

KOMPUTASI MULTIMEDIA

MODUL 11

MEMAINKAN DAN MEMANIPULASI SOUND

Oleh:

KINANTI PERMATA PUTRI

NIM. 1841720022

TI-3D



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

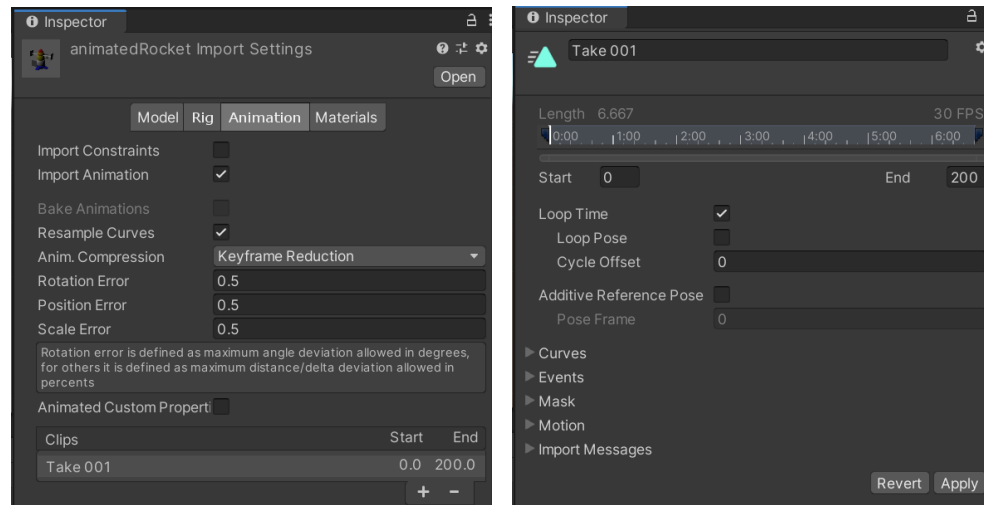
POLITEKNIK NEGERI MALANG

MEI 2021

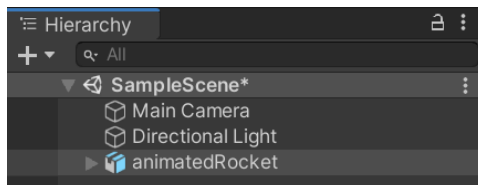
A. Praktikum

1. Matching audio pitch pada kecepatan animasi

- 1) Buat project baru 3D.
- 2) Import file animatedRocket.fbx dari folder 1362_09_01.
- 3) Pada Project view pilih animatedRocket.fbx, kemudian pada Inspector view amati import Settings. Pilih Animation tab kemudian pilih pada clip Take 001, dan pastikan Loop Time di centang/check. Langkah terakhir klik button Apply untuk menyimpan perubahan.



- 4) Drag animatedRocket.fbx pada Hierachy view.

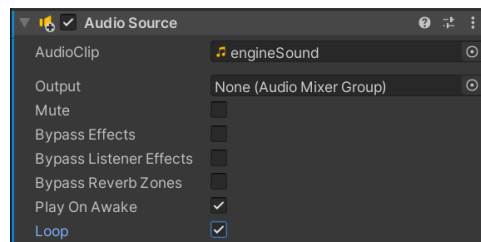


- 5) Import engineSound.wav dari folder 1362_09_01.

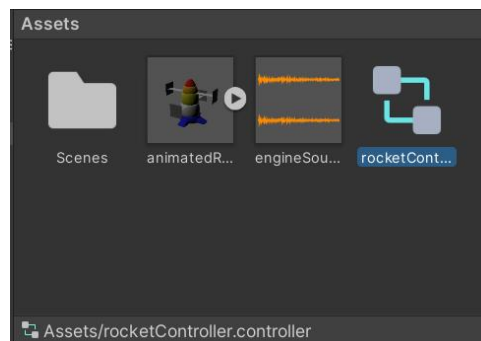
- 6) Tambahkan Audio Source pada Game Object animatedRocket. Dengan cara drag engineSound pada Object animatedRocket. Setelah di drag hasilnya sebagai berikut:



