSource audio;
16.     public AudioClip hitSound;
17.
18.     // Use this for initialization
19.     void Start()
20.     {
21.         rigid = GetComponent<Rigidbody2D>();
22.         Vector2 arah = new Vector2(2, 0).normalized;
23.         rigid.AddForce(arah * force);
24.         scoreP1 = 0;
25.         scoreP2 = 0;
26.         scoreUIP1 = GameObject.Find("Score1").GetComponent<Text>();
27.         scoreUIP2 = GameObject.Find("Score2").GetComponent<Text>();
28.         panelSelesai = GameObject.Find("PanelSelesai");
29.         panelSelesai.SetActive(false);
30.         audio = GetComponent<AudioSource>();
31.     }
32.
33.     // Update is called once per frame
34.     void Update()
35.     {
36.
37.     }
38.
39.     private void OnCollisionEnter2D(Collision2D coll)
40.     {
41.         audio.PlayOneShot(hitSound);
42.         if (coll.gameObject.name == "TepiKanan")
43.         {
44.             scoreP1 += 1;
45.             TampilkanScore();
46.             if (scoreP1 == 5)
```

```
47.             {
48.                 panelSelesai.SetActive(true);
49.                 txPemenang = GameObject.Find("Pemenang").GetComponent<Text>();
50.                 txPemenang.text = "Player Biru Pemenang!";
51.                 Destroy(gameObject);
52.                 return;
53.             }
54.             ResetBall();
55.             Vector2 arah = new Vector2(2, 0).normalized;
56.             rigid.AddForce(arah * force);
57.         }
58.         if (coll.gameObject.name == "TepiKiri")
59.         {
60.             scoreP2 += 1;
61.             TampilkanScore();
62.             if (scoreP2 == 5)
63.             {
64.                 panelSelesai.SetActive(true);
65.                 txPemenang = GameObject.Find("Pemenang").GetComponent<Text>();
66.                 txPemenang.text = "Player Merah Pemenang!";
67.                 Destroy(gameObject);
68.                 return;
69.             }
70.             ResetBall();
71.             Vector2 arah = new Vector2(-2, 0).normalized;
72.             rigid.AddForce(arah * force);
73.         }
74.         if (coll.gameObject.name == "Pemukul1" || coll.gameObject.name == "Pemukul2")
75.         {
76.             float sudut = (transform.position.y - coll.transform.position.y) * 5f;
77.             Vector2 arah = new Vector2(rigid.velocity.x, sudut).normalized;
78.             rigid.velocity = new Vector2(0, 0);
79.             rigid.AddForce(arah * force * 2);
80.         }
81.     }
82.
83.     void ResetBall()
84.     {
85.         transform.localPosition = new Vector2(0, 0);
86.         rigid.velocity = new Vector2(0, 0);
87.     }
88.
89.     void TampilkanScore()
90.     {
```

```
91.         Debug.Log("Score P1: " + scoreP1 + " Score P2: " + scoreP2);
92.         scoreUIP1.text = scoreP1 + "";
93.         scoreUIP2.text = scoreP2 + "";
94.     }
95. }
```

2. Terakhir buka Inspector Bola. Lalu masukkan berkas“**Glass and Metal Collision.mp3**” ke field **Hit Sound** Ball Controller (Script).