

MODUL I

KOMPUTER GRAFIK 2D
PENGENALAN GODOT

D4 TEKNIK INFORMATIKA
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA
POLITEKNIK NEGERI BANDUNG



MAHASISWA 000 | KOMPUTER GRAFIK | FEBRUARI, 11 2025

CONTENTS

SETUP GODOT.....	0
GODOT PROJECT MANAGER	0
GODOT UI.....	5
SCENE SYSTEM	5
TASK PRAKTIKUM.....	7

SETUP GODOT

Godot merupakan game engine open-source portable yang multi-platform (Windows, Linux & Mac OS). Game engine pada godot dapat membantu developer untuk membangun video game dengan mudah karena menggunakan pendekatan high level programming language (C#, GDScript dan Python). Bersifat portable, untuk menggunakan Godot, developer tidak perlu dilakukan instalasi; berikut langkah-langkah setup Godot pada sistem operasi Windows.

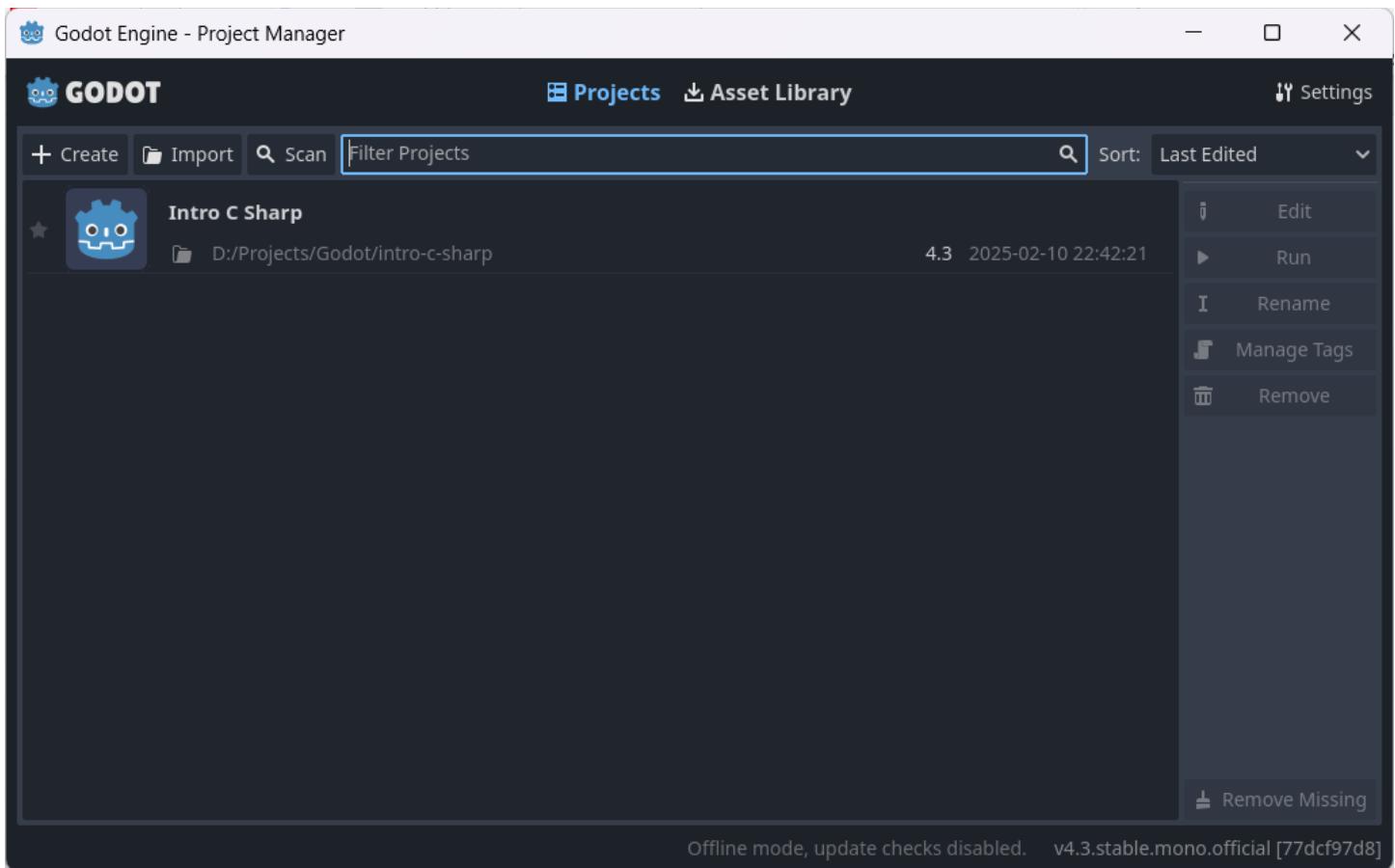
- I. Buka halaman <https://godotengine.org/download/windows> , atau pilih versi Godot sesuai dengan sistem operasi anda.



2. Download Versi .Net, pada praktikum Komputer grafik 2D Bahasa yang digunakan adalah C#
3. Jika komputer/laptop anda mengalami kendala saat menjalankan Godot versi 4.3 silakan untuk mendownload versi Godot yang digunakan; rekomendasi untuk praktikum gunakan Godot 3.6 Anda dapat mendownload versi sebelumnya pada tautan berikut: <https://godotengine.org/download/3.x/windows/>
4. Install dotnet versi 8 <https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/sdk-8.0.405-windows-x64-binaries> install step by step dan pastikan setelah diinstall eksekusi perintah dotnet.exe di command line dan berjalan.
5. Silakan extract file Godot_v4.3, anda dapat menggunakan 7zip, atau tools lainnya.
6. Jalankan file executable Godot_4.3, anda sudah dapat mengoperasikan Game Engine Godot.