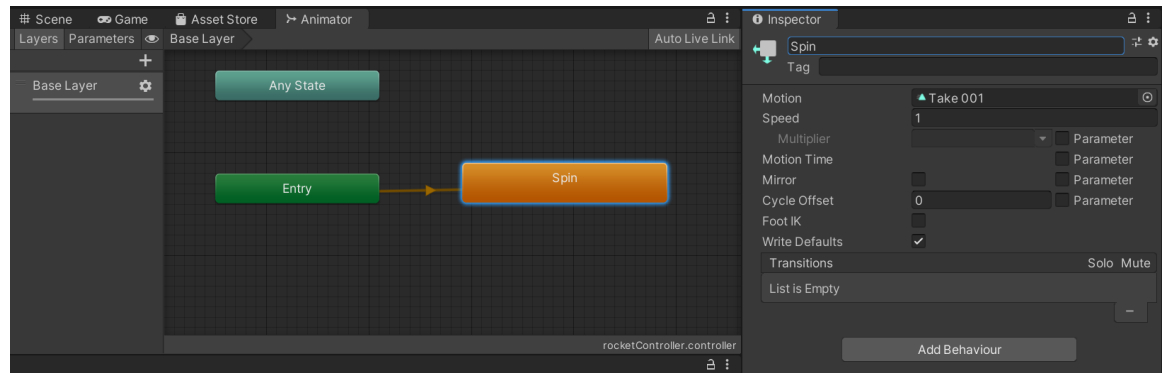
- 7) Pada komponen Audio Source dari animatedRocket, centang loop. Seperti pada gambar berikut:



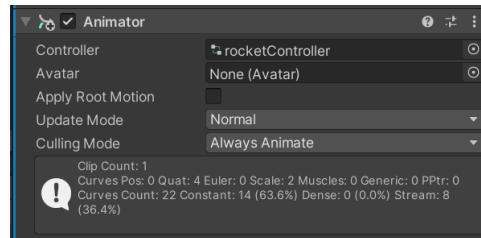
- 8) Selanjutnya buat Controller untuk animatedRocket. Pada Project view, klik Create dan pilih Animator Controller, kemudian beri nama sebagai rocketController.



- 9) Doppel klik objek rocketController untuk membuka Animator window. Kemudian klik kanan pada grid area dan pilih Create State | empty.
- 10) Beri nama state baru dengan nama Spin dan ubah nilai Motion Field dengan Take 001.



- 11) Dari Hierarchy view, pilih animatedRocket. Kemudian pada komponen Animator ubah nilai Controller menjadi rocketController dan pastikan Apply Root Motion tidak dicentang.

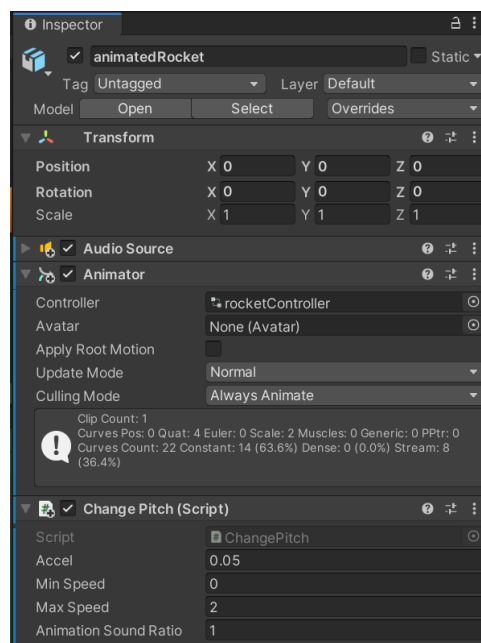


- 12) Pada project view buat C# Script dengan nama ChangePitch.

