Modul 1-3



Komputer Grafik 2D

Godot Part 1

071 –Tsabit – 22/08/25

Contents

[Setup Godot 3](#_Toc206435283)

[Godot Project Manager 5](#_Toc206435284)

[Godot Scene UI 8](#_Toc206435285)

[Konsep Godot 8](#_Toc206435286)

[Praktikum Template Modul 1: Membuat Menu 9](#_Toc206435287)

[Task Praktikum 11](#_Toc206435288)

[Task 1: Hello Godot 11](#_Toc206435289)

[Task 2: Modifikasi Menu 12](#_Toc206435290)

[Line Drawing Algorithm 14](#_Toc206435291)

[Line Dda 14](#_Toc206435292)

[Bersenham 15](#_Toc206435293)

[Fungsi Perubahan Kartesian Koordinat 17](#_Toc206435294)

[Fungsi World Koordinat 17](#_Toc206435295)

[Task 1: Manipulasi Garis 17](#_Toc206435296)

[Task 2: Bentuk Dasar Sederhana 18](#_Toc206435297)

[Lingkaran 20](#_Toc206435298)

[MIDPOINT CIRCLE ALGORITHM 21](#_Toc206435299)

[Ellips 24](#_Toc206435300)

[Midpoint Ellips Algorithm 25](#_Toc206435301)

[TASK PRAKTIKUM 31](#_Toc206435302)

[Task 1: Implementasi Midpoint Lingkaran dan Midpoint Garis dan lengkapi bentuk dasar 31](#_Toc206435303)

[Task 2: Modifikasi Garis 31](#_Toc206435304)

Modul 1 Pengenalan Godot

# 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 mengunakan 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.

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

A screenshot of a computer

Description automatically generated

Gambar 1 Halaman download Godot

1. Download Versi .Net, pada praktikum Komputer grafik 2D Bahasa yang digunakan adalah C#

A screenshot of a computer

Description automatically generated

Gambar 2 Halaman download .net

1. Install dotnet versi 8 LTS <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.

A screenshot of a computer

Description automatically generated

Gambar 3 Verifikasi dotnet.exe terinstall

1. Silakan extract file Godot\_v4.3, anda dapat menggunakan 7zip, atau tools lainnya.

A screenshot of a computer

Description automatically generated

Gambar 4 Hasil extract file Godot 4.4.1

1. Jalankan file executable Godot\_4.3, anda sudah dapat mengoperasikan Game Engine Godot.
2. Jika komputer/laptop anda mengalami kendala saat menjalankan Godot versi 4.3 silakan untuk mendowngrade versi Godot yang digunakan; rekomendasi untuk praktikum gunakan Godot 4.3 Anda dapat mendownload versi sebelumnya pada tautan berikut: <https://godotengine.org/download/3.x/windows/>

# Godot Project Manager

Project Manager adalah antar muka pertama setelah mengeksekusi Godot, berfungsi untuk melakukan menajemen 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.

A screenshot of a computer

Description automatically generated

Gambar 5 Project Manager UI

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

A screenshot of a computer

Description automatically generated

Gambar 6 Scan Folder Project Manager

* New Project

A screenshot of a computer

Description automatically generated

Gambar 7 Project Manager UI

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\_D3\_2024]\_Modul1\_001

|  |  |
| --- | --- |
| Kode | Penjelasan |
| KG2025 | Komputer Grafik tahun 2025 |
| 2A/B/C | Kelas 2 |
| D3 | Prodi D3 |
| 2024 | Angkatan |
| Modul1 | No Modul |
| 001 | 3 Digit Nim Terakhir |

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

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

* Import

Drag & Drop, merupakan alternative 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

A screenshot of a computer

Description automatically generated

Gambar 8 Import UI Project Manager

# Godot Scene UI

A screenshot of a computer

Description automatically generated with medium confidence

Gambar 9 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 | Create Root Node  Bagian untuk membuat Node utama dari scene., terdiri dari 2D,3D,UI,Other scene |
| 5 | FileSystem Dock  Menampilkan semua file project |
| 6 | Viewport  Area kerja utama tempat untuk menyusun kode |
| 7 | Node Dock  Menampilkan hierarki node dalam scene yang sedang aktif |
| 8 | Console/Bottom Panel  Digunakan sebagai melihat output, debugging, audio, dan animation |