GODOT PROJECT MANAGER

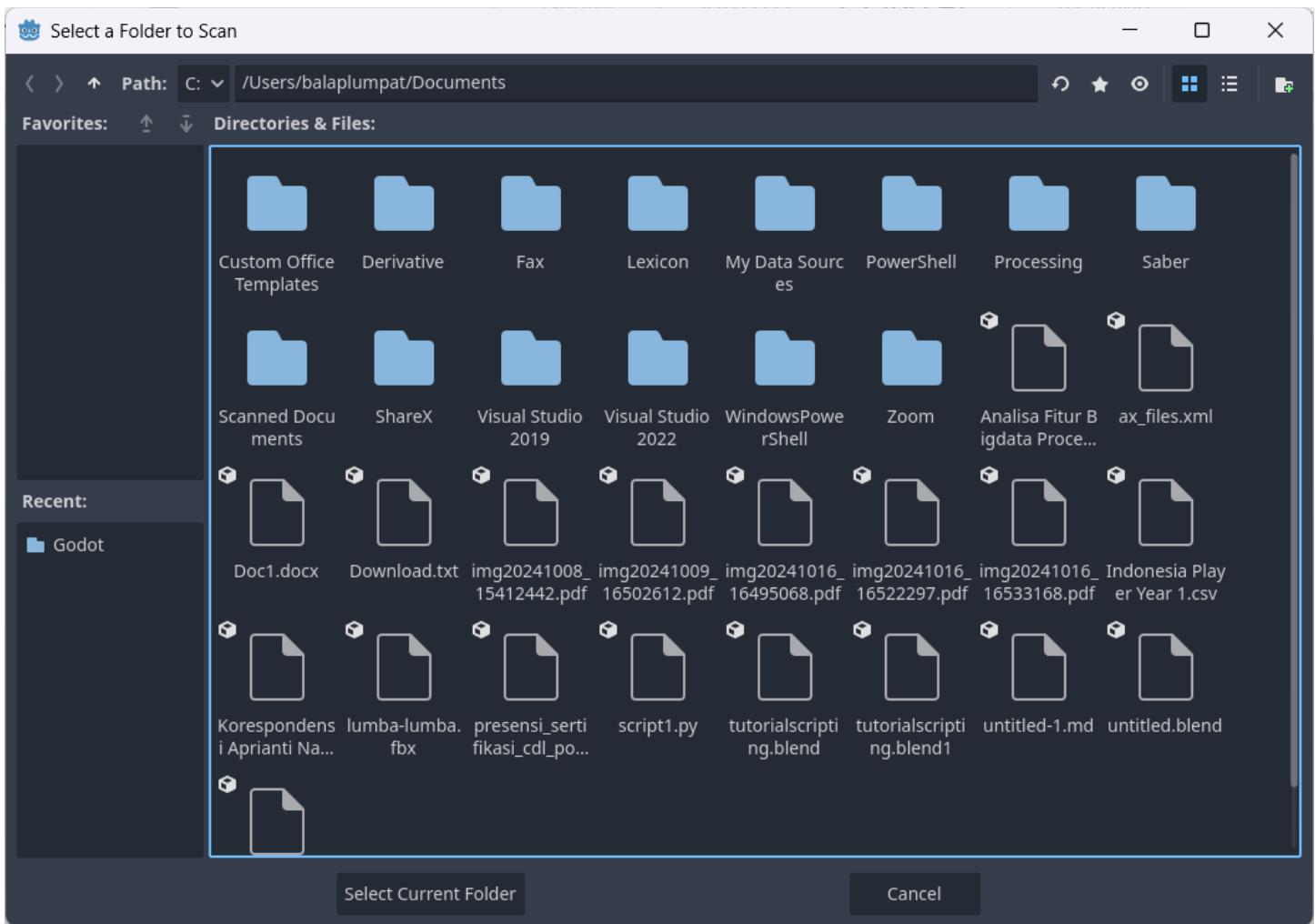
Project Manager adalah antar muka pertama setelah mengeksekusi Godot, berfungsi untuk melakukan manajemen proyek-proyek karya 2D/3D seperti membuat proyek baru, load atau import proyek yang sudah ada sebelumnya, menghapus dan melaukan running karya 2D/3D. Selain memulai dari nol, terdapat pula pilihan template karya 2D/3D yang sudah ada untuk dimodifikasi oleh developer.



