

LAPORAN PRAKTIKUM PEMROGRAMAN VISUAL



Disusun oleh Group 4:

Muhammad Marshal Putra Al Barry (24)
Alfito Endriyanto (15)
Sendi Vellya Destiani (03)
Zahwa Olivia Marshanti (09)
Zaenal Anshori (21)
Bryan Royvaldo Al Assyura (19)

Dosen Pengampu :

Evianita Dwi Fajrianti, S.Tr.T., M.Tr.T., Ph.D

**POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
DEPARTEMEN TEKNOLOGI MULTIMEDIA KREATIF
TEKNOLOGI MULTIMEDIA BROADCASTING KAMPUS
LAMONGAN
2025**

Laporan Praktikum Mission: Player Control (Unity Learn - Junior Programmer)

I. PENDAHULUAN

1. Unity Learn & Pathways Junior Programmer :

Unity Learn adalah platform pembelajaran resmi dari Unity yang menyediakan berbagai kursus dan pathways untuk membantu siapa saja belajar membuat game dan aplikasi interaktif menggunakan Unity. Salah satu jalur pembelajaran (pathway) yang paling utama adalah Junior Programmer ditujukan bagi pemula yang ingin menguasai dasar-dasar pemrograman C# serta proses pembuatan game berbasis Unity tanpa harus memiliki latar belakang matematika atau pengalaman pemrograman sebelumnya. Setiap Pathway diakhiri dengan demonstrasi kesiapan kerja (Job-Ready) dan rencana digital yang dapat dibagikan.

2. Tujuan dan Rangkaian Misi Control :

Misi "Player Control" adalah bagian awal dari pathway Junior Programmer yang bertujuan memperkenalkan konsep dasar pemrograman, pembuatan skrip C#, dan penggunaan variabel serta fungsi di Unity. Melalui serangkaian Misi dari Lesson 1.1 Start Your 3D Engines, Lesson 1.2 Pedal to the Metal, hingga Challenge 1 – Plane Programming, mahasiswa berhasil membuat proyek game sederhana di mana objek pemain dapat dikendalikan secara dinamis. Hasil pengujian akan menunjukkan bahwa kontrol pemain dapat berjalan dengan baik sesuai dengan parameter kecepatan dan rotasi yang diatur dalam kode program.

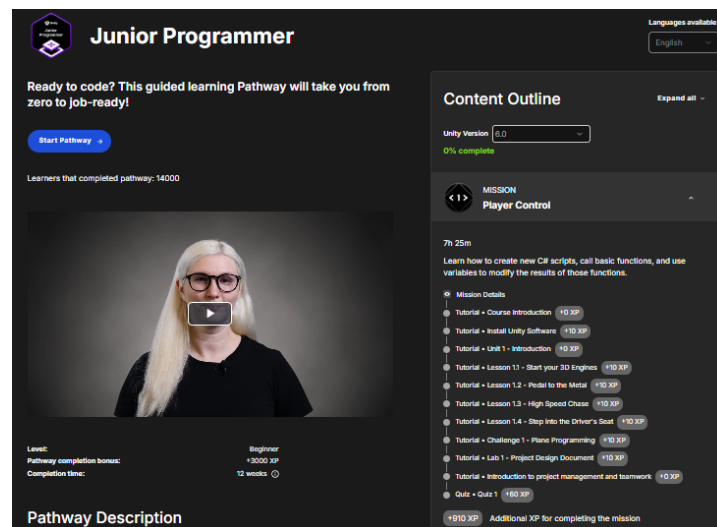
3. Tujuan Praktikum

- A. Memahami pembuatan sistem kontrol pemain menggunakan bahasa pemrograman C#.
- B. Mempelajari penggunaan komponen Rigidbody dan transformasi objek 3D di Unity.
- C. Mampu mengatur input pemain menggunakan keyboard.
- D. Mengembangkan pemahaman tentang manajemen proyek dan dokumentasi desain sederhana.
- E. Melatih keterampilan debugging dan pengujian hasil.

II. LANGKAH – LANGKAH PRAKTIKUM : MISI PLAYER CONTROL

1. Memulai Pathways Junior Programmer

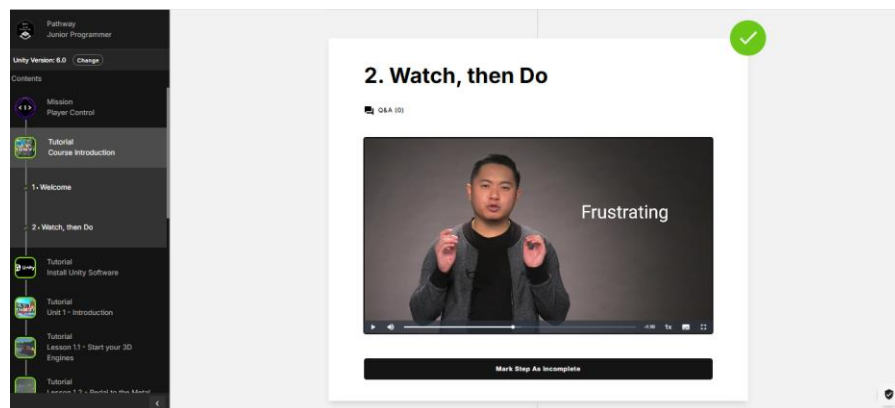
Pertama, Mahasiswa mengakses platform Unity Learn, melakukan pendaftaran atau login, lalu memilih pathway "Junior Programmer" dan menemukan misi "Player Control" sebagai fokus praktikum. Tahap ini bertujuan agar peserta memahami kerangka pembelajaran yang akan dijalani.