# Konsep Godot

|  |  |
| --- | --- |
| Scene | Satu unit/potongan game di Godot. Bisa berupa level, karakter, UI, atau objek kecil. Scene bisa disusun dari banyak node dan dapat dipakai ulang. |
| Nodes | Elemen dasar di Godot. Setiap node punya fungsi tertentu (contoh: Sprite2D untuk gambar, AudioStreamPlayer untuk suara, Button untuk UI). |
| Scene Trees | Struktur hierarki yang menyusun scene. Node utama (root) berada di atas, node lainnya menjadi anak. Hierarki ini menentukan bagaimana node berinteraksi. |
| Signals | istem event di Godot. Memungkinkan node mengirim pesan/aksi ke node lain tanpa harus saling terhubung langsung. Contoh: tombol (Button) mengirim signal pressed(). |

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

|  |  |
| --- | --- |
| Node2D | Dokumentasi: <https://docs.godotengine.org/en/stable/classes/class_node2d.html?highlight=Node2D>    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> |
| Control | Dokumentasi: <https://docs.godotengine.org/en/stable/classes/class_control.html?highlight=Control>    GUI Demo: <https://github.com/godotengine/godot-demo-projects/tree/master/gui> |

# Praktikum Template Modul 1: Membuat Menu

Praktikum 2D memliki 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 4-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:

A picture containing text, electronics, screenshot, computer

Description automatically generated

Gambar 10 Contoh Menu Praktikum

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.

**Dokumentasi:**

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>

A screenshot of a computer program

Description automatically generated

Assets

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

# Task Praktikum

## Task 1: Hello Godot

1. Load Project template [KG2025\_2A\_D4\_2024]\_Modul1\_000
2. Amati dan jalankan project Godot tersebut
3. Ubah Name pada Project Setting sesuai 3 digit NIM terakhir anda

A screenshot of a computer

Description automatically generated

Gambar 11 Contoh Menu Praktikum

1. Review Node control yang digunakan
2. Modifikasi gunakan identitas pribadi
3. Perhatikan Script pada template

|  |
| --- |
| Dari Task 1 ini, saya mempelajari:   * Cara membuka dan menjalankan project template di Godot. * Bagaimana mengubah Project Settings sesuai kebutuhan. * Pengenalan dasar Node Control seperti Label, Button, dan Container. * Cara melakukan modifikasi scene menggunakan identitas pribadi (nama & NIM). * Memahami bagaimana signal dan script bekerja untuk menghubungkan interaksi UI dengan kode. * Konsep dasar Godot: project → scene → node → script. * Cara setup godot engine (installer) * Mengalami sedikit masalah namun bisa diselesaikan |

## Task 2: Modifikasi Menu

1. Perhatiakan gambar 10, berikut adalah gambaran menu yang akan digunakan untuk praktikum komputer grafik
2. Modifikasi menu dengan assets, style pilihan mahasiswa.
3. Struktur Scene sbb:
   1. Scene 1: Welcome Scene / Menu
   2. Scene 2: Mekanik Aplikasi (Cara Kerja)
   3. Scene 3: About (Identitas Mahasiswa)
   4. Scene 4-N: Sesuai task yang diberikan.