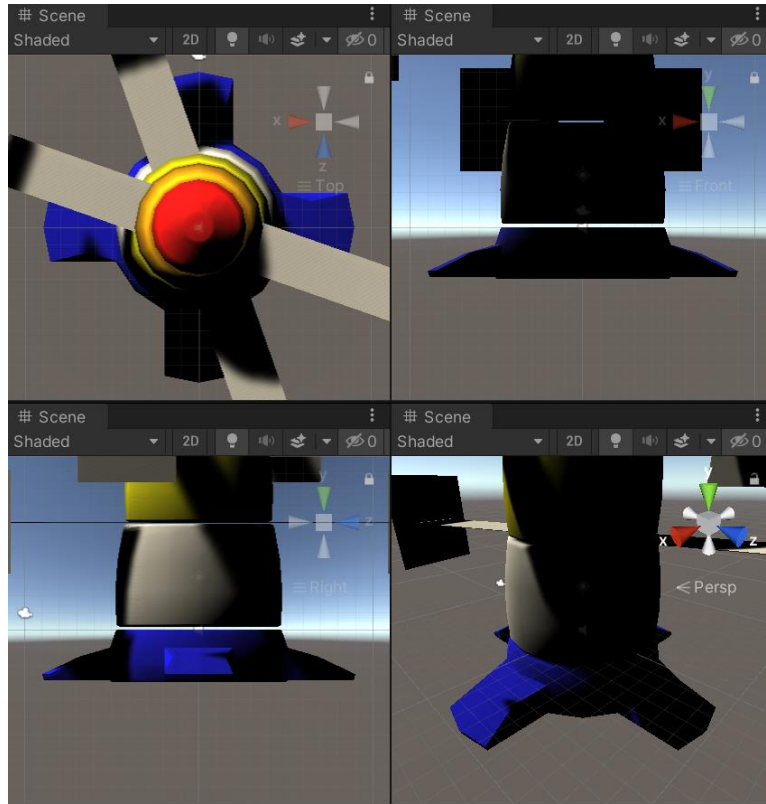
13) Klik dubel ChangePitch, ubah source code yang ada dengan source code berikut:

```
1 using UnityEngine;
2
3 0 references
4 public class ChangePitch : MonoBehaviour{
5     2 references
6     public float accel = 0.05f;
7     1 reference
8     public float minSpeed = 0.0f;
9     2 references
10    public float maxSpeed = 2.0f;
11    1 reference
12    public float animationSoundRatio = 1.0f;
13    6 references
14    private float speed = 0.0f;
15    4 references
16    private Animator animator;
17    2 references
18    private AudioSource audioSource;
19
20 0 references
21 void Start(){
22     animator = GetComponent<Animator>();
23     audioSource = GetComponent<AudioSource>();
24     speed = animator.speed;
25     AccelRocket (0f);
26 }
27
28 0 references
29 void Update(){
30     if (Input.GetKey (KeyCode.Alpha1))
31         AccelRocket(accel);
32     if (Input.GetKey (KeyCode.Alpha2))
33         AccelRocket(-accel);
34 }
35
36 3 references
37 public void AccelRocket(float accel){
38     speed += accel;
39     speed = Mathf.Clamp(speed,minSpeed,maxSpeed);
40     animator.speed = speed = Mathf.Clamp (speed, 0, maxSpeed);
41     float soundPitch = animator.speed * animationSoundRatio;
42     audioSource.pitch = soundPitch;
43 }
44 }
```

14) Simpan perubahan script dan tambahkan ChangePitch sebagai komponen GameObject animatedRocket.



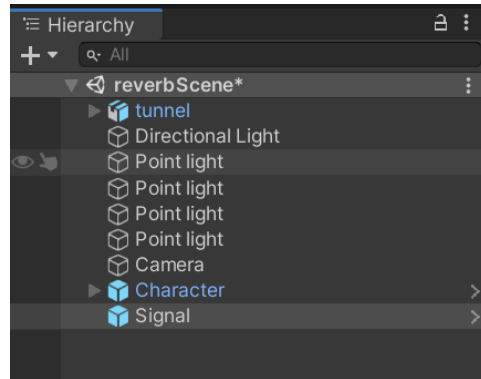
15) Play project yang telah dibuat hasilnya adalah seperti pada gambar dibawah ini. Kecepatan animasi dapat dingan menekan button 1 (percepatan) dan button 2 (memperlambat) pada keyboard. Audio akan mengikuti kecepatan object.



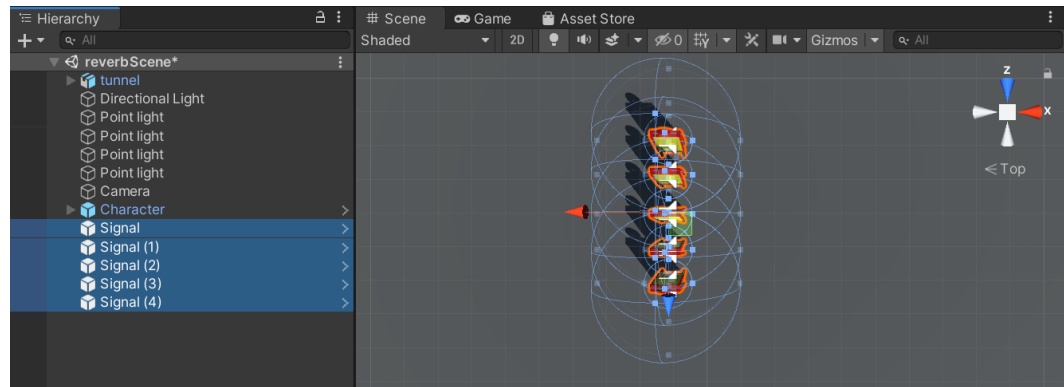
Penjelasan: Pada praktikum yang pertama kita bisa mempelajari cara menyesuaikan animasi dan sound di sebuah objek rocket dengan menggunakan animator dan objek audio source di rocket.

