

Laporan Resmi



WORKSHOP PRODUKSI GAME 3

Membuat Game Space Shooter menggunakan Unity Engine 3D

Disusun oleh :

Muhammad Barkah Rahmanu Mauludi 4210171022

**PROGRAM STUDI TEKNOLOGI GAME
DEPARTEMEN TEKNOLOGI MULTIMEDIA KREATIF
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
SURABAYA**

2018

- **Dasar Teori**

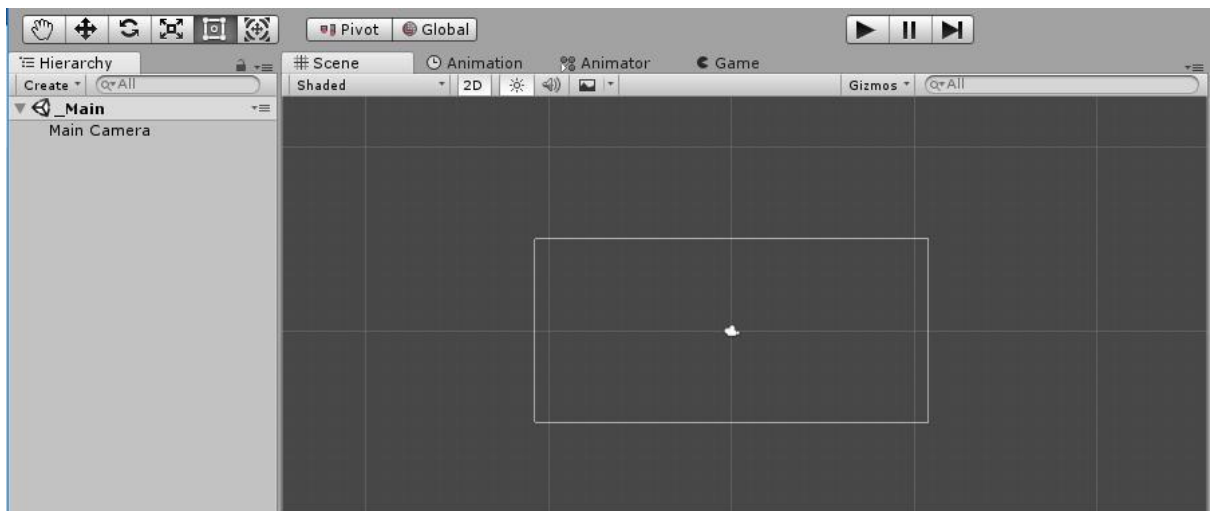
Bahasa C++ adalah bahasa pemrograman komputer yang merupakan pengembangan dari bahasa C yang disempurnakan oleh Bjarne Stroustrup pada tahun 1980 menjadi C dengan kelas (C With Classes) dan berganti nama pada tahun 1983 menjadi C++. awalnya C++ disebut "a better C" kemudian berganti nama pada tahun 1983 menjadi C++. Symbol ++ merupakan operator C untuk operasi increment (penaikan), digunakan untuk menunjukkan bahwa bahasa baru ini merupakan versi yang lebih canggih dari C.

Unity adalah Engine (Sebutan Untuk software pengembang game) yang dapat digunakan untuk membuat game di ruang 3 Dimensi. Unity menggunakan Metode Assets Workflow yaitu Dia menggunakan Aset-Aset dalam merangkai sebuah game.

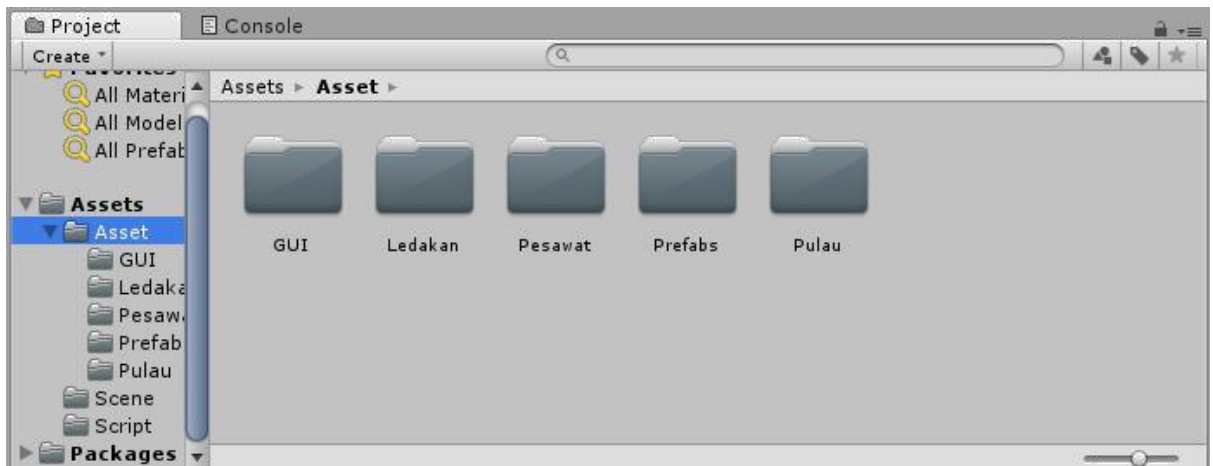
- **Alat dan Bahan**

- a. Modul Workshop Produksi Game 3 (Space Shooter with Unity Engine 3D)
- b. Laptop/PC
- c. Game Engine Unity