Tahap 1. Buka & Login Pathway Junior Programmer

2. Course Introduction

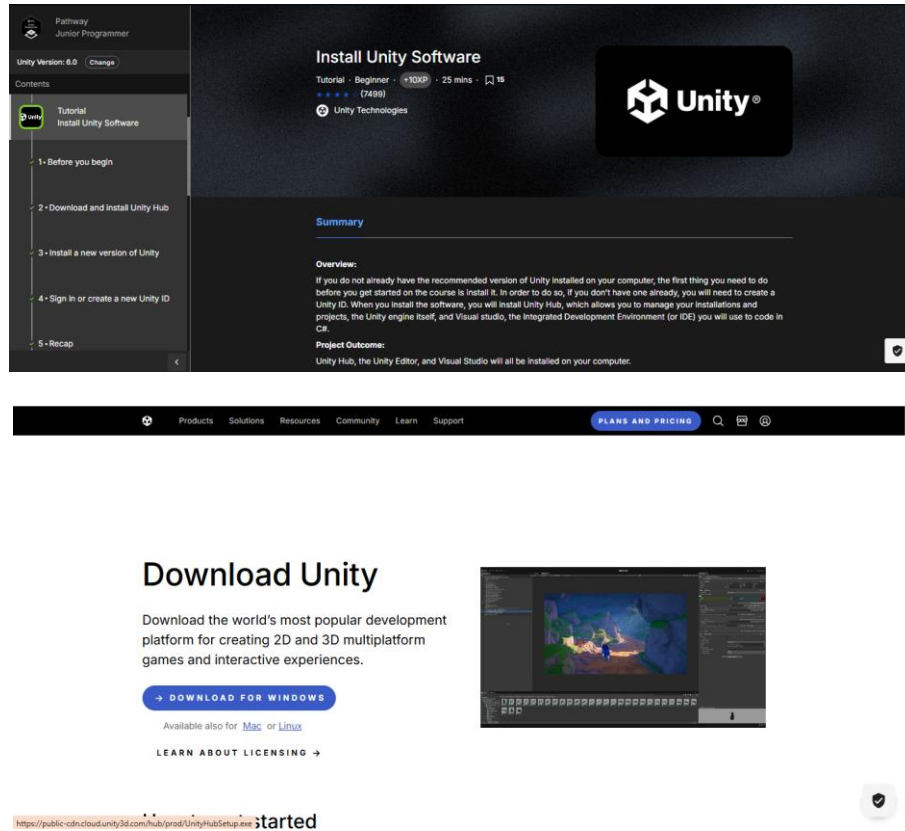
Pada bagian Course Introduction terdapat video yang berisi uraian tentang tujuan misi, durasi estimasi waktu, dan hasil akhir yang diharapkan. Pemahaman ini penting supaya peserta dapat mengarahkan fokus belajar pada penguasaan konsep dan praktik utama yang diberikan dalam misi.



Tahap 2. Pengenalan Tentang Tujuan Misi & Durasi Estimasi Waktu

3. Instal Unity Software

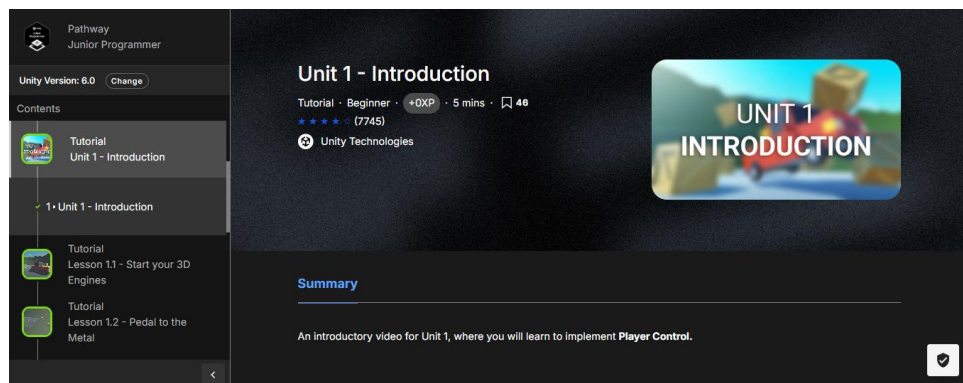
Sebelum praktikum, Mahasiswa diharapkan untuk mengunduh dan memasang Unity Hub serta Unity Editor versi yang direkomendasikan. Mahasiswa juga perlu memastikan semua modul serta paket yang diperlukan (misalnya fitur 3D) telah diinstal dan siap digunakan, supaya eksekusi tutorial berjalan tanpa kendala teknis.



Tahap 3. Mengunduh & Memasang Unity Hub serta Unity Editor

4. Unit Introduction

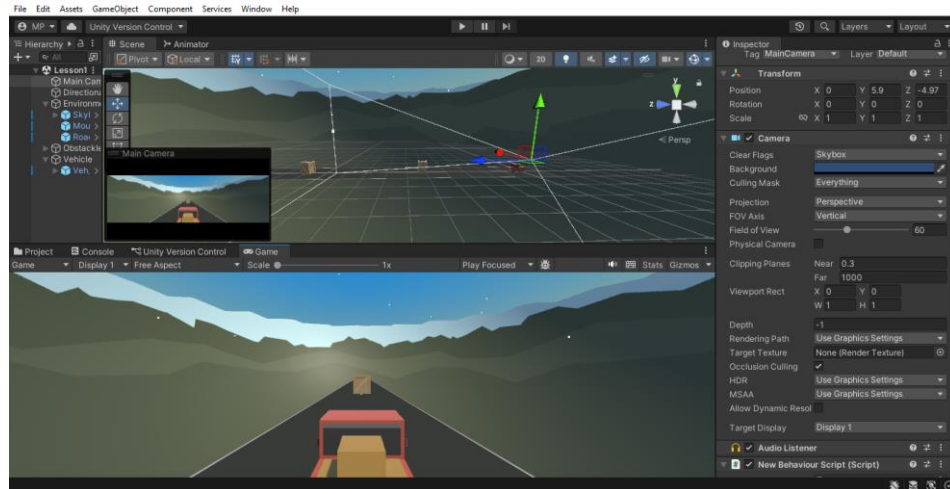
Sebelum memulai coding, Mahasiswa melihat dan menyimak penjelasan umum unit yang akan dipelajari, terutama mengenai skenario simulasi mobil. Ini membantu Mahasiswa memahami konteks dan alur kerja yang akan dilakukan di tahap-tahap berikutnya.



Tahap 4. Melihat & Menyimak Penjelasan Umum Unit yang Akan Di Pelajari

5. Start Your 3D Engines (Persiapan Scene)

Pada tahap ini dibuat project baru di Unity Editor dengan tipe 3D. Objek utama seperti mobil dan rintangan ditempatkan di scene dengan posisi dan orientasi tepat. Kamera utama disiapkan agar mengarah ke objek mobil supaya gameplay mudah diikuti. Tahapan ini membiasakan Mahasiswa menggunakan editor Unity untuk manajemen objek.



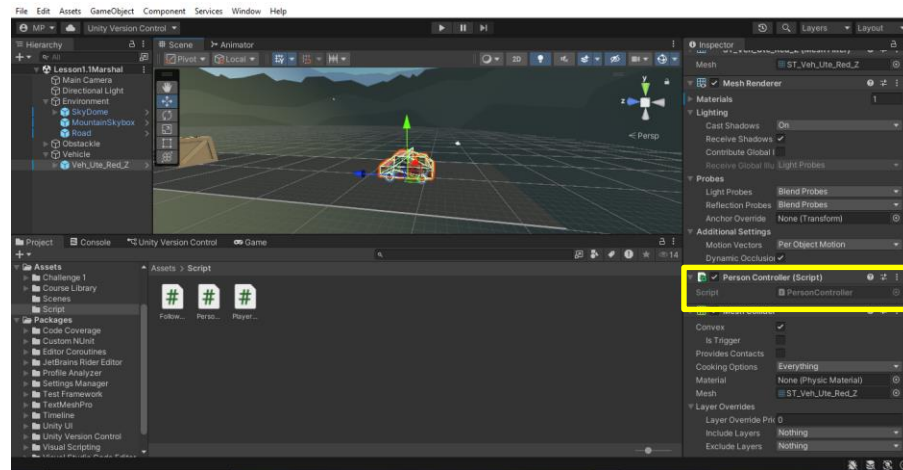
Tahap 5. Menempatkan Mobil, Rintangan, Kamera

6. Pedal to Metal (Membuat Script PlayerController)

Mahasiswa mulai membuat folder Scripts lalu membuat skrip C# bernama PlayerController. Dalam skrip ini, mahasiswa menulis kode untuk mengambil input keyboard (W/A/S/D atau panah) dan menerapkan gaya menggunakan Rigidbody pada mobil agar bergerak maju, mundur, dan menyamping. Ini melibatkan pemahaman fungsi Start, Update, dan FixedUpdate

```
C:\Users\DELL LATITUDE\Lesson1\Assets\Script\PersonController.cs - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
SpinPropellerX_Version2.cs PlayerControllerX.cs FollowPlayer.cs PersonController.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class PersonController : MonoBehaviour
6 {
7     //Private Variables
8     private float speed = 5.0f;
9     private float turnSpeed = 25.0f;
10    private float horizontalInput;
11    private float forwardInput;
12
13    // Start is called before the first frame update
14    void Start()
15    {
16    }
17
18    // Update is called once per frame
19    void Update()
20    {
21        //This is where we get player input
22        horizontalInput = Input.GetAxis("Horizontal");
23        forwardInput = Input.GetAxis("Vertical");
24
25        // We move the vehicle forward
26        transform.Translate(Vector3.forward * Time.deltaTime * speed * forwardInput);
27        // We turn the vehicle
28        transform.Rotate(Vector3.up, turnSpeed * horizontalInput * Time.deltaTime);
29    }
30 }
31
C# source file length: 826 lines: 32 Ln: 1 Col: 1 Pos: 1 Windows (CR LF) UTF-8 INS
```