|  |
| --- |
| Praktikum ini bertujuan untuk melatih saya dalam merancang dan mengimplementasikan antarmuka pengguna (UI) yang interaktif dan dinamis menggunakan struktur *multi-scene*. Saya akan memodifikasi menu yang diberikan (bertema motor custom/caferacer) dan mengembangkan empat *scene* utama: Menu Utama, Mekanisme Aplikasi, Tentang, dan *Scene* untuk tugas selanjutnya.  **Tahap 1: Perencanaan dan Analisis Desain**  **1. Analisis Tema & Gaya**   * Saya akan memperhatikan gambar menu utama yang diberikan (gambar 10, yaitu menu motor caferacer). Saya akan mengidentifikasi elemen-elemen kuncinya:   + **Skema Warna:** Dominasi warna gelap (hitam, abu-abu), aksen metalik (perak, chrome), dan warna cerah untuk tombol aktif (merah, kuning).   + **Tipografi (Font):** Font bergaya *vintage* atau *grunge* untuk judul, dan font sans-serif yang bersih untuk teks tombol.   + **Elemen Desain:** Panel metal, indikator (seperti speedometer), tombol fisikal, dan latar belakang yang dinamis.   **2. Konsep Desain & Modifikasi**   * Saya akan menentukan aset, gaya, dan palet warna yang akan saya gunakan. Saya akan mempertahankan tema motor *caferacer*, namun menambahkan sentuhan personal. * Saya akan membuat sketsa kasar untuk setiap *scene* pada selembar kertas atau aplikasi desain digital.   + **Scene 1 (Menu Utama):** Tata letak tombol "Start", "Mekanisme", "About", "Keluar". Saya akan memikirkan bagaimana setiap tombol akan terlihat (normal, *hover*, *pressed*).   + **Scene 2 (Mekanisme):** Tata letak untuk teks yang menjelaskan cara kerja aplikasi. Di mana tombol "Kembali" akan saya tempatkan?   + **Scene 3 (About):** Desain untuk menampilkan identitas diri saya (Nama, NIM).   + **Scene 4:** Saya akan merencanakan elemen utama untuk *scene* ini (misalnya: *scene* 3D, galeri, dll.).   **Tahap 2: Persiapan Aset Grafis**   * **Aset Visual:** Saya akan mencari atau membuat sendiri aset-aset berikut sesuai dengan konsep desain saya.   + **Latar Belakang (Background):** Sebuah gambar berkualitas tinggi (misalnya: jalanan, garasi motor). Ukuran yang ideal adalah 1920x1080 piksel (untuk resolusi Full HD).   + **Logo/Judul:** Gambar judul "Main Menu" dalam format PNG dengan latar belakang transparan.   + **Tombol (Buttons):** Saya akan menyiapkan tiga variasi gambar untuk setiap tombol:     1. **Normal:** Tombol dalam keadaan default.     2. **Hover:** Tombol saat kursor berada di atasnya.     3. **Pressed:** Tombol saat ditekan.   + **Ikon:** Ikon-ikon kecil untuk melengkapi tombol (misalnya: ikon roda gigi untuk pengaturan, ikon kunci pas untuk mekanik).   **Persegi Panjang**  Logika kode Anda sudah benar, yaitu menggambar empat sisi menggunakan algoritma **Bresenham**. Namun, untuk mencegah tumpang tindih piksel dan menjaga kode lebih bersih, versi berikut lebih disarankan.  **SegitigaSiku**  Kode Anda sudah **tepat** dan **efisien**. Kode ini secara logis menggambar tiga sisi yang membentuk segitiga siku-siku dengan memanggil LineBresenham untuk setiap sisi: alas, tinggi, dan sisi miring.  Kode Anda sudah menunjukkan pemahaman yang baik dalam menggunakan algoritma Bresenham untuk membangun bentuk-bentuk dasar. Dengan logika yang sama, Anda dapat melanjutkan untuk membuat bentuk-bentuk yang lebih kompleks seperti Trapesium. |