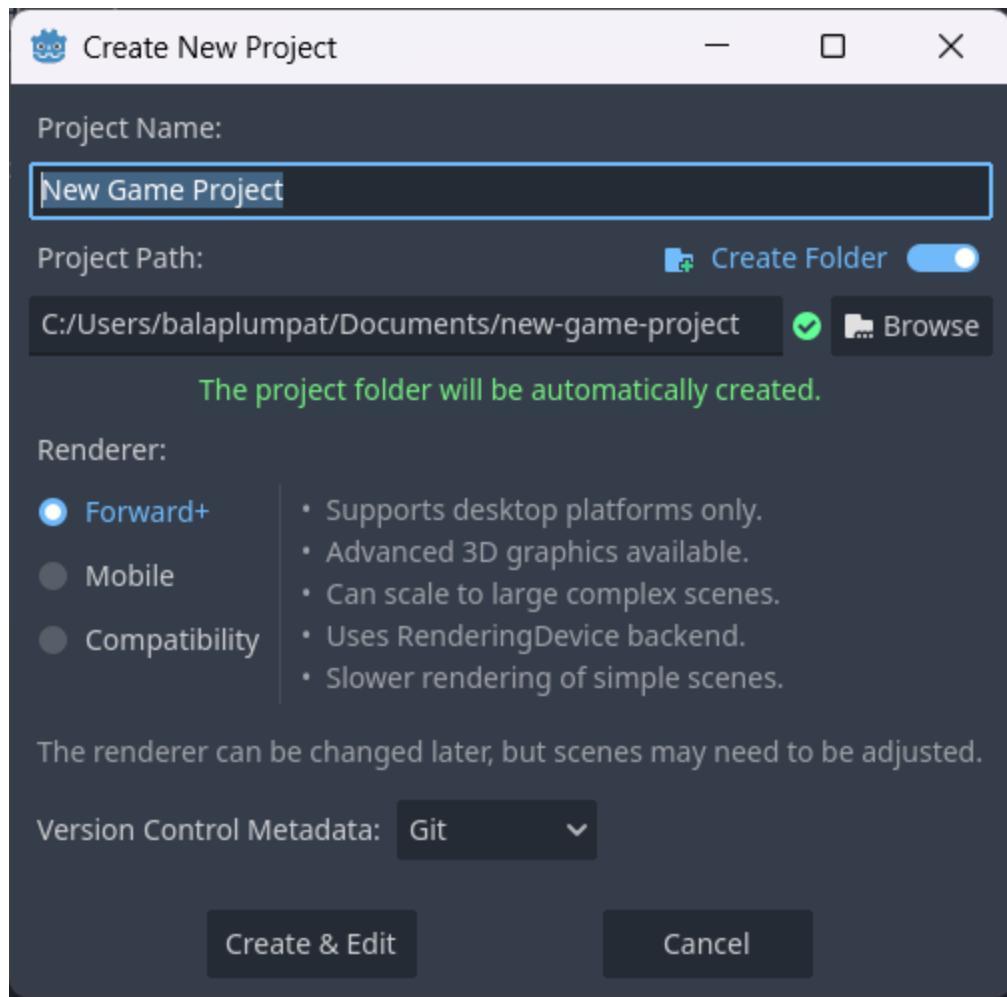
1. List dari Local Project yang telah di scan oleh Godot Engine pada sesi sebelumnya, saat ini masih kosong karena inisiasi pertama.
2. Asset Library Project merupakan template karya 2D/3D atau game yang disediakan oleh kontributor Godot.
3. Menu untuk melakukan manajemen proyek.

Secara mekanisme, untuk melakukan manajemen proyek 2D/3D dimulai dari pencarian folder pada direktori yang dimiliki User/Pengguna Godot, setelah itu dilakukan sesuai dengan task nya, seperti scan, import dll. Kecuali pada menu new project

- Scan



- New Project



Langkah-langkah membuat new project pada Godot, diperlukan tiga inputan yaitu nama project, path, dan jenis renderer.

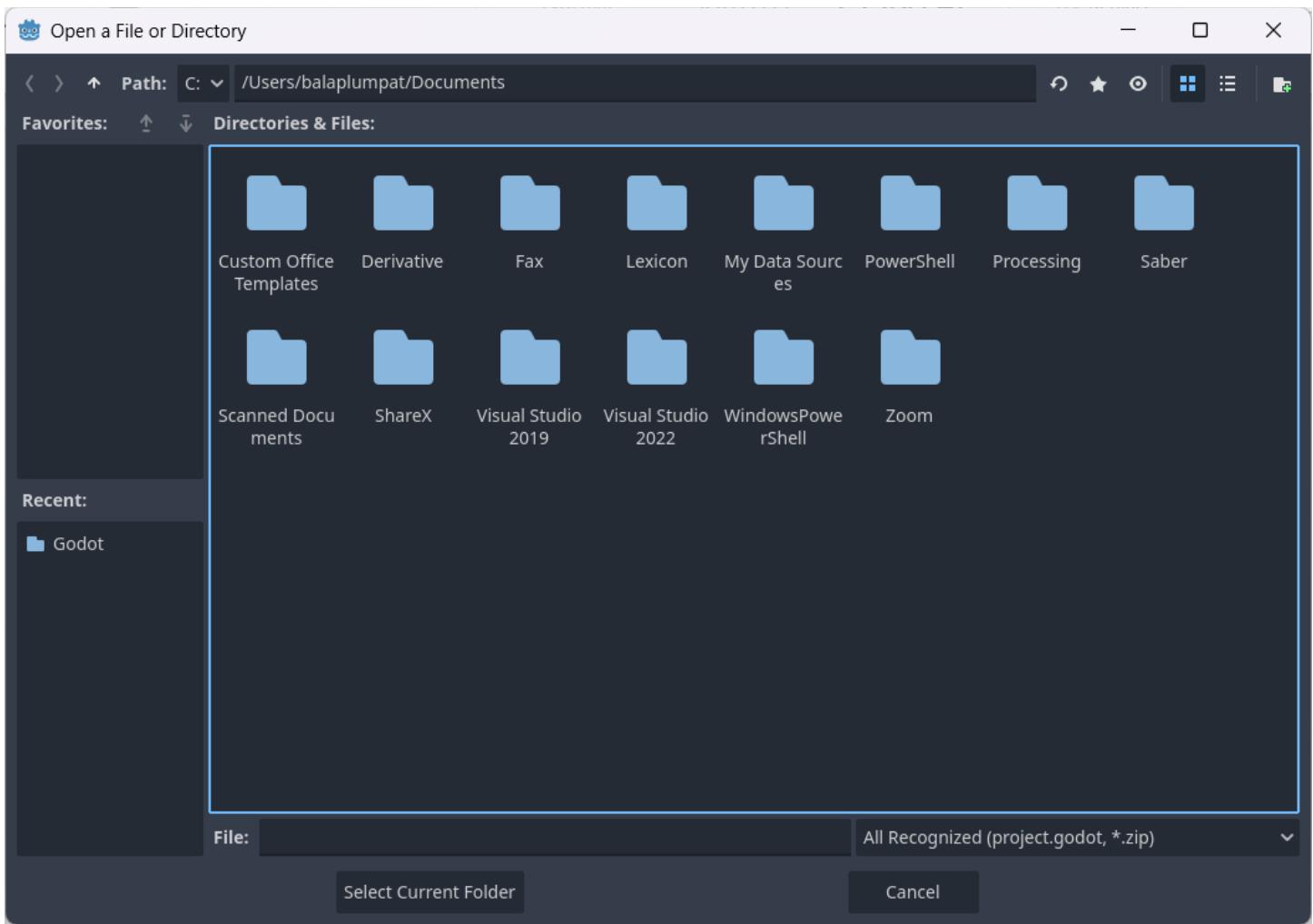
1. Pada praktikum Komputer Grafik 2D, penamaan nama project memiliki standar sbb:

[KG2025_2A_D4_2024]_Modull_001

Kode	Penjelasan
KG2023	Komputer Grafik tahun 2025
2A/B	Kelas 2
D4	Prodi D4
2024	Angkatan
Modull	No Modul
001	3 Digit Nim Terakhir

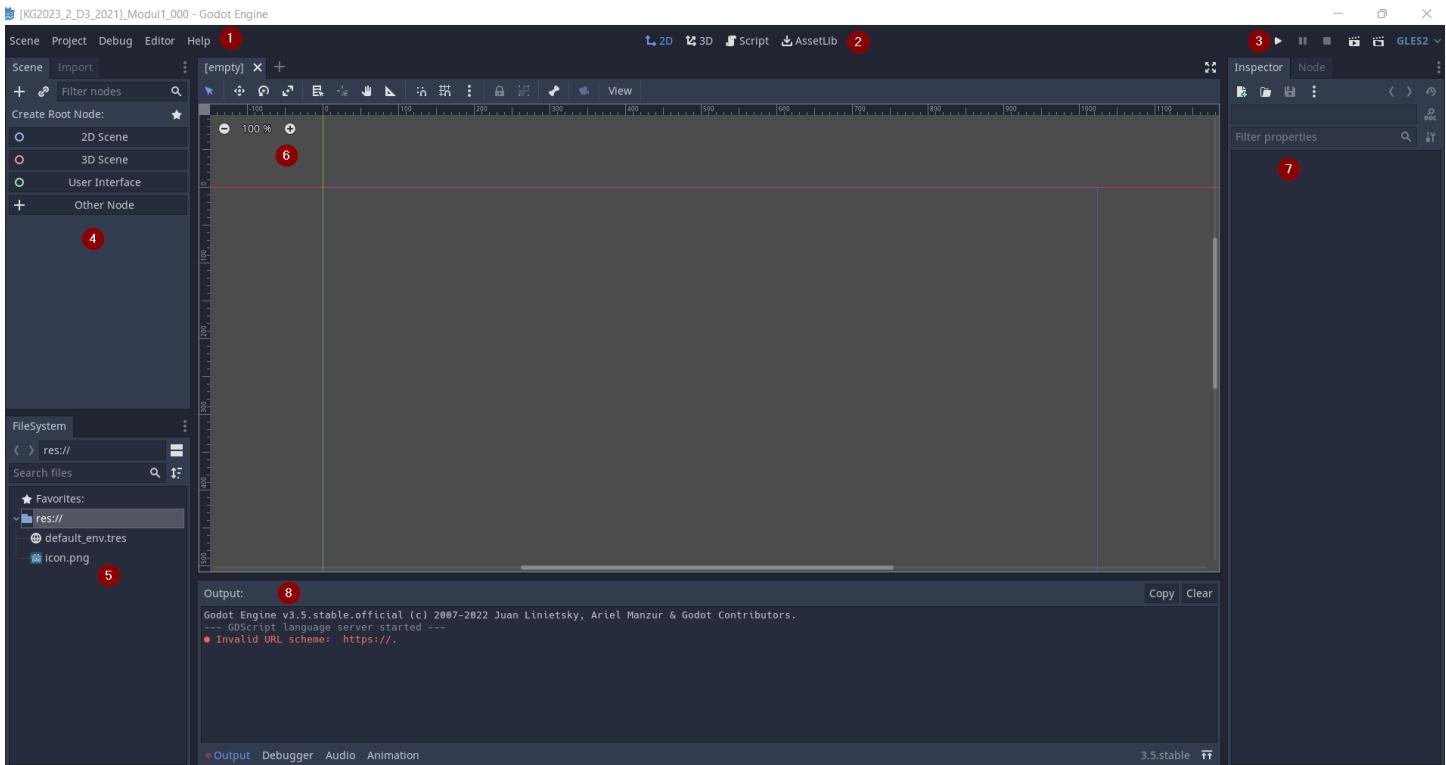
Untuk pengumpulan tugas project dikompresi menjadi **.zip**, memudahkan saat import project bagi Dosen pengampu praktikum.

2. Untuk path, disesuaikan dengan nama project, gunakan folder kosong.
 3. Untuk praktikum komputer grafik 2D, renderer yang digunakan adalah Forward+, karena karya 2D yang dibuat berbasis bentuk Dasar sederhana.
- Import



- Drag & Drop, merupakan alternatif dari import, User dapat melakukan drag drop pada file / directory proyek 2D yang didalamnya ada file project.godot atau kompresi pada karya 2D berjenis file .zip

GODOT UI



1	Main Menu Terdapat 4 item menu yaitu Scene, Project, Debug, Editor, Help. Main menu berfungsi untuk menyediakan beragam opsi seperti membuat scene baru, keluar dari project/program, atau mengubah tampilan interface
2	Workspace Workspace digunakan untuk mengganti tampilan pada viewport bisa berupa tampilan 2D, 3D, ataupun Script
3	Run/Play Digunakan untuk melakukan running, pause, stop sebuah scene
4	Scene Menunjukkan scene yang aktif
5	FileSystem Menunjukkan project file, termasuk script, gambar, audio sample, dan lainnya
6	Viewport Toolbar yang digunakan untuk bergerak, scaling, atau mengunci node scene
7	Inspector Mengizinkan kita mengedit property pada node yang ditunjuk
8	Console/Bottom Panel Digunakan sebagai melihat output, debugging, audio, dan animation

KONSEP GODOT

Scene	A scene can be a character, a weapon, a menu in the user interface, a single house, an entire level, or anything you can think of. Godot's scenes are flexible; they fill the role of both prefabs and scenes in some other game engines. Nodes A scene is composed of
Nodes	A scene is composed of one or more nodes. Nodes are your game's smallest building blocks that you arrange into trees. Here's an example of a character's nodes.
Scene Trees	All your game's scenes come together in the scene tree, literally a tree of scenes. And as scenes are trees of nodes, the scene tree also is a tree of nodes. But it's easier to think of your game in terms of

	scenes as they can represent characters, weapons, doors, or your user interface.
Signals	Nodes emit signals when some event occurs. This feature allows you to make nodes communicate without hard-wiring them in code. It gives you a lot of flexibility in how you structure your scenes.

Dokumentasi : https://docs.godotengine.org/en/stable/getting_started/introduction/key_concepts_overview.html

TUTORIAL SCENE SYSTEM : NODE 2D & CONTROL

Praktikum 2D memiliki standar antar muka untuk karya 2D atau tugas yang akan dikumpulkan. Suatu tugas memiliki 3-N Scene yaitu sbb:

1. Scene 1: Welcome Scene / Menu
2. Scene 2: Mekanik Aplikasi (Cara Kerja)
3. Scene 3: About (Identitas Mahasiswa)
4. Scene 3-N: Sesuai task yang diberikan.

Terdapat 2 Scene Mode untuk karya 2D di Godot yaitu Node2D dan Control, pada tutorial ini anda akan mempelajari membangun antar muka standar karya 2D Praktikum Komputer Grafik. Control digunakan untuk mengembangkan User Interface, sedangkan Node2D digunakan untuk mengembangkan karya 2D modifikasi.

Contoh Antarmuka:



Node2D	Dokumentasi: https://docs.godotengine.org/en/stable/classes/class_node2d.html?highlight=Node2D
--------	--

	<p>Custom drawing in a 2D node is <i>really</i> useful. Here are some use cases:</p> <ul style="list-style-type: none"> • Drawing shapes or logic that existing nodes can't do, such as an image with trails or a special animated polygon. • Visualizations that are not that compatible with nodes, such as a tetris board. (The tetris example uses a custom draw function to draw the blocks.) • Drawing a large number of simple objects. Custom drawing avoids the overhead of using a large number of nodes, possibly lowering memory usage and improving performance. • Making a custom UI control. There are plenty of controls available, but when you have unusual needs, you will likely need a custom control. <p>Custom Drawing : https://docs.godotengine.org/en/stable/tutorials/2d/custom_drawing_in_2d.html 2D Demo : https://github.com/godotengine/godot-demo-projects/tree/master/2d</p>
Control	<p>Dokumentasi: https://docs.godotengine.org/en/stable/classes/class_control.html?highlight=Control</p> <p>Inherits: CanvasItem < Node < Object</p> <p>Inherited By: BaseButton, ColorRect, Container, GraphEdit, ItemList, Label, LineEdit, NinePatchRect, Panel, Popup, Range, ReferenceRect, RichTextLabel, Separator, Tabs,TextEdit, TextureRect, Tree, VideoPlayer</p> <p>All user interface nodes inherit from Control. A control's anchors and margins adapt its position and size relative to its parent.</p> <p>GUI Demo: https://github.com/godotengine/godot-demo-projects/tree/master/gui</p>

Struktur Folder antar muka standar karya 2D minimal terdiri dari

1. Scene : Untuk menyimpan semua Scene Karya 2D
2. Script : Untuk menyimpan semua GDScript
3. Assets: Untuk menyimpan file multimedia seperti file gambar, font, dan suara.

TASK PRAKTIKUM

TASK I: HELLO WORLD

1. Load Project template [KG2023_3_D4_2021]_ModulI_000
2. Amati dan jalankan project Godot tersebut
3. Review Node control yang digunakan
4. Modifikasi gunakan identitas pribadi
5. Perhatikan Script pada template
6. Ubah Script menjadi C#

Script	Deskripsi
<pre> using Godot; public partial class MainMenu : Control { public override void _Ready() { // Tidak perlu melakukan sesuatu di _Ready untuk saat ini } private void OnBtnKarya1Pressed() { var result = GetTree().ChangeSceneToFile("res://Scenes/Karya1.tscn"); if (result != Error.Ok) { GD.Print("Scene Tidak Ada"); } } private void OnBtnKarya2Pressed() { var result = GetTree().ChangeSceneToFile("res://Scenes/Karya2.tscn"); if (result != Error.Ok) { GD.Print("Scene Tidak Ada"); } } private void OnBtnAboutPressed() { var result = GetTree().ChangeSceneToFile("res://Scenes/About.tscn"); if (result != Error.Ok) { GD.Print("Scene Tidak Ada"); } } private void OnBtnGuidePressed() { var result = GetTree().ChangeSceneToFile("res://Scenes/Guide.tscn"); if (result != Error.Ok) { GD.Print("Scene Tidak Ada"); } } private void OnBtnExitPressed() { GetTree().Quit(); } } </pre>	<p>Skrip ini mengelola navigasi dalam menu utama proyek Godot dengan beberapa tombol untuk berpindah ke scene lain atau keluar dari aplikasi. Jika terjadi kesalahan, program memberikan notifikasi di output debug.</p>

<pre> } using Godot; using System.Collections.Generic; public partial class Karya1 : Node2D { private Primitif Primitif = new Primitif(); private int screenWidth; private int screenHeight; private const int OffsetLeft = 50; private const int OffsetTop = 50; private int offsetRight; private int offsetBottom; public override void _Ready() { screenWidth = (int)GetWindow().Size.X; screenHeight = (int)GetWindow().Size.Y; offsetRight = screenWidth - OffsetLeft; offsetBottom = screenHeight - OffsetTop; QueueRedraw(); // Memicu ulang _Draw() } public override void _Draw() { Color color = new Color("#32CD30"); List<Vector2> res = new List<Vector2>(); res.AddRange(Primitif.LineDDA(OffsetLeft, OffsetTop, offsetRight, OffsetTop)); res.AddRange(Primitif.LineDDA(OffsetLeft, offsetBottom, offsetRight, offsetBottom)); res.AddRange(Primitif.LineDDA(OffsetLeft, OffsetTop, OffsetLeft, offsetBottom)); res.AddRange(Primitif.LineDDA(offsetRight, OffsetTop, offsetRight, offsetBottom)); PutPixelAll(res, color); } private void PutPixel(float x, float y, Color color) { DrawPoint(new Vector2(x, y), color); } private void PutPixelAll(List<Vector2> dots, Color color) { foreach (var dot in dots) { PutPixel(dot.X, dot.Y, color); } } } </pre>	<p>Program ini adalah skrip C# dalam proyek Godot yang menggambar bentuk menggunakan metode algoritma DDA (Digital Differential Analyzer). Skrip ini diturunkan dari kelas Node2D, yang berarti dapat menggambar langsung di layar menggunakan metode <code>_Draw()</code>. Skrip ini berfungsi untuk menggambar sebuah persegi panjang dengan warna hijau menggunakan metode DDA dalam Godot. Persegi panjang tersebut menyesuaikan ukuran layar dengan margin tertentu.</p>
---	---

<pre> } } </pre>	
<pre> extends Button func _on_BtnBack_pressed(): get_tree().change_scene_to_file("res://Scenes/Welcome.tscn") if get_tree().change_scene_to_file("res://Scenes/Welcome.tscn") != OK: print("Scene Tidak Ada") </pre>	<p>Skrip ini adalah skrip GDScript yang digunakan untuk menangani aksi tombol (button) dalam proyek Godot. Skrip ini turunan dari kelas Button, sehingga otomatis terhubung dengan tombol dalam scene.</p>
<pre> using Godot; using System.Collections.Generic; public partial class DrawArc : Node2D { private void DrawCircleArc(Vector2 center, float radius, float angleFrom, float angleTo, Color color) { int nbPoints = 32; List<Vector2> pointsArc = new List<Vector2>(); for (int i = 0; i <= nbPoints; i++) { float anglePoint = Mathf.DegToRad(angleFrom + i * (angleTo - angleFrom) / nbPoints - 90); pointsArc.Add(center + new Vector2(Mathf.Cos(anglePoint), Mathf.Sin(anglePoint)) * radius); } for (int i = 0; i < nbPoints; i++) { DrawLine(pointsArc[i], pointsArc[i + 1], color); } } public override void _Draw() { Vector2 center = new Vector2(200, 200); float radius = 80; float angleFrom = 75; float angleTo = 195; Color color = new Color(1.0f, 0.0f, 0.0f); DrawCircleArc(center, radius, angleFrom, angleTo, color); } } </pre>	<p>Skrip ini digunakan untuk menggambar busur lingkaran (arc) dalam Godot menggunakan metode <code>_Draw()</code>. Skrip ini turunan dari kelas Node2D, yang memungkinkan penggambaran langsung di layar. Skrip ini belum diedit, dan merupakan template yang diberikan.</p>
<pre> using Godot; public partial class BtnBack : Button { public void OnBtnBackPressed() { var result = GetTree().ChangeSceneToFile("res://Scenes/Welcome.tscn"); if (result != Error.Ok) { </pre>	<p>Skrip ini adalah skrip C# dalam proyek Godot yang menangani aksi tombol "Back" untuk kembali ke scene utama (Welcome.tscn). Skrip ini merupakan turunan dari kelas Button, sehingga dapat langsung digunakan pada tombol di dalam scene.</p>

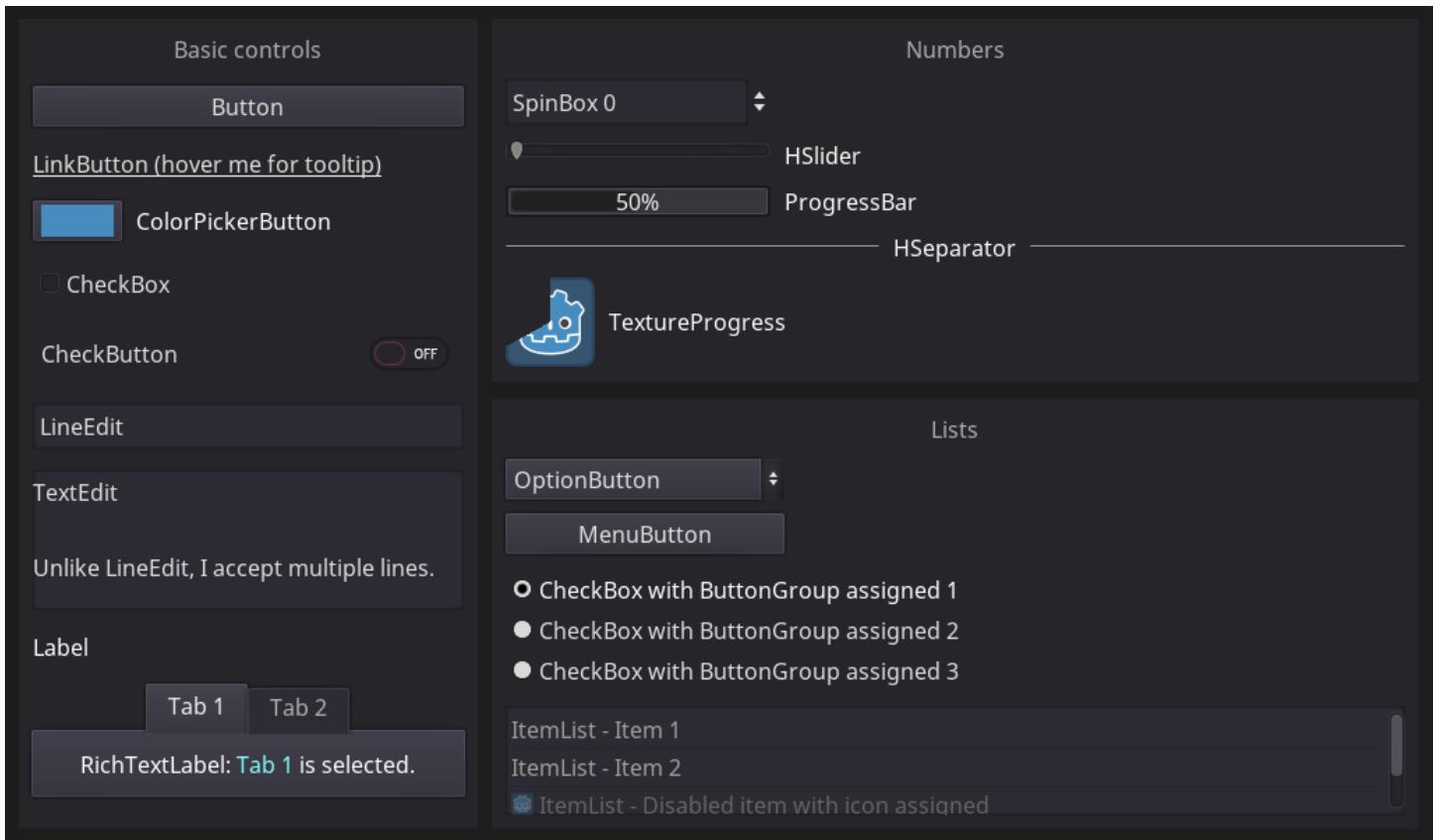
<pre> GD.Print("Scene Tidak Ada"); } } </pre>	
<pre> using Godot; using System; public partial class about : Node { // Called when the node enters the scene tree for the first time. public override void _Ready() { } // Called every frame. 'delta' is the elapsed time since the previous frame. public override void _Process(double delta) { } } </pre>	<p>Skrip ini adalah skrip C# dalam proyek Godot yang berfungsi sebagai kontrol dasar untuk scene "About". Skrip ini merupakan turunan dari Node, yang berarti berfungsi sebagai node umum tanpa fitur tampilan atau input langsung.</p>
<pre> using Godot; public partial class SceneController : Control { public override void _Ready() { // Kosong, sesuai dengan GDScript } private void OnBtnBackPressed() { var result = GetTree().ChangeSceneToFile("res://Scenes/Welcome.tscn"); if (result != Error.Ok) { GD.Print("Scene Tidak Ada"); } } } </pre>	<p>Skrip ini berfungsi sebagai pengontrol scene untuk bagian "Guide" dalam proyek Godot. Skrip ini merupakan turunan dari Control, yang berarti dirancang untuk menangani UI dalam scene.</p>

Dokumentasi:

Modifikasi

1. https://docs.godotengine.org/en/stable/tutorials/ui/gui_using_fonts.html
2. <https://docs.godotengine.org/en/stable/tutorials/ui/index.html>
3. https://docs.godotengine.org/en/stable/tutorials/ui/size_and_anchors.html#/layout-presets
4. https://docs.godotengine.org/en/stable/tutorials/ui/gui_containers.html
5. https://github.com/godotengine/godot-demo-projects/tree/master/gui/control_gallery
- 6.





Assets

1. <https://github.com/heylalexj/awesome-images>
2. <https://fonts.google.com/>

TASK II: C# : BASIC

Ubah GDScript dengan C#

Code	Gdscript	C#
Komentar	<pre># Single-line comments are written using hash symbol. """ Multi-line comments are written using docstrings. """</pre>	<pre>// Single line comments /* Multi line comments For something like a Long comments */</pre>
Variabel	<pre>var x = 8 # int var y = 1.2 # float var b = true # bool var s = "Hello World!" # String var a = [1, false, "brown fox"] # Array</pre>	<pre>// Variabel int x = 8; // int float y = 1.2f; // float bool b = true; // bool string s = "Hello World!"; // String object[] a = { 1, false, "brown fox" }; // Array</pre>
Control Flow	<pre>if x < y: print("x is smaller than y") elif x > y: print("x is bigger than y") else: print("x and y are equal")</pre>	<pre>if (x < y) { GD.Print("x is smaller than y"); } else if (x > y) { GD.Print("x is bigger than y"); }</pre>

		<pre> } else { GD.Print("x and y are equal"); } </pre>
Loop	<pre> for i in range(20): print(i) for i in ["two", 3, 1.0]: print(i) while x > y: print(x, y) y += 1 </pre>	<pre> for (int i = 0; i < 20; i++) { GD.Print(i); } foreach (var i in new object[] { "two", 3, 1.0 }) { GD.Print(i); } while (x > y) { GD.Print(x, y); y += 1; } </pre>
Switch Case	<pre> match x: 1: print("Match is similar to switch.") 2: print("However you don't need to put cases before each value.") 3: print("Furthermore each case breaks on default.") break # ERROR! Break statement is unnecessary! 4: print("If you need fallthrough use continue.") continue _: print("Underscore is a default case.") </pre>	<pre> switch (x) { case 1: GD.Print("Match is similar to switch."); break; case 2: GD.Print("However you don't need to put cases before each value."); break; case 3: GD.Print("Furthermore each case breaks on default."); break; // Diperlukan di C# case 4: GD.Print("If you need fallthrough use continue."); goto case 5; // Simulasi "continue" dalam switch default: GD.Print("Underscore is a default case."); break; } </pre>
Function	<pre> func add(first, second): return first + second func printing(): print("GDScript ", "is ", " awesome.") prints("These", "words", "are", "divided", "by", "spaces.") printt("These", "words", "are", "divided", "by", "tabs.") printraw("This gets printed to system console.") </pre>	<pre> public int Add(int first, int second) { return first + second; } public void Printing() { GD.Print("C# ", "is ", "awesome."); GD.Print("These", "words", "are", "divided", "by", "spaces."); GD.Print("These", "words", "are", "divided", "by", "tabs."); GD.PrintRaw("This gets printed to system console."); } </pre>

Dokumentasi GDScript:

1. <https://learnxinyminutes.com/docs/c++/>
2. <https://learnxinyminutes.com/docs/gdscript/>
3. <https://learnxinyminutes.com/csharp/>
4. https://docs.godotengine.org/en/stable/getting_started/step_by_step/scripting_languages.html#doc-scripting
5. https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_basics.html
6. https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_advanced.html
7. https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_exports.html
8. https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_styleguide.html
9. https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/static_typing.html
10. https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_format_string.html

11. https://docs.godotengine.org/en/stable/tutorials/scripting/c_sharp/c_sharp_style_guide.html
12. https://docs.godotengine.org/en/stable/tutorials/scripting/c_sharp/c_sharp_basics.html
13. https://docs.godotengine.org/en/stable/tutorials/scripting/c_sharp/c_sharp_features.html
14. https://docs.godotengine.org/en/stable/tutorials/scripting/c_sharp/index.html#toc-learn-scripting-c
15. https://docs.godotengine.org/en/stable/tutorials/scripting/c_sharp/c_sharp_differences.html
16. <https://learn.microsoft.com/en-us/dotnet/api/system.environment.stacktrace?view=net-8.0>
17. <https://github.com/godotengine/godot-demo-projects/tree/master/mono>
- 18.

Tugas Task I

1. [How to solve Square and Hollow pattern programs in Python](#)
2. [Solve any Star Pattern program in Python](#)

Implementasi 2-3 Pattern Konversikan Solusi dari Python ke GDScript (print di console)

Lesson Learnt

1. Perbedaan Sintaks dan Paradigma

GDScript lebih sederhana dan langsung, sementara C# lebih ketat dengan tipe data dan membutuhkan struktur yang lebih eksplisit. Memahami keduanya memberi fleksibilitas dalam memilih bahasa yang sesuai dengan kebutuhan proyek.

2. Penggunaan [Tool] di C#

Atribut [Tool] memungkinkan script berjalan di editor Godot, mempermudah pengaturan properti langsung dari panel Inspector tanpa harus menjalankan game terlebih dahulu.

3. Penamaan yang Jelas dan Unik

Gunakan konvensi penamaan yang konsisten untuk kelas, variabel, dan node agar terhindar dari konflik atau ambiguitas dalam kode, terutama saat bekerja dalam tim atau proyek besar.

4. Sistem Manajemen Scene yang Terstruktur

Manfaatkan **PackedScene** untuk memuat dan mengganti scene secara dinamis. Ini memungkinkan sistem yang lebih modular dan fleksibel dalam manajemen scene.

5. Struktur Folder yang Terorganisir

Susun folder dengan baik sejak awal proyek, seperti memisahkan scene, script, asset, dan UI. Struktur yang rapi mempermudah navigasi dan pemeliharaan kode dalam jangka panjang.

PENGUMPULAN

Ikuti Format yang diberikan di Google Classroom.