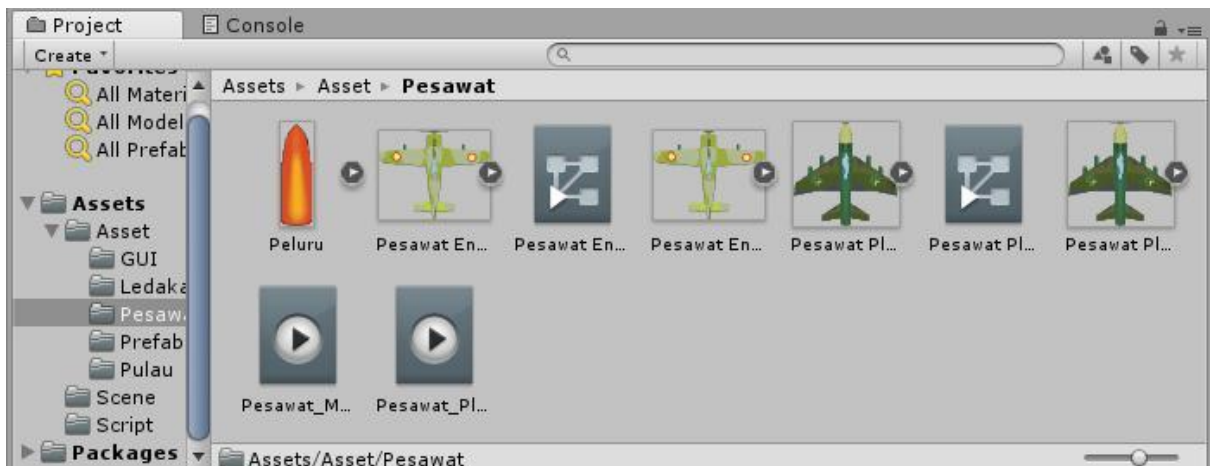
- **Analisa Pengembangan dan Penjelasan Per-bagian**



Mula-mula buatlah Scene baru bernama _Main, scene tersebut merupakan scene dimana kita akan meletakkan seluruh komponen yang dibutuhkan pada saat pembuatan game.



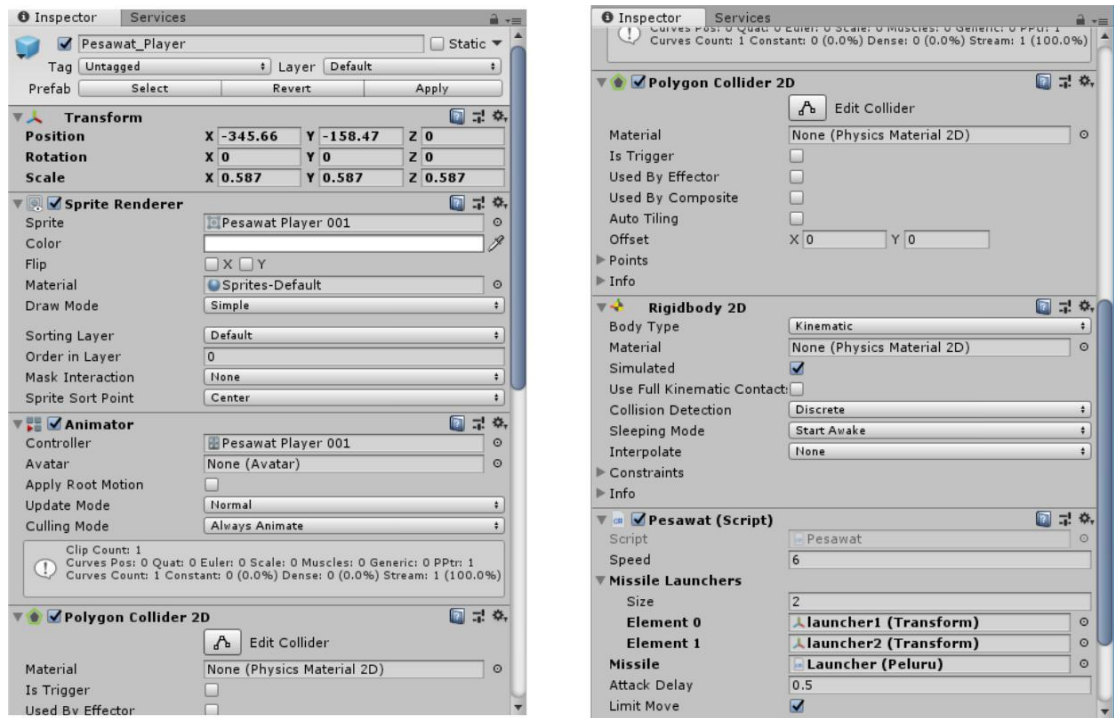
Pada folder Asset silahkan buat 5 folder yaitu GUI, Ledakan, Pesawat, Prefabs dan Pulau, folder tersebut digunakan untuk meletakkan masing-masing Asset sesuai dengan nama dan fungsi pada game tersebut



Contohnya pada folder Pesawat , dapat diisi dengan Asset yang berhubungan dengan pesawat, seperti peluru, pesawat musuh, pesawat player dan Animasi dari 2 Pesawat tersebut



Masukkan semua Asset Pesawat player, Pesawat musuh, Pulau , Background Air dan Peluru pada Scene tersebut.