Modul 2 Primitif 2D

# Line Drawing Algorithm

A close-up of a document

Description automatically generated with medium confidence

A picture containing chart

Description automatically generated

# Line Dda

A picture containing text, antenna

Description automatically generated

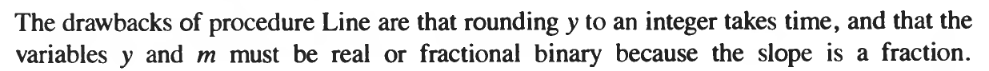
Text

Description automatically generated

Text, letter

Description automatically generated

# Bersenham





Text, letter

Description automatically generated

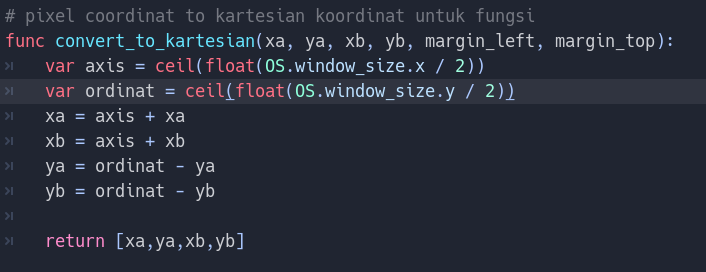
A picture containing text, antenna

Description automatically generated

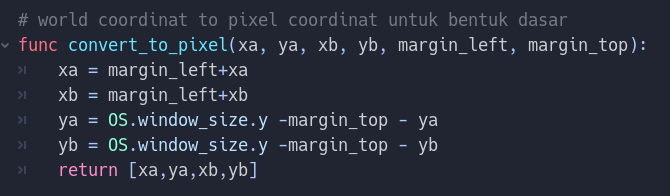
Table

Description automatically generated

## Fungsi Perubahan Kartesian Koordinat



## Fungsi World Koordinat



## Task 1: Manipulasi Garis

**Tugas Task 1**

1. Pelajari implementasi LineDDA dan Line Bresenham
2. Konversi fungsi Kartesian Koordinat dan World Koordinat
3. Tambahkan sebuah fungsi menggunakan rumus lain seperti absolute, exponensial, limit dll pada Karya 1 pada kuadran 1
4. Tuliskan Lesson Learnt Task 1

|  |
| --- |
| Lesson Learnt |
| **Yang berhasil dipelajari:**   1. **Algoritma LineDDA vs Bresenham**: LineDDA menggunakan floating point sehingga lebih akurat untuk kurva halus, sedangkan Bresenham lebih efisien dengan integer arithmetic 2. **Konversi Koordinat**: Fungsi ToScreenCoordinate() penting untuk mengubah koordinat matematika (0,0 di kiri bawah) ke koordinat layar Godot (0,0 di kiri atas) 3. **Step Size**: Parameter step yang terlalu besar membuat kurva terputus-putus, terlalu kecil membuat rendering lambat   **Tantangan yang dihadapi:**   * Overflow pada fungsi eksponensial dengan nilai x besar * Logaritma tidak terdefinisi untuk x ≤ 0, perlu validasi input * Perlu penyesuaian scaling agar kurva terlihat proporsional di layar   **Perbaikan yang dilakukan:**   * Menambahkan pembatasan range input untuk mencegah overflow * Menggunakan faktor skala (seperti /10 atau \*10) untuk menyesuaikan bentuk kurva * Validasi input untuk fungsi yang memiliki domain terbatas |

## Task 2: Bentuk Dasar Sederhana

**Tugas Task 2**

1. Perhatikan Contoh Bentuk Sederhana Persegi dan Persegi Panjang
2. Lengkapi fungsi pembentuk bentuk dasar, segi tiga siku-siku, trapesium siku, segitiga sama kaki, trapesium sama kaki.
3. Sebelum melakukan programming, lakukan dahulu perhitungan menggunakan google sheet atau geogebra dan sejenisnya untuk mempermudah membuat bentuk dasar, bentuk dasar dibuat parametric bukan secara manual.
4. Konversi fungsi Kartesian Koordinat dan World Koordinat, buat 4 di masing kuadran bentuk dasar itu.

|  |
| --- |
| Lesson Learnt |
| 1. **Perhitungan Parametrik**: Bentuk geometri harus dihitung dengan rumus matematika, bukan koordinat manual. Contoh: untuk segitiga sama kaki, puncak berada di (x + alas/2, y + tinggi) 2. **Simetri Geometri**: Trapesium sama kaki memerlukan offset (panjangBawah - panjangAtas)/2 untuk membuatnya simetris 3. **Koordinat 4 Kuadran**:    * Kuadran I: x positif, y positif    * Kuadran II: x negatif, y positif    * Kuadran III: x negatif, y negatif    * Kuadran IV: x positif, y negatif   **Tantangan yang dihadapi:**   * Menghitung titik-titik trapesium sama kaki yang simetris * Memastikan bentuk tidak overlap saat ditempatkan di kuadran berbeda * Debugging bentuk yang tidak sesuai ekspektasi * Error overload fungsi   **Perbaikan yang dilakukan:**   * Menggunakan GeoGebra untuk memverifikasi perhitungan geometri sebelum coding * Membuat fungsi helper untuk menghitung offset dan titik tengah * Testing per bentuk sebelum menggabungkan semuanya * Memerhatikan kembali di fungsi primitif nya |
|  |