2. Simulating Acoustic environment with reverb zone

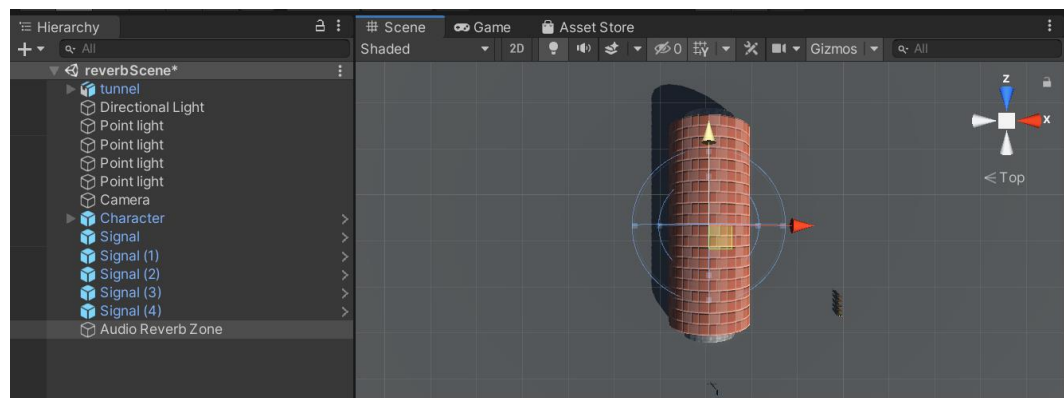
- 1) Import file ReverbZone dari folder 1362_09_02 ke Unity Project.
- 2) Pada Project view, klik doble pada reverbScene yang terdapat pada folder ReverbZones. Ini adalah basic scene pada praktikum ke-2 ini, yang terdiri dari terowongan / tunnel, karakter dan kontroler untuk mengatur gerakan karakter.
- 3) Setelah reverbScene dibuka, drag Signal prefab pada Hierarchy view. Letakkan Signal di tengah terowongan. Hal ini digunakan untuk menambahkan pemancar suara.



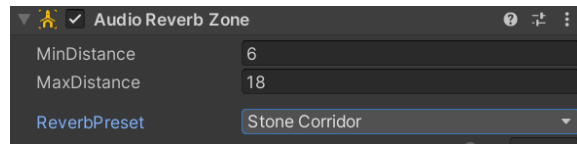
- 4) Copy Signal sebanyak 5 kali dan atur posisinya seperti pada gambar dibawah ini.



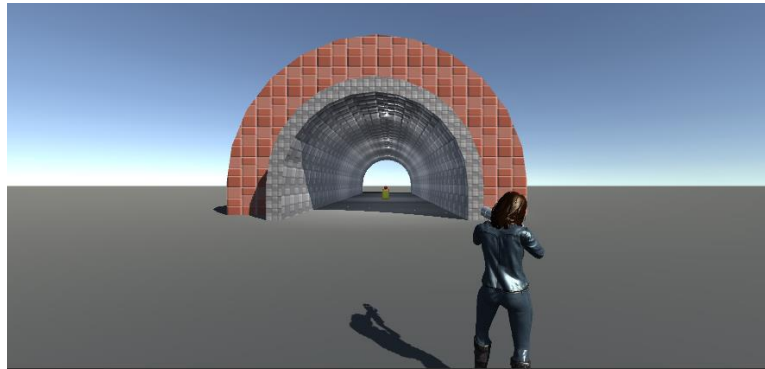
- 5) Pada Hierarchy view buat Audio Reverb Zone dengan cara Create | Audio | Audio Reverb Zone, kemudian letakkan di tengah terowongan.



- 6) Pilih game Object Reverb Zone. Pada Inspector view, ubah parameter komponen Reverb Zone, seperti berikut Min Distance:6; Max Distance:18 dan Preset:StoneCorridor. seperti pada gambar berikut:



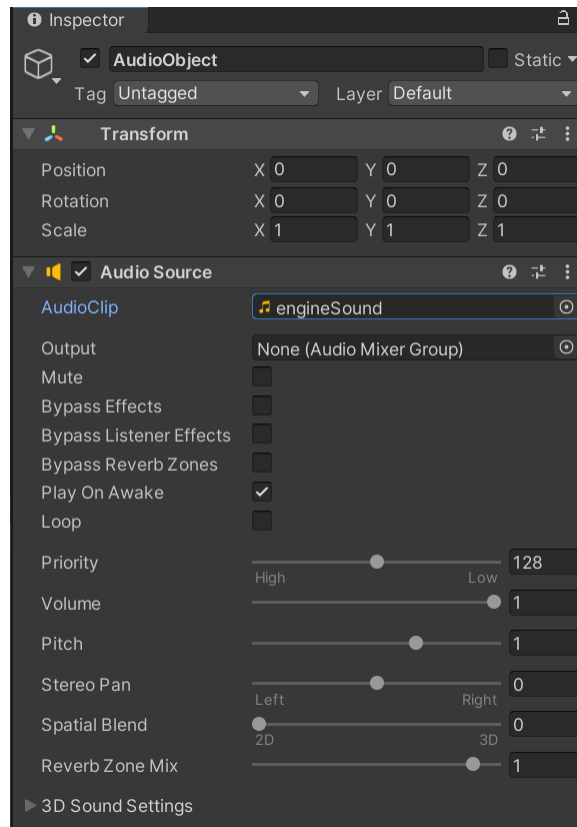
- 7) Play project yang telah dibuat. Anda dapat menggerakkan karakter dengan menekan button WASD dan Shift untuk berlari. Hasil akhirnya adalah anda dapat mendengar suara bergema pada Reverb Zone, sebaliknya suara akan semakin mengecil jika menjauhi Reverb Zone.



Penjelasan: Pada praktikum kedua kita telah mempelajari cara mengatur suara yang agar bisa menggema dengan menggunakan reverb zone. Reverb zone ini bisa diatur berdasarkan luas nya zona agar bisa mengeluarkan suara di sekitar zona itu.

3. Preventing an Audio Clip from restarting if it already playing
- 1) Buat project baru 2D.
 - 2) Buat Empty GameObject dan beri nama sebagai AudioObject. Selanjutnya tambahkan komponen Audio Source dengan cara Select AudioObject | Add Component | Audio | Audio Source.

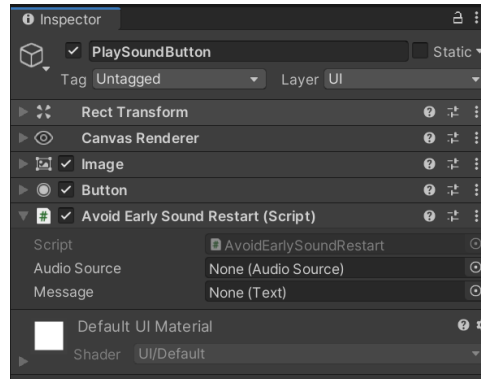
- 3) Import engineSound.wav Audio clip dari folder 1362_09_03. Kemudian drag engineSound dari Project view ke parameter Audio Clip pada AudioSource.



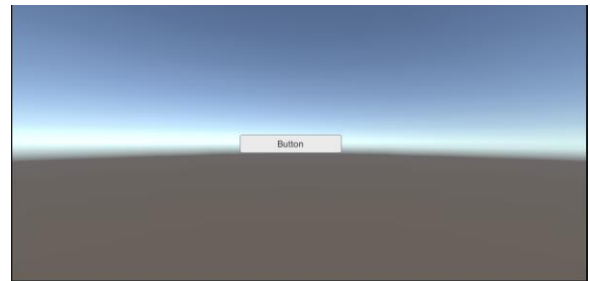
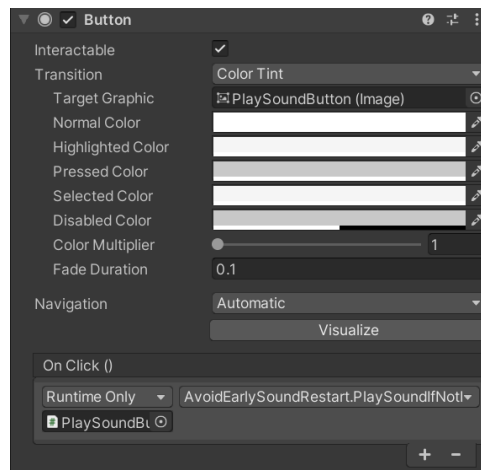
- 4) Pada langkah keempat, buat UI Button dengan nama PlaySoundButton pada hierarchy view dan tambahkan C# script pada AvoidEarlySoundRestart. Berikut adalah source codednya:

```
1 using UnityEngine;
2 using System.Collections;
3 using UnityEngine.UI;
4
5 0 references
6 public class AvoidEarlySoundRestart : MonoBehaviour {
7     3 references
8     public AudioSource audioSource;
9     1 reference
10    public Text message;
11
12    0 references
13    void Update(){
14        string statusMessage = "Play sound";
15        if(audioSource.isPlaying)
16            statusMessage = "(sound playing)";
17        message.text = statusMessage;
18    }
19
20    // button click handler
21    0 references
22    public void PlaySoundIfNotPlaying(){
23        if( !audioSource.isPlaying)
24            audioSource.Play();
25    }
26 }
```

- 5) Pada langkah kelima pilih PlaySoundButton pada hierarchy view selanjutnya tambahkan event baru. Untuk menambahkan event pada PlaySoundButton dengan cara tekan “+” pada event selanjutnya PlaySoundButton untuk mengisi object, selengkapnya seperti pada gambar berikut:



- 6) Pada langkah terakhir Play project yang telah dibuat. Hasil akhirnya adalah sound dimainkan hanya sekali, setelah itu akan akan berhenti dan text pada button akan berubah menjadi Play Sound. Jika pada saat text button adalah Play Sound, button dapat di klik untuk memainkan sound, selanjutnya teks pada button berubah menjadi Sound Playing.



Penjelasan: Setelah praktikum 3 kita bisa mempelajari cara membuat button untuk mencegah sound berbunyi.

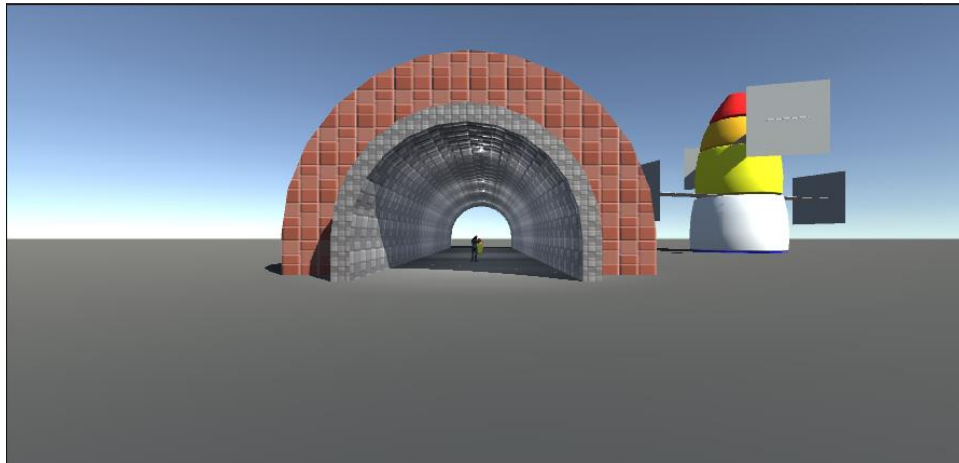
B. Tugas Praktikum

- 1) Kerjakan seluruh project yang ada dimodul ini dan kumpulkan. Berikan penjelasan yang kalian dapatkan setelah melakukan praktikum.

Penjelasan sudah ada di setiap praktikum.

- 2) Buat project menggunakan package pada folder Tugas, tambahkan Praktikum ke 1 dan Praktikum 2 pada package yang telah disediakan. Ulangi pembuatan project supaya lebih paham.

Hasil:



Selanjutnya hasil penjelasan akan dijelaskan di video Youtube.

Link Youtube: https://youtu.be/8jgXSZnJg_A

Link Github: https://github.com/kinanpermata/TI-3D_17_Komputasi-Multimedia