Pada Inspector Pesawat_player terdapat Transform yang digunakan untuk melihat dengan detail bagaimana posisi, besar dan rotasi pada object tersebut. Lalu Sprite Render digunakan untuk mengatur pada layer berapa Pesawat_player diletakkan, di inputkan 0 agar Pesawat_player berada di atas object Background Air, Lalu Animator digunakan untuk menganimasikan Pesawat_Player. Lalu Polygon Collider 2D digunakan untuk memberikan collision dengan jenis Polygon agar mudah disesuaikan dengan Asset yang dimasukkan. Lalu RigidBody 2D memberikan Efek Kinematic agar object dapat bergerak secara kinematis. Lalu menambahkan Script Pesawat yang berisi pengaturan Speed dari pesawat dan Child dari peluru bernama Launcher, dan Attack delay berguna untuk memberikan jeda waktu kepada pesawat untuk melesatkan peluru selanjutnya.

```
Pesawat.cs [X]
Assembly-CSharp Pesawat

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Pesawat : MonoBehaviour {
6
7      public float speed = 10;
8      public Transform[] missileLaunchers;
9      public Peluru missile;
10     public float attackDelay = 0.5f;
11     public bool limitMove = true;
12     private Vector2 minPosition, maxPosition;
13     private float attackDelayCounter;
14
15     void Start()
16     {
17         minPosition = Camera.main.ScreenToWorldPoint(new Vector2(0, 0));
18         maxPosition = Camera.main.ScreenToWorldPoint(new
19             Vector2(Screen.width, Screen.height));
20     }
21     void Update()
22     {
```

```
Pesawat.cs [X]
Assembly-CSharp Pesawat

23         attackDelayCounter -= Time.deltaTime;
24     }
25
26     public void MoveUp()
27     {
28         if (limitMove)
29         {
30             Vector2 newPosition = (Vector2)this.transform.position +
31                 Vector2.up * speed * Time.deltaTime;
32             if (newPosition.y < maxPosition.y)
33             {
34                 this.transform.position = newPosition;
35             }
36         }
37         else
38         {
39             this.transform.Translate(Vector2.up * speed *
40                 Time.deltaTime);
41         }
42     }
43     public void MoveDown()
44     {
```

```
Pesawat.cs [X]
Assembly-CSharp Pesawat

45     if (limitMove)
46     {
47         Vector2 newPosition = (Vector2)this.transform.position -
48         Vector2.up * speed * Time.deltaTime;
49         if (newPosition.y > minPosition.y)
50         {
51             this.transform.position = newPosition;
52         }
53     }
54 }
55 public void MoveLeft()
56 {
57     if (limitMove)
58     {
59         Vector2 newPosition = (Vector2)this.transform.position -
60         Vector2.right * speed * Time.deltaTime;
61         if (newPosition.x > minPosition.x)
62         {
63             this.transform.position = newPosition;
64         }
65     }
66 }
```

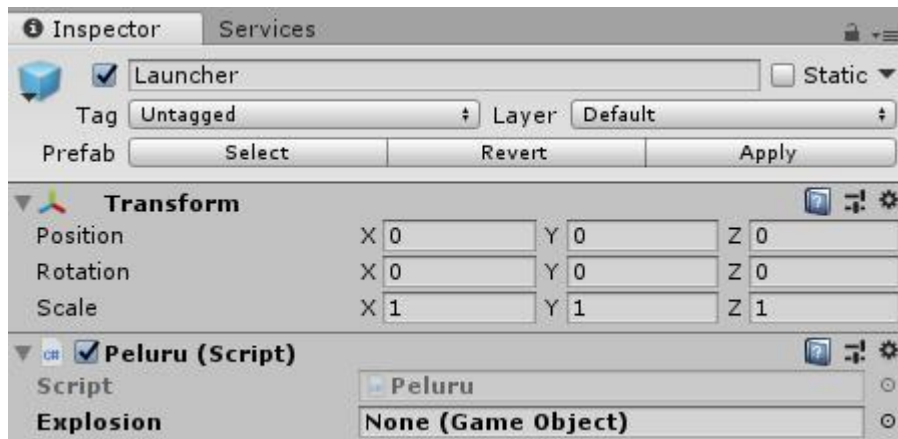
100 %

```
Pesawat.cs [X]
Assembly-CSharp Pesawat

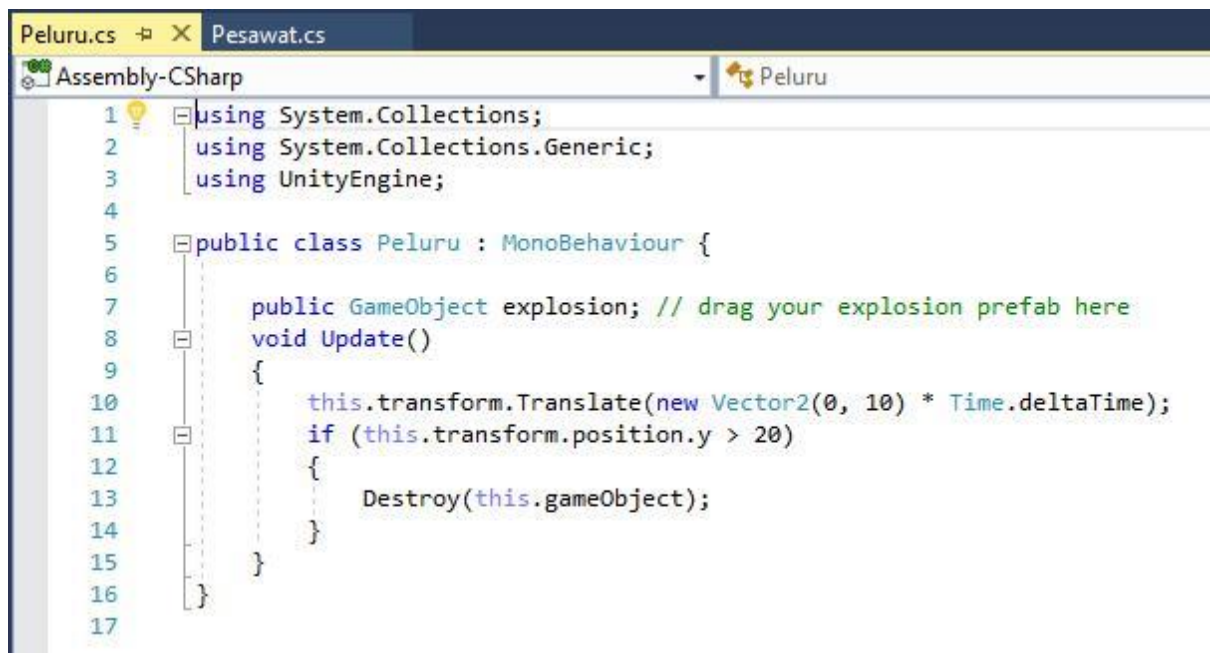
67 public void MoveRight()
68 {
69     if (limitMove)
70     {
71         Vector2 newPosition = (Vector2)this.transform.position +
72         Vector2.right * speed * Time.deltaTime;
73         if (newPosition.x < maxPosition.x)
74         {
75             this.transform.position = newPosition;
76         }
77     }
78 }
79 public void Attack()
80 {
81     if (attackDellayCounter <= 0)
82     {
83         foreach (Transform missileLauncher in missileLaunchers)
84         {
85             Instantiate(missile, missileLauncher.position,
86             missileLauncher.rotation);
87         }
88         attackDellayCounter = attackDelay;
89     }
90 }
```

100 %

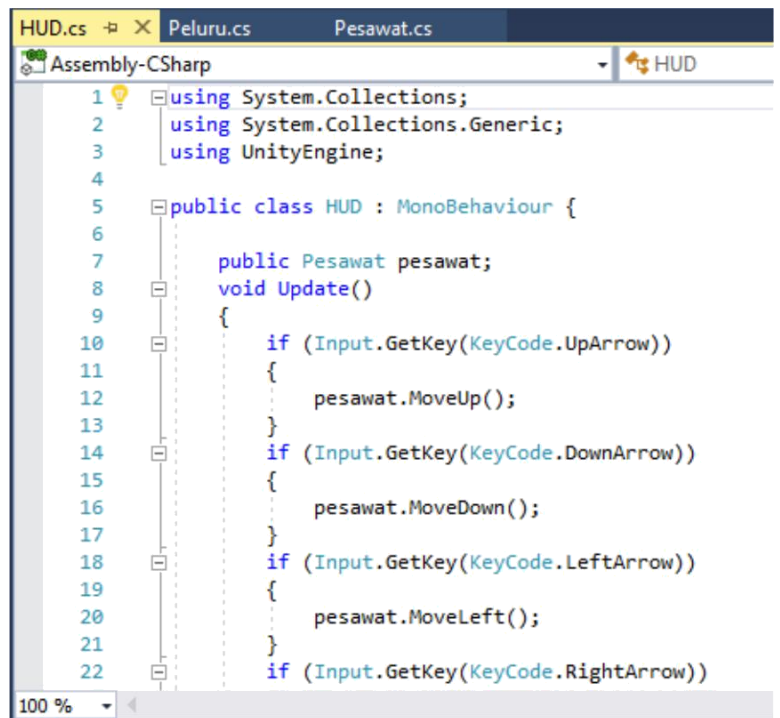
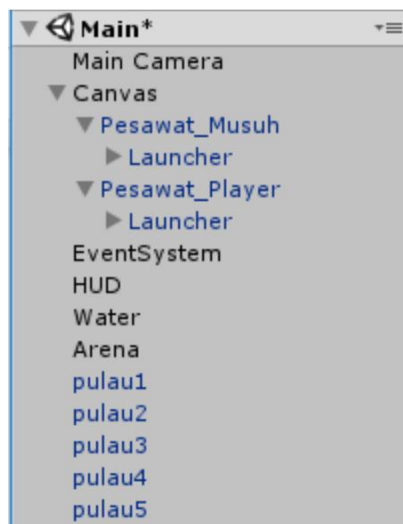
Gambar diatas merupakan isi dari Script Player yang diletakkan pada Object Pesawat_player , Script diatas diibaratkan sebagai mesin pada pesawat sehingga apabila tidak ada Script di atas pada object Pesawat_player maka semua fungsi tidak akan dijalankan dan pesawat tidak dapat melakukan apapun.



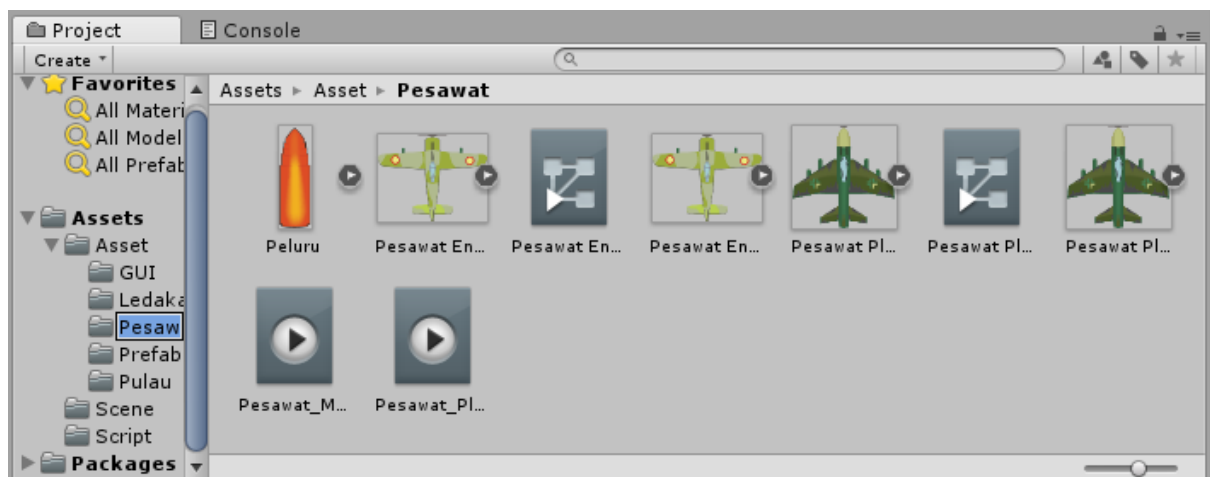
Lalu pada object Launcher hanya diisi oleh Script Peluru.



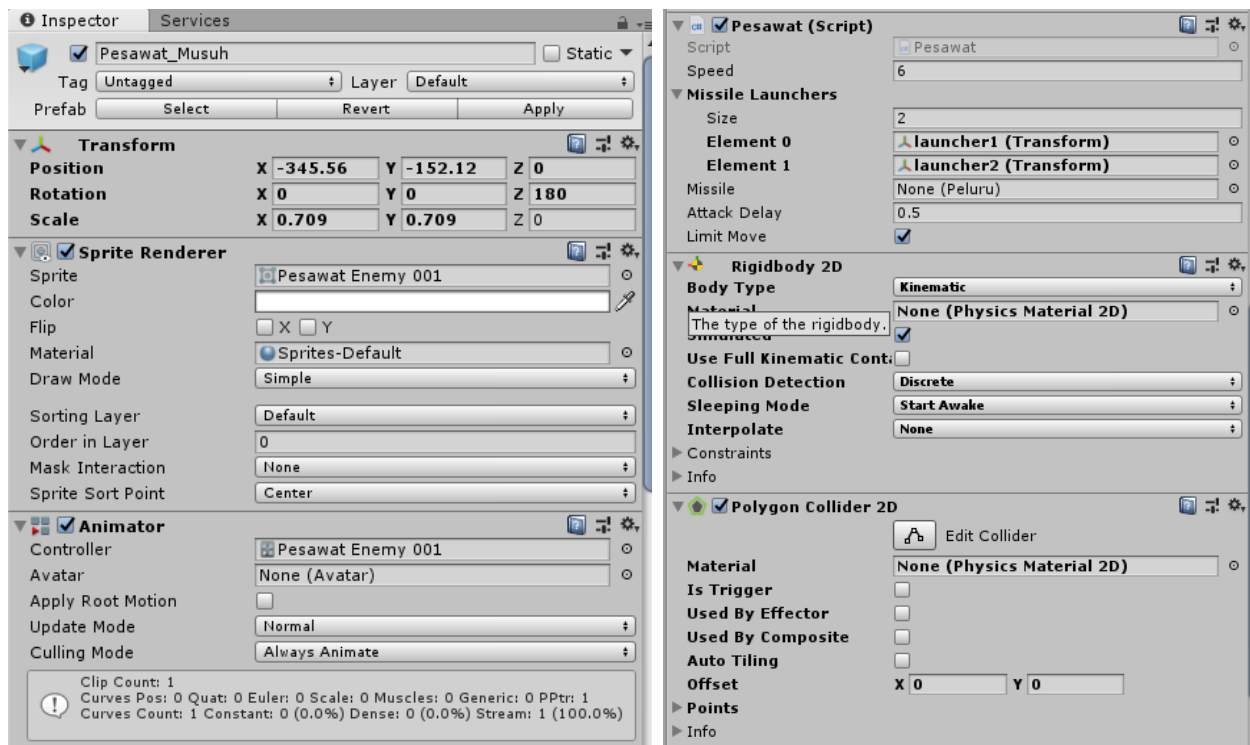
Gambar diatas merupakan isi dari Script Peluru yang berisi pergerakan dan kondisi apabila peluru bertabrakan dengan object lain maka object peluru akan menghilang



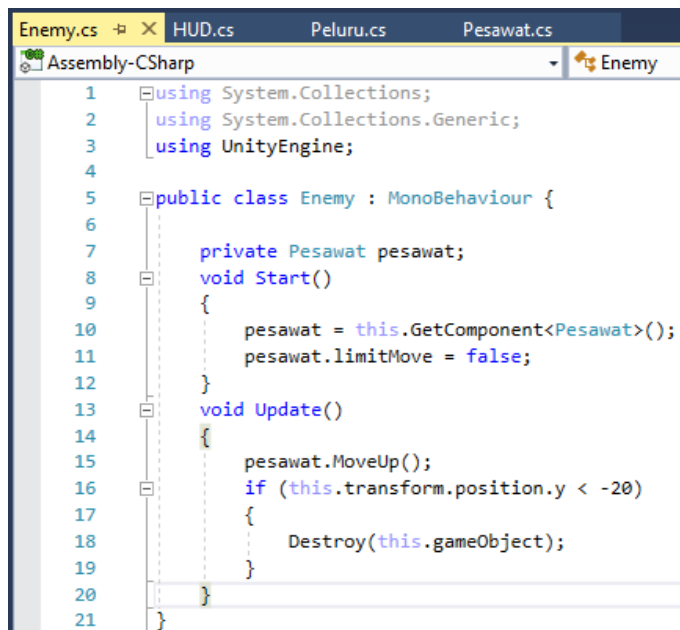
Lalu pada Hierarchy buatlah Object bernama HUD yang digunakan untuk input kepada pesawat tersebut , dengan memasukkan Script di atas pada HUD Pesawat sudah bisa digerakkan sesuai input yang ada. Player bergerak dengan 4 arah dengan menggunakan Arrow pada keyboard sesuai dengan fungsinya, dan Z untuk menembakkan peluru.



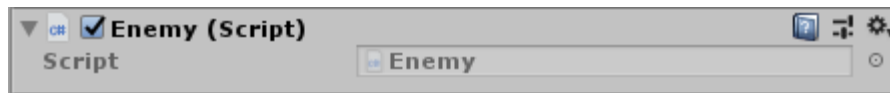
Tambahkan Asset Pesawat musuh dan tambahkan juga Animasi pada pesawat musuh seperti pada Pesawat Player



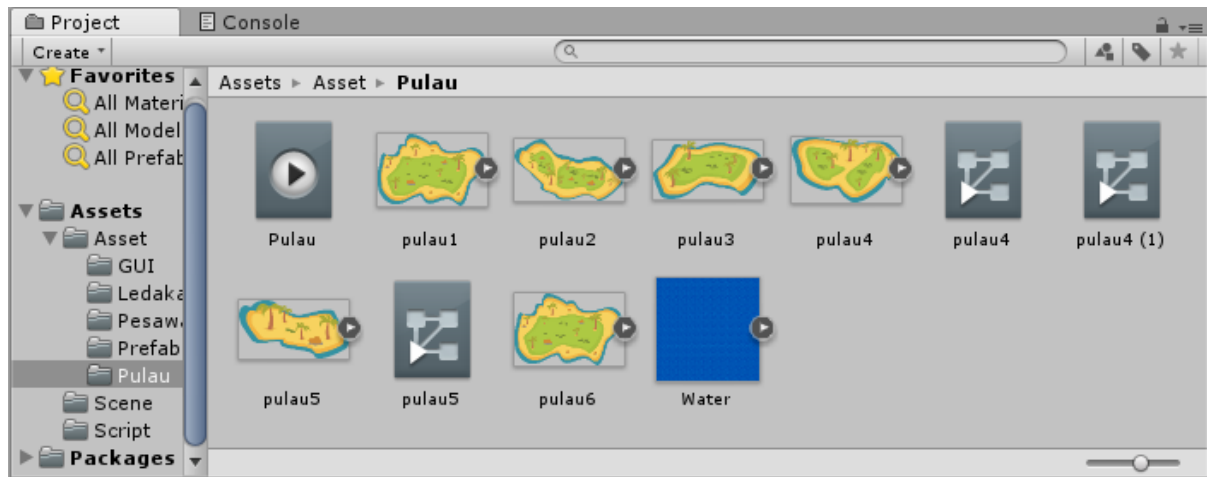
Sehingga pada Inspect Pesawat_musuh terdapat Animator lalu Script Pesawat yang disamakan dengan Pesawat_Player.



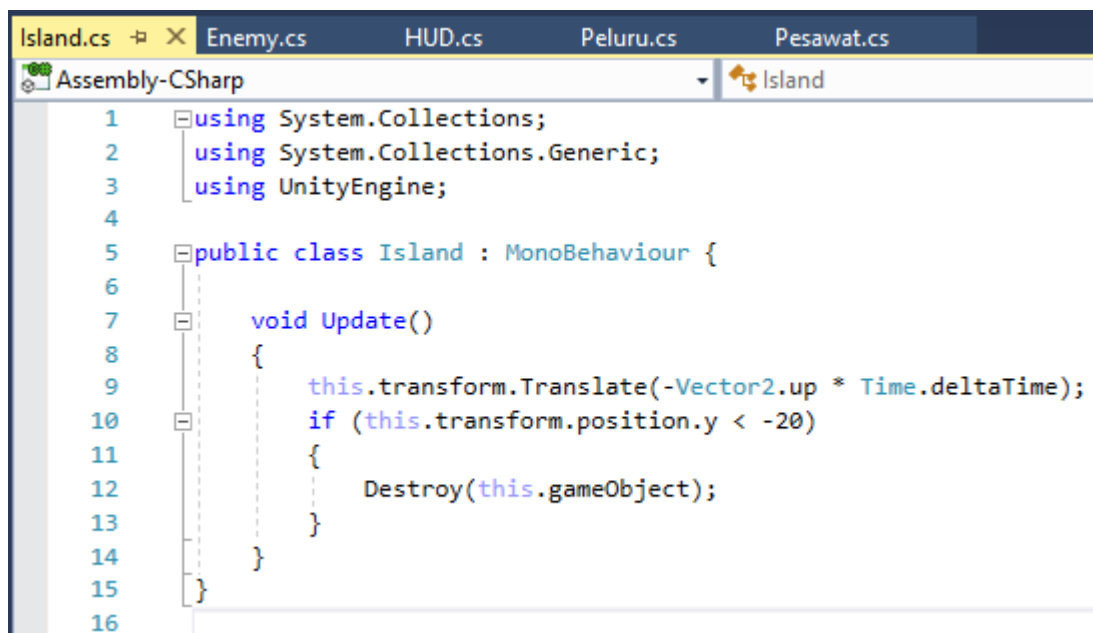
Lalu buatlah Script Enemy.cs yang berisi apabila Pesawat_Musuh melebihi layar dengan pengaturan Transform Position.y < 20 maka object akan dihancurkan, agar tidak menyimpan banyak data pada Game.



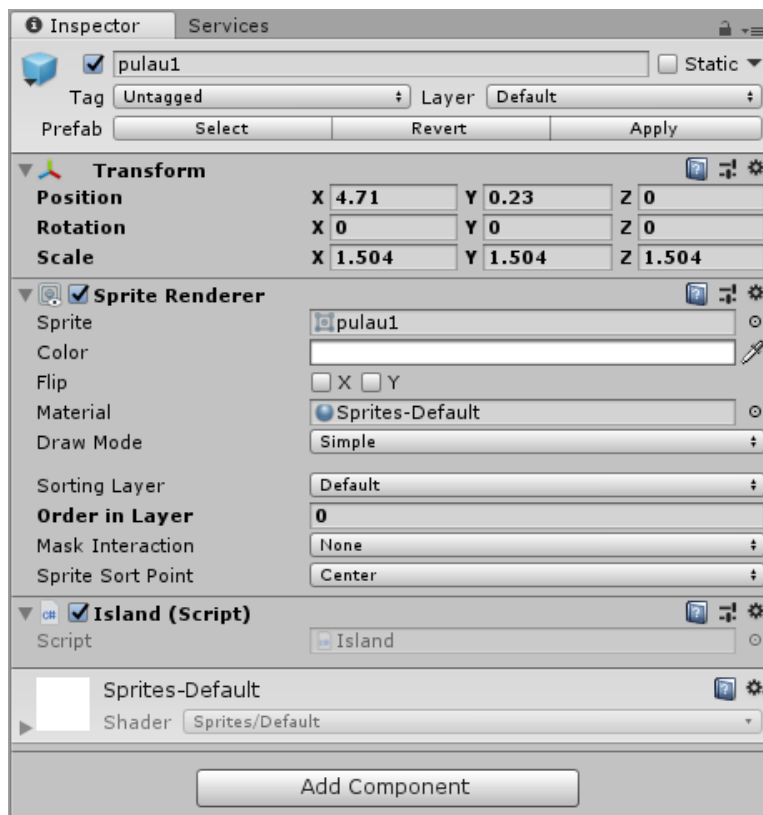
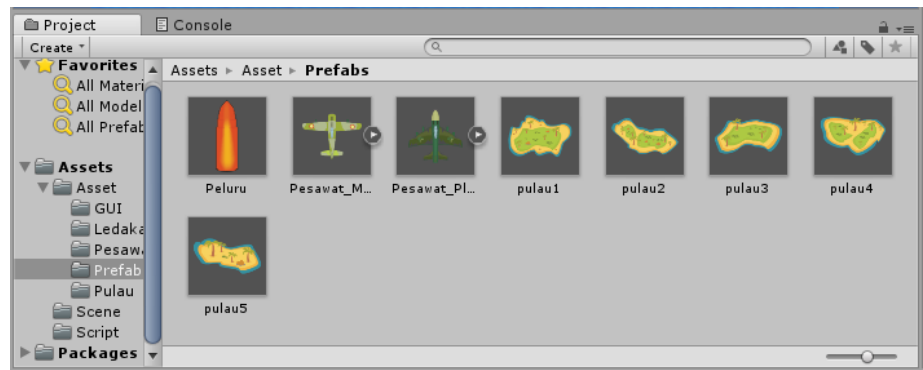
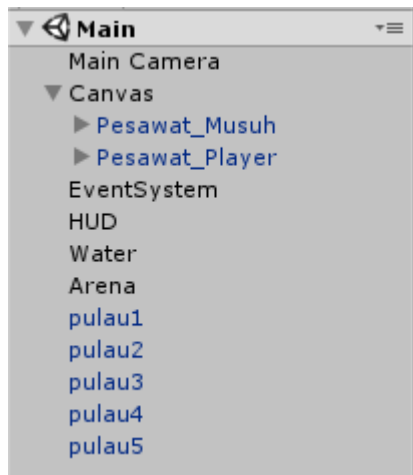
Lalu masukan Script Enemy yang telah kita buat menuju Object Pesawat_Musuh seperti gambar yang ada di atas



Masukkan Sprite Pulau dan Air pada Unity agar dapat digunakan pada game dan dimasukkan pada Hierarchy



Buatlah Script Island.cs yang berisi fungsi apabila pulau telah melebihi UI pada game maka otomatis object akan dihancurkan agar tidak membebani game.



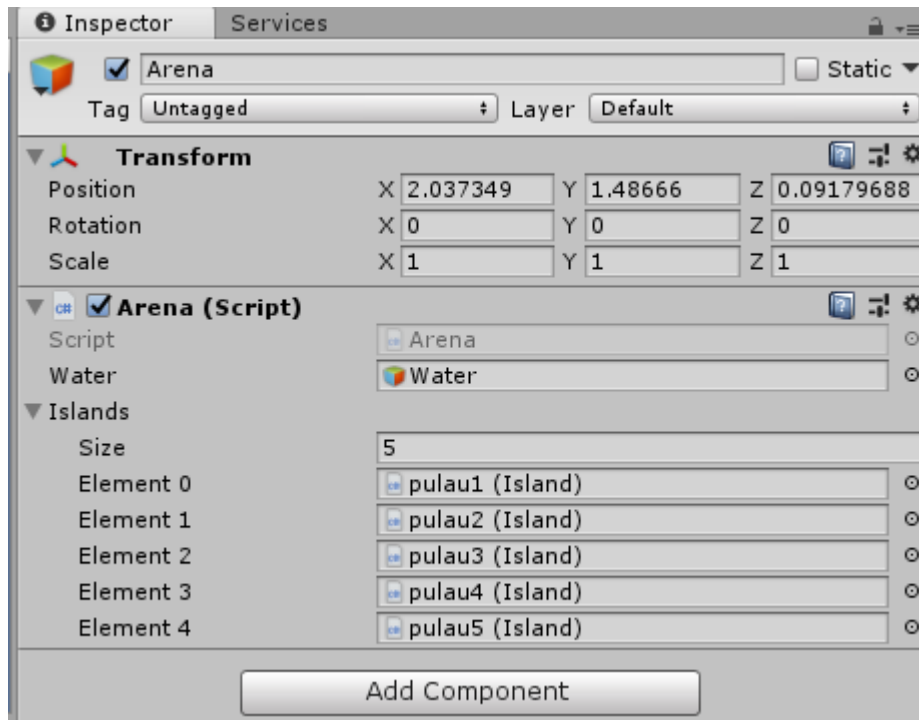
Lalu Prefabkan object pulau dengan cara mendrag sprite pulau menuju hierarchy lalu atur posisi dan tambahkan Script Island, lalu pada Assets buatlah 1 folder baru bernama Prefabs, drag object pulau1 hingga 5 menuju folder Prefabs maka otomatis object akan di Prefabs kan, dan akan berwarna biru seperti yang ada pada gambar diatas.

```

Arena.cs  Island.cs  Enemy.cs  HUD.cs  Peluru.cs  Pesawat.cs
Assembly-CSharp  Arena
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class Arena : MonoBehaviour
6  {
7      public GameObject water;
8      public Island[] islands;
9      private float generateIslandDelayCount;
10     private Vector2 minPosition, maxPosition;
11     void Start()
12     {
13         minPosition = Camera.main.ScreenToWorldPoint(new Vector2(0, 0));
14         maxPosition = Camera.main.ScreenToWorldPoint(new Vector2(Screen.width, Screen.height));
15         float waterPosX = minPosition.x, waterPosY = minPosition.y;
16         SpriteRenderer instWater = (Instantiate(water, new
17         Vector2(waterPosX, waterPosY), Quaternion.identity) as
18         GameObject).GetComponent<SpriteRenderer>();
19         instWater.transform.parent = this.transform;
20         while (waterPosY - 2 * instWater.sprite.bounds.max.y <
21             maxPosition.y)
22     {
23         waterPosX = minPosition.x;
24         while (waterPosX - 2 * instWater.sprite.bounds.max.x <
25             maxPosition.x)
26         {
27             instWater = (Instantiate(water, new
28             Vector2(waterPosX, waterPosY), Quaternion.identity) as
29             GameObject).GetComponent<SpriteRenderer>();
30             instWater.transform.parent = this.transform;
31             waterPosX += 2 * instWater.sprite.bounds.max.x;
32         }
33         waterPosY += 2 * instWater.sprite.bounds.max.y;
34     }
35     }
36     void Update()
37     {
38         generateIslandDelayCount -= Time.deltaTime;
39         if (generateIslandDelayCount <= 0)
40         {
41             Instantiate(islands[Random.Range(0, islands.Length - 1)], new
42             Vector2(Random.Range(minPosition.x, maxPosition.x), 20),
43             Quaternion.identity);
44             generateIslandDelayCount = Random.Range(5, 15);
45         }
46     }
47 }
48

```

Buatlah Script Arena.cs yang digunakan untuk menggenerate otomatis Pulau dan Air agar tetap berada pada UI game, dan untuk pulau agar bergerak secara vertical dengan kecepatan yang telah ditentukan pada game.



Buatlah Object baru pada Unity dan beri nama Arena dan masukkan Script Arena didalamnya, maka script akan meminta object Water dan 5 Object pulau yang telah diprefabs agar pulau dapat generate secara Otomatis.



Lakukan Playtest pada Game tersebut agar kita mengetahui bahwa yang telah kita kerjakan sesuai dengan apa yang kita inginkan.

- **Kesimpulan**

Pada pembuatan game Space Shooter ini, akan mudah diaplikasikan dengan menggunakan Unity Engine 3D, karena pada Unity Engine 3D sudah disediakan beberapa fitur yang mempermudah pengguna dari segi Script, Asset dan Animasi.