Modul 3 Lingkaran dan Ellips

## Lingkaran

Lingkaran merupakan bentuk dasar yang biasa digunakan untuk membuat gambar atau objek yang kompleks, seperti dekorasi, batik, dan lain-lain. Pada paket library komputer grafik, sebagian atau lingkaran penuh dapat dibuat menggunakan sebuah prosedur. Secara general, prosedur tersebut dapat menghasilkan garis dan ellips.

|  |  |
| --- | --- |
| A diagram of a circle with a line  Description automatically generated | A diagram of a circle with a point  Description automatically generated |

Persamaan Lingkaran

Persamaan Lingkaran Polar Coordinates

Step size :

Komputasi lingkaran dapat direduksi karena lingkaran adalah bentuk yang simetris. Setiap kuadran meliliki bentuk bagian lingkaran yang sama.

|  |
| --- |
| A diagram of a circle with lines and dots  Description automatically generated |

Algoritma paling efisien berdasarkan kalkulasi incremental dari decision parameter, seperti bersenham line algorithm, yang hanya membutuhkan operasi integer sederhana. Algoritma line bersenham diadaptasi untuk pembentukan lingkaran dengan melakukan setup decision parameter untuk mencari pixel terdekat untuk mendapatkan lingkar untuk setiap sampling step.

Metode untuk mendapatkan jarak secara langsung pada lingkaran, adalah dengan melakukan pengecekan posisi tengah dari dua pixel untuk menentukan apakah titik “midpoint” ini ada pada dalam atau luar lingkaran. Metode midpoint dapat diaplikasikan untuk bentuk-bentuk conics.

# MIDPOINT CIRCLE ALGORITHM

Seperti pada line bersenham, tujuan dari algoritma untuk mendapatkan sampling titik dan menentukan pixel terdekat pada setiap step. Pada lingkaran, untuk setiap dan titik posisi center . Algoritma dimulai dengan mengkalkulasi posisi pixel dalam jalur lingkaran yang memiliki titik tengah origin . Lalu untuk setiap posisi yang dikalkulasi dipindahkan pada posisi yang benar dengan menambahkan pada dan pada .

Pada bagian lingkaran kuadran 1 mulai dari , nilai slope bervariasi dari sampai . Maka pergerakan unit step sesuai arah x positif dan menggunakan decision parameter untuk menentukan dua posisi yang mungkin lebih dekat dengan jalur lingkaran untuk setiap langkah.

Jika point ada didalam interior lingkaran , fungsi lingkaran negative, dan sebaliknya.

|  |
| --- |
| A graph of a line with a circle and lines  Description automatically generated with medium confidence |

Jika midpoint terletak pada bagian dalam lingkaran dan lebih dekat pada batas lingkaran. Sebaliknya lebih dekat pada batas lingkaran.

Decision parameter selanjutnya didapatkan menggunakan kalkulasi incremental integer, yaitu

Dimana, antara atau , bergantung pada tanda dari

Evaluasi dan didapatkan secara inkemental menggunakan.

Pada titik awal , decision parameter pertama didapatkan dengan melakukan evaluasi pada fungsi lingkaran dengan nilai :

A paper with math equations

Description automatically generated

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| Implementasi Circle Algorithm menggunakan C# pada bentukdasar.cs |
| Silakan lengkapi |

|  |
| --- |
| Contoh Pemanggilan Lingkaran yang akan digunakan pada karya1 |
|  |

## Ellips

Untuk menggambarkan sebuah ellips, kita dapat mengadopsi pola penggambaran lingkaran. Ellips bisa juga disebut lingkaran yang pipih, dengan modifikasi lingkaran yang memiliki dimensi vertikal dan horizontal yang berbeda atau disebut major dan minor axes.