Tahap 6. Pembuatan Script Pada PersonController



Tahap 6. Menerapkan Script ke Object Mobil

7. High Speed Chase

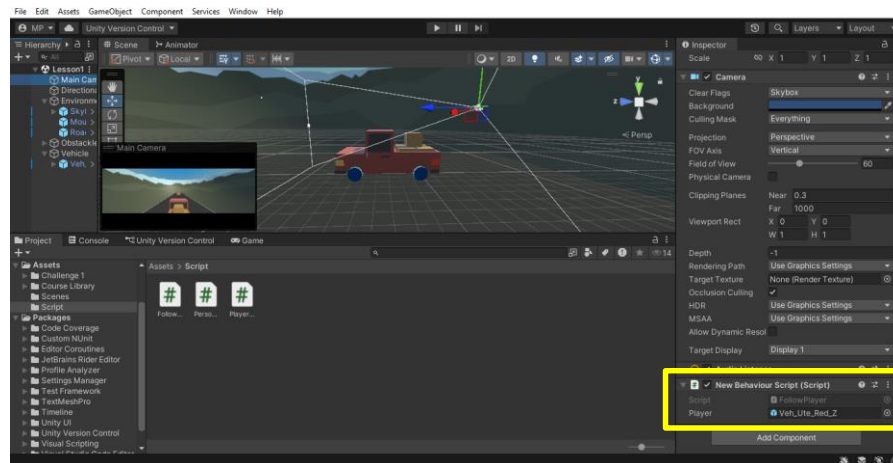
Selanjutnya peserta membuat skrip CameraController yang membuat kamera mengikuti posisi dan rotasi mobil secara dinamis saat game berjalan. Kamera diposisikan agar selalu berada di belakang mobil dengan jarak dan sudut yang pas, sehingga pemain mendapat pandangan yang nyaman.

```

C:\Users\DELL LATITUDE\Lesson1\Assets\Script\FollowPlayer.cs - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
FollowPlayer_Version2.cs SpinPropellerX_Version2.cs PlayerControllerX.cs FollowPlayer.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class NewBehaviourScript : MonoBehaviour
6 {
7     public GameObject player;
8     private Vector3 offset = new Vector3(0, 5, -7);
9
10    // Start is called before the first frame update
11    void Start()
12    {
13    }
14
15    // Update is called once per frame
16    void LateUpdate()
17    {
18        transform.position = player.transform.position + offset;
19    }
20
21 }
22
C# source file length: 455 lines: 22 Ln: 22 Col: 1 Pos: 456 Windows (CR LF) UTF-8 INS

```

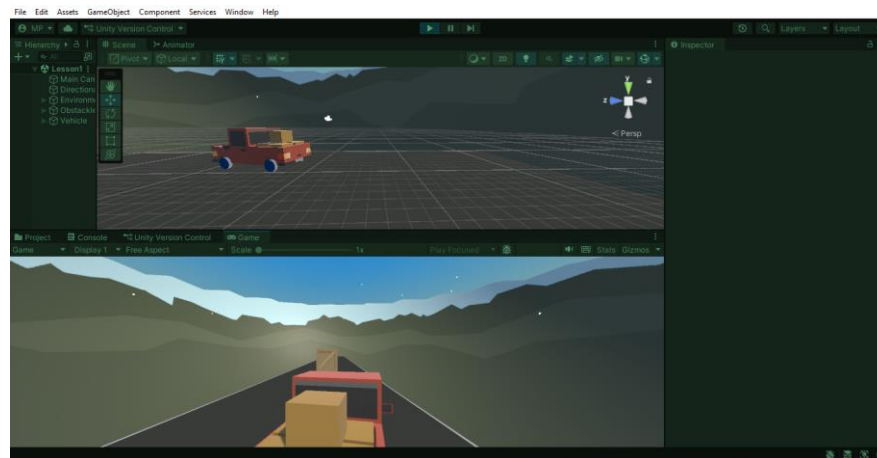
Tahap 7. Pembuatan Script CameraController



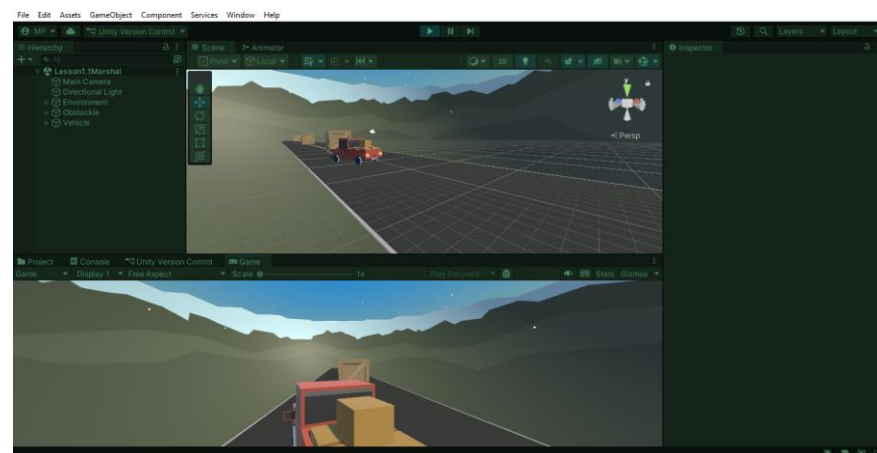
Tahap 7. Menerapkan Script ke Kamera

8. Step Into The Drive Seat

Di tahap ini, skrip PlayerController dikembangkan agar mobil dapat berbelok kiri dan kanan menggunakan input keyboard (A dan D). Mahasiswa belajar menghubungkan input dengan rotasi objek menggunakan transformasi rotasi yang responsif dan natural.



Tahap 8. Hasil Mengembangkan Agar Mobil Berbelok Ke Kiri & Kanan

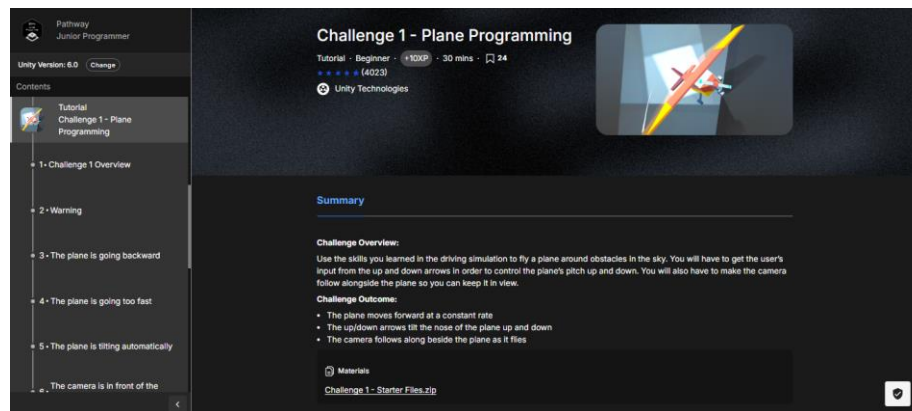


Tahap 8. Hasil Mengembangkan Agar Mobil Berbelok Ke Kiri & Kanan

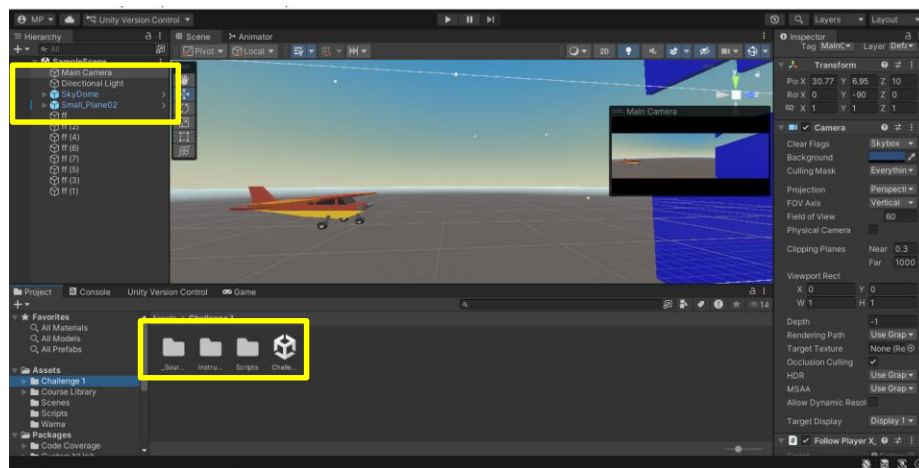
9. Challenge 1 Plane Programming

Mahasiswa mengaplikasikan konsep kontrol pemain ke objek baru yaitu pesawat, dengan karakteristik pergerakan yang berbeda (termasuk gerakan vertikal). Ini menstimulasi kreativitas dan penerapan konsep yang telah diajarkan secara mandiri. Berikut Langkah – langkah Plane Programming :

- A. Pada tahap awal challenge, Mahasiswa memulai dengan membuka proyek Junior Programmer di platform Unity Learn dan memilih starter scene dari Challenge 1 - Plane Programming. Scene berisi objek pesawat utama (Plane), kamera utama (Main Camera), dan komponen tambahan seperti baling-baling (Propeller). Untuk menjaga kerapian proyek, seluruh skrip dibuat di dalam folder baru bernama Scripts di direktori Assets.

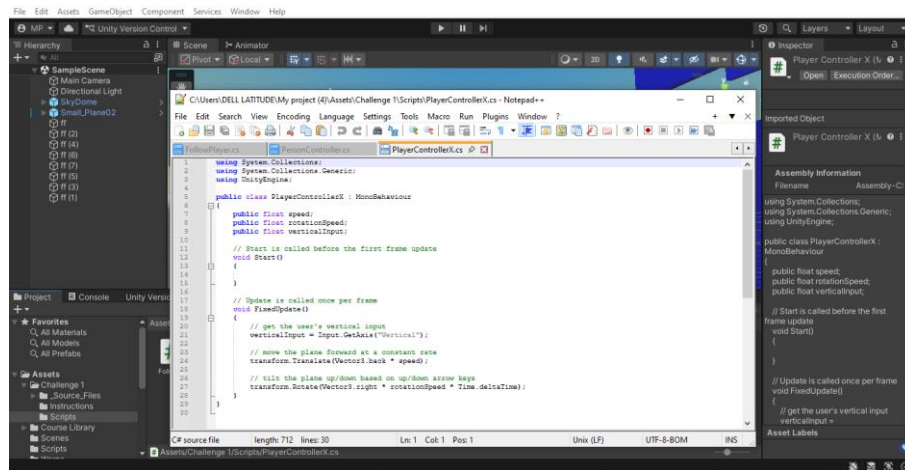


Tahap 1. Download Asset Plane Programming pada Website UnityLearn

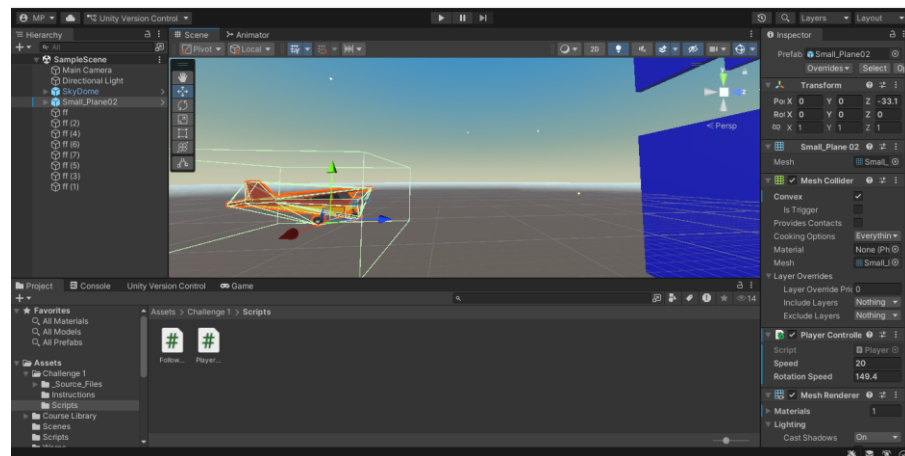


Tahap 2. Import Asset dan Merapikan Asset ke Dalam Folder

- B. Langkah pertama yang dilakukan adalah membuat skrip `PlayerControllerX.cs` dan memasangnya pada objek `Plane`. Dalam skrip ini, pesawat digerakkan maju secara konstan menggunakan perintah `transform.Translate(Vector3.forward*speed*Time.deltaTime)`. Selain itu, arah hidung pesawat diatur dengan input tombol panah atas dan bawah melalui `Input.GetAxis("Vertical")`, di mana nilai input dikalikan dengan `rotationSpeed` dan diterapkan pada rotasi pesawat menggunakan `transform.Rotate`. Penggunaan `Time.deltaTime` memastikan gerakan dan rotasi berjalan lancar dan konsisten meskipun frame rate berubah. Dengan penggabungan ini, pesawat dapat bergerak maju sambil menyesuaikan arah hidung secara responsif dalam satu skrip sederhana.

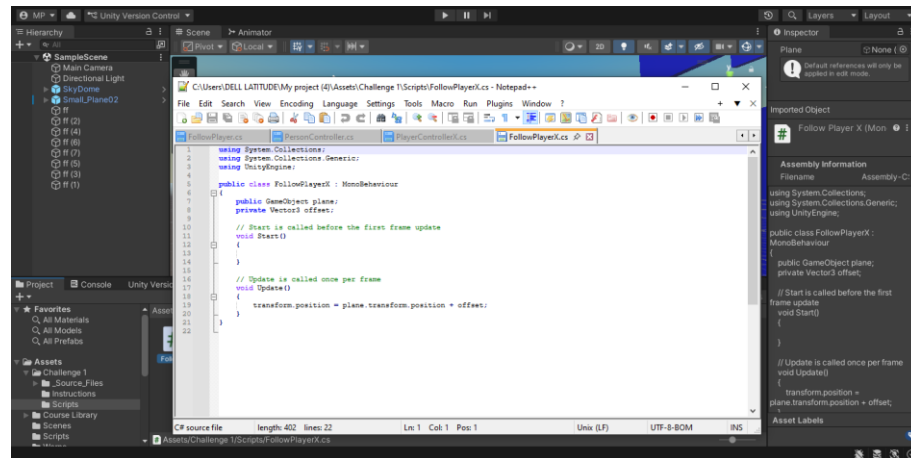


Tahap 3. Membuat Script `PlayerControllerX`

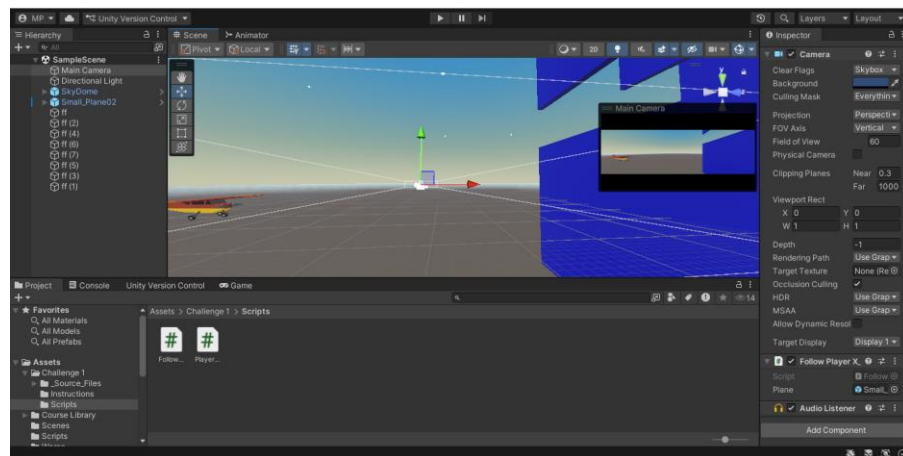


Tahap 4. Menerapkan Script `PlayerControllerX` Ke Object `Plane`

- C. Pada tahap berikutnya, Mahasiswa membuat kamera mengikuti pesawat dengan membuat skrip FollowPlayerX.cs yang dipasangkan pada objek Main Camera. Skrip mengatur posisi kamera agar senantiasa berada pada jarak tertentu dari pesawat dengan konsep offset position. Perintah `transform.position = plane.position + offset` memastikan posisi kamera selalu mengikuti pesawat, sementara logika di dalam fungsi `LateUpdate()` menjamin gerakan kamera lebih halus dan stabil karena dijalankan setelah semua perubahan posisi pesawat diperbarui.

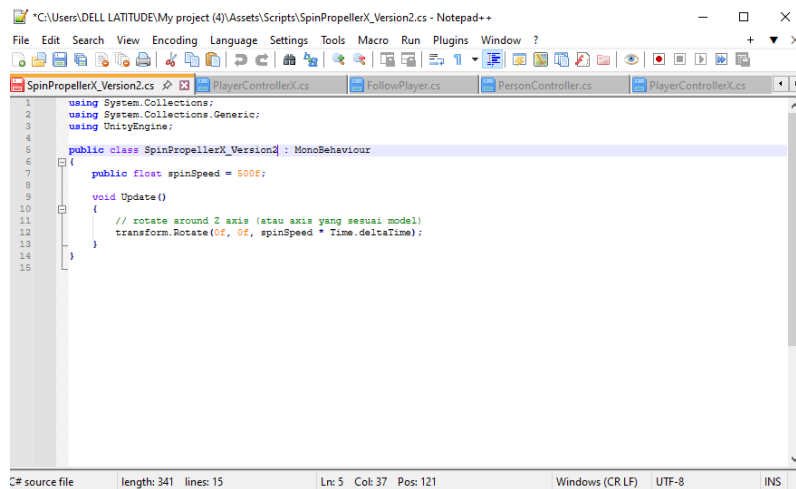


Tahap 5. Pembuatan Script FollowPlayerX



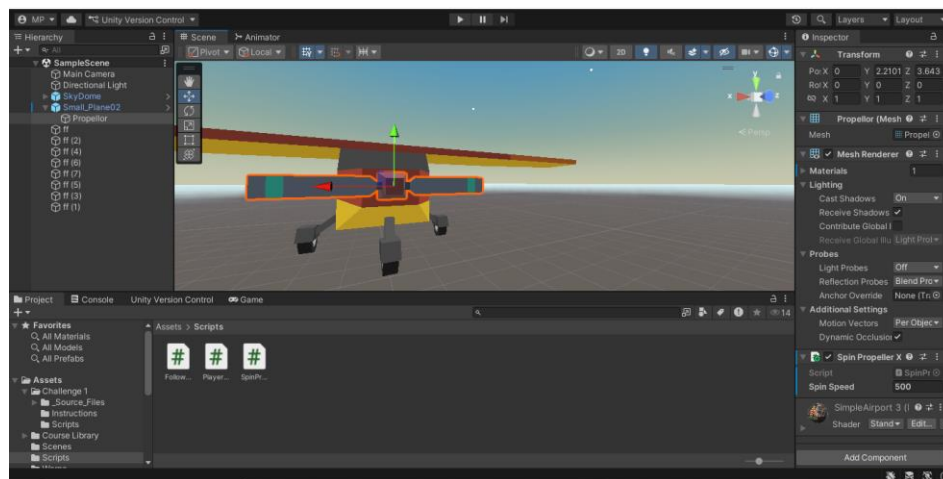
Tahap 6. Menerapkan Script FollowPlayerX ke Object MainCamera

- D. Untuk menambahkan efek rotasi baling-baling pesawat agar meningkatkan kesan visual realistis. Skrip `SpinPropellerX.cs` dipasang pada objek `Propeller` dan menggunakan perintah `transform.Rotate(0f, 0f, spinSpeed * Time.deltaTime)` untuk membuat baling-baling berputar terus-menerus sepanjang sumbu porosnya. Kecepatan putar bisa dikontrol melalui variabel publik `spinSpeed` di `Inspector`, sehingga dapat disesuaikan dengan kebutuhan.



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class SpinPropellerX_Version2 : MonoBehaviour
6 {
7     public float spinSpeed = 500f;
8
9     void Update()
10    {
11        // rotate around Z axis (atau axis yang sesuai model)
12        transform.Rotate(0f, 0f, spinSpeed * Time.deltaTime);
13    }
14 }
15
```

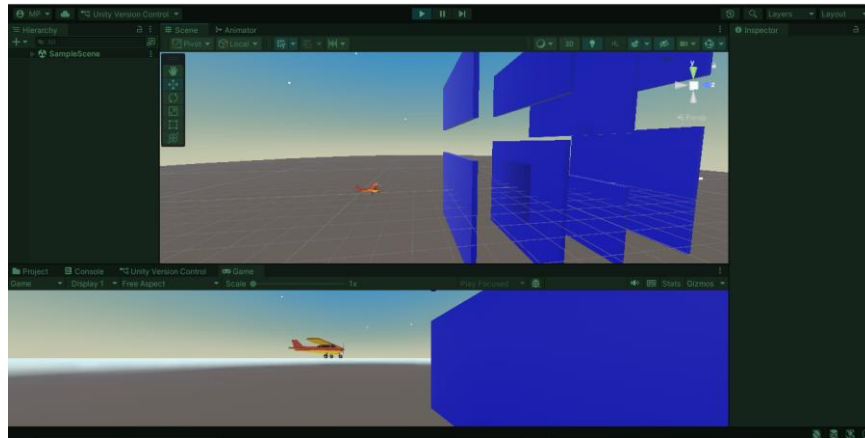
Tahap 7. Pembuatann Script untuk Proppeler(Baling - Baling)



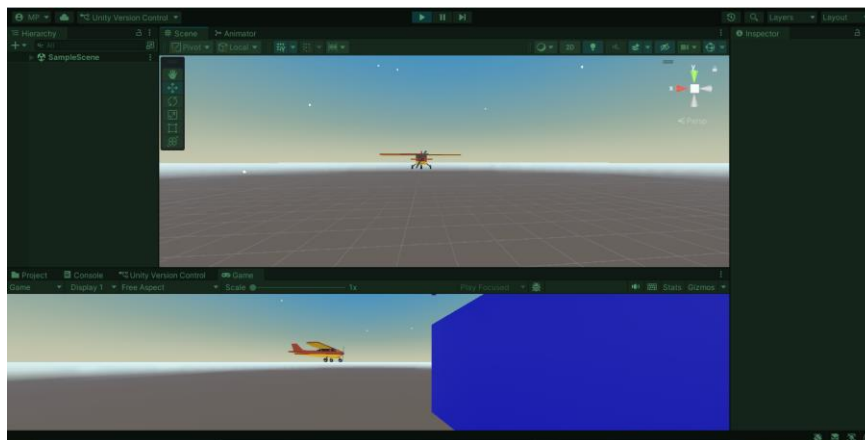
Tahap 8. Menerapkan Script Proppeler ke Proppeler(Baling - Baling)

- E. Pemenambahan skrip pada objek pesawat agar ketika terjadi tabrakan dengan obstacle, permainan otomatis melakukan reset. Implementasi ini dilakukan dengan menambahkan fungsi `OnCollisionEnter` yang mendeteksi tabrakan dan langsung memuat ulang scene aktif menggunakan `SceneManager.LoadScene`. Dengan demikian, saat pesawat menabrak hambatan, seluruh permainan akan kembali ke kondisi awal, memungkinkan simulasi penerbangan dimulai ulang secara otomatis.

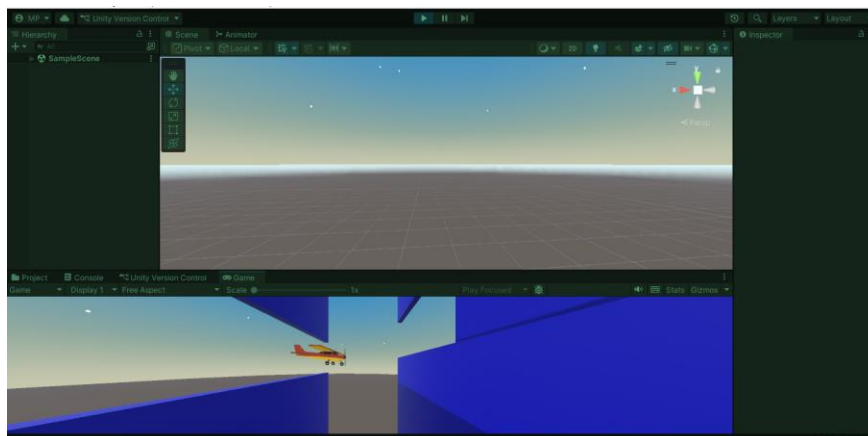
- F. Setelah semua skrip dipasang, Mahasiswa melakukan pengujian dengan menekan tombol Play di Unity Editor. Hasilnya, pesawat bergerak maju secara konstan, kamera mengikuti dari samping dengan posisi offset yang stabil, dan kontrol pitch pesawat merespons input pengguna dengan baik. Pada saat yang sama, baling-baling pesawat berputar terus memberikan efek visual dinamis yang memperkaya simulasi.



Hasil Plane Programming



Hasil Plane Programming



Hasil Plane Programming

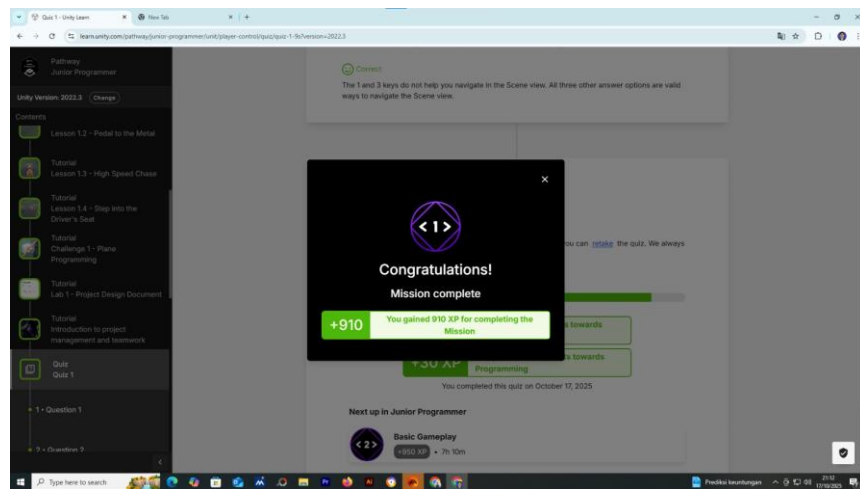
10. Project Design Document

Mahasiswa mempelajari tentang menyusun dokumen desain proyek yang mencakup ide utama game, fitur gameplay, dan rencana implementasinya. Latihan ini menguatkan kemampuan perencanaan dan komunikasi ide program secara profesional.

11. Introduction to Project Management and Teamwork

Mahasiswa mempelajari pentingnya manajemen proyek serta kolaborasi tim dalam proses pengembangan aplikasi dan game. Tutorial ini memperkenalkan konsep dasar dan praktik terbaik yang banyak digunakan dalam dunia kerja nyata, meliputi tahapan perencanaan proyek, fase-fase produksi, manajemen pelaksanaan, hingga strategi kerja kelompok.

12. Quiz



Hasil Quiz

III. Analisis dan Kesimpulan

1. Analisis :

Praktikum Pathway UnityLearn – Misi Player Control setiap tahap dalam misi Player Control mulai dari pengenalan Unity dan instalasi software, pembuatan scene mobil, hingga pengembangan skrip kontrol pesawat. Melalui tutorial-tutorial yang tersedia, Mahasiswa mempelajari pembuatan skrip C#, penerapan fungsi logika dasar seperti pergerakan objek menggunakan transform, serta pemrosesan input dari pengguna untuk mengarahkan pesawat secara dinamis, Mahasiswa diajak memahami hubungan antara konsep pemrograman dan implementasi praktis di engine Unity dan Penambahan mekanisme restart otomatis saat tabrakan di Plane Programming juga memberikan pengalaman simulasi game yang lebih realistis dan interaktif. Melalui pengujian berulang dan penyesuaian variabel seperti speed dan rotationSpeed, peserta belajar pentingnya debugging dan optimasi untuk menghasilkan permainan yang responsif dan mulus.

2. Kesimpulan :

Praktikum pada pathway Player Control berhasil membangun pondasi yang kuat dalam pemahaman pengembangan game menggunakan Unity dan bahasa pemrograman C#. Setiap tahap tutorial memberikan instruksi jelas dan sistematis, sehingga penulis mampu mengaplikasikan konsep pemrograman dan logika kontrol ke dalam simulasi pesawat yang interaktif. Implementasi challenge berhasil memperkuat skill pemecahan masalah, adaptasi kode, dan pengujian fitur.