|  |
| --- |
| A diagram of a graph  Description automatically generatedA graph of ellipse and a line  Description automatically generated |

Ellips dapat didefinisikan sebagai Kumpulan dari titik yang sedemikian hingga jumlah dari jarak antara dua titik tetap (foci atau fixed position) adalah sama untuk seluruh titik. Jika jarak dari dua foci dari sebuah titik pada sebuah ellips diberi label dan , maka persamaan ellips dapat dinyatakan sebagai:

Eksprsi dari jarak d1 dan d2 dalam foci dan maka didapatkan :

Persamaan ellips lain ketika posisi mayor axes dan minor axes pada posisi standar dapat dinyatakan sebagai:

Dengan menggunakan koordinat polar ellips pada posisi standar dapat dinyatakan dengan:

## Midpoint Ellips Algorithm

Seperti pada penggambaran lingkaran, Proses sampling pada satu unit koordinat ( atau ) dan menentukan nilai integer terdekat yang sesuai dengan jalur garis dari koordinat lain.

Metode midpoint ellips diaplikasikan pada quadran 1 untuk dua bagian yaitu Region I (Mayor axes) dan Region II (Minor Axis)

Steps atau iterasi dilakukan pada arah x jika sebaliknya iterasi dilakukan pada arah y jika . Titik awal pada posisi dan arah mengikuti jarum jam atau (CW) pada kuadran I ellips. Perubahan unit step dari x ke unit step y ketika . Lalu, mirip dengan lingkaran ada property simetris pada penggambaran ellips, secara parallel untuk dua region.

|  |  |
| --- | --- |
| A graph of a mathematical equation  Description automatically generated with medium confidence | A diagram of a circle with a point in the center  Description automatically generated |

Didefinisikan fungsi ellips pada sebuah titik sebagai berikut:

Yang mengikuti syarat-syarat sebagai berikut:

fungsi ellipse berfungsi sebagai decision parameter pada algoritma midpoint. Untuk setiap posisi sampling, pemilihan pixel selanjutnya pada jalur ellips tergantung pada kondisi dari fungsi ellips yang dievaluasi pada midpoint untuk dua kandidat pixel.

Ellipse slope atau m didapatkan dari :

Pada boundary antara region 1 dan region 2, nilai dari dan

Maka, pergantian region terjadi saat

|  |
| --- |
| A diagram of a graph  Description automatically generated |

Atau di expand sbb:

Subsitusi antara atau bergantung pada tanda , decision parameter di inkrement sebagai berikut:

Pada region 1 nilai pertama pada decision paramater didapatakn dengan mengevaluasi fungsi ellips pada posisi awal

Atau

|  |
| --- |
| A close-up of a graph  Description automatically generated |

Untuk region 2, sampling unit step terjadi pada arah y negatif, dan midpoint diambil diantara dua titik kandidat horizontal pixel untuk setiap step. Maka decision parameter dievaluasi sbb:

Atau

Para region 2, nilai pertama adalah diambil berdasarkan posisi terakhir dari region 1 dan dicision parameter pertama region ke 2 adalah.

Untuk menyederhanakan perhitungan, titik pertama dapat diambil pada . Unit step akan berjalan dengan arah y positif sampai dengan posisi terakhir dari Region 1.

A white paper with black text

Description automatically generated

A paper with text and equations

Description automatically generated with medium confidence

A white paper with black text

Description automatically generated

A screenshot of a computer program

Description automatically generated

|  |
| --- |
| Implementasi Ellips Algorithm pada C# pada bentukdasar.cs |
| Silakan lengkapi |

|  |
| --- |
| Contoh Pemanggilan Ellips pada karya1 |
|  |

# TASK PRAKTIKUM

### Task 1: Implementasi Midpoint Lingkaran dan Midpoint Garis dan lengkapi bentuk dasar

1. Lanjutkan kode minggu lalu dan rename menjadi [KG2025\_2A\_D4\_2023]\_Modul3\_000, modifikasi scene karya1 dan karya 2 sesuai dengan tugas modul 3.
2. Amati Algoritma Midpoint untuk menggambar lingkaran dan ellips
3. Implementasi Algoritma tersebut dan buatlah fungsi lingkaran / circle dan ellips / ellipse pada class bentukdasar.cs
4. Karya 2 pada pertemuan minggu lalu dipindahkan menjadi karya 1, selain Persegi, Persegi Panjang, Segitiga Siku-Siku, dan Trapesium Siku-Siku, tambahkan Lingkaran dan Ellips di sembarang kuadran

|  |
| --- |
| Lesson Learnt (Code, Print Screen Hasil Karya, dan Komentar) |
| * Gunakan Kelas Abstraksi yang Tepat   Untuk menggambar bentuk dasar seperti lingkaran dan elips, gunakan method dari kelas BentukDasar, bukan langsung dari Primitif.  Kelas BentukDasar sudah menyediakan method seperti Lingkaran dan Elips yang mengelola pemanggilan ke algoritma di Primitif.   * Pastikan Instance yang Dipakai   Di kode, instance yang benar untuk pemanggilan adalah \_bbentukDasar (bukan \_bentukDasar yang bertipe Primitif).  Contoh:  var lingkaran = \_bbentukDasar.Lingkaran(...);  var elips = \_bbentukDasar.Elips(...);   * Perhatikan Parameter dan Tipe Data   Pastikan parameter yang dikirim sesuai dengan definisi method, misal Vector2 untuk titik pusat dan int untuk radius.   * Manfaatkan Utility dan Konversi Koordinat   Untuk penempatan objek di layar, gunakan fungsi konversi koordinat seperti ScreenUtils.ToScreenCoordinate agar posisi sesuai dengan sistem koordinat aplikasi.   * Hindari Duplikasi Instance   Tidak perlu membuat dua instance untuk bentuk dasar (\_bentukDasar dan \_bbentukDasar). Cukup satu instance BentukDasar saja untuk semua pemanggilan.   * Error Handling   Jika muncul error seperti CS1061, cek apakah method sudah ada di kelas yang digunakan dan apakah instance yang dipakai sudah benar.  Kesimpulan:  Selalu gunakan kelas dan method sesuai desain arsitektur aplikasi, pahami struktur inheritance dan komposisi, serta pastikan pemanggilan method sesuai dengan tipe instance yang digunakan. |

### Task 2: Modifikasi Garis

1. Buatlah berbagai macam attribute garis seperit titik-titik, titik-garis-titik, garis-garis dan lain-lain lalu buatlah scene karya2 seperti berikut

Shape

Description automatically generated with medium confidence

1. Buatlah Scene baru, karya3 yang merupakan Modifikasi Bentuk Dasar Persegi, Persegi Panjang, Segitiga Siku-Siku, dan Trapesium Siku-Siku, Lingkaran dan Ellips yang attribute garisnya sesuai tipe yang diiginkan, normal, titik-titik, titik-garis-titik, garis-garis.
2. Contoh Scene sbb:

Graphical user interface

Description automatically generated with low confidence

A picture containing graphical user interface

Description automatically generated

A picture containing text

Description automatically generated

Shape, polygon

Description automatically generated

|  |
| --- |
| Lesson Learnt (Code, Print Screen Hasil Karya, dan Komentar) |
| **Yang berhasil dipelajari:**   1. **Pattern Drawing**: Menggunakan modulo operator i % (gap + 1) untuk membuat pattern berulang 2. **Style Differentiation**: Setiap bentuk (lingkaran, ellipse, garis) memerlukan treatment berbeda karena cara generate titiknya berbeda 3. **Visual Impact**: Gap dan strip length sangat mempengaruhi estetika hasil akhir   **Tantangan yang dihadapi:**   * Menentukan parameter gap yang tepat untuk setiap style * Sinkronisasi pattern antara berbagai bentuk geometri * Optimasi performa saat rendering banyak titik dengan style kompleks   **Perbaikan yang dilakukan:**   * Menggunakan parameter yang dapat disesuaikan untuk setiap pemanggilan * Membuat mapping khusus untuk circle dan ellipse style * Testing visual untuk menemukan kombinasi parameter terbaik   **Insight Penting:**   * Style garis sangat mempengaruhi persepsi visual bentuk * Pattern yang terlalu rapat atau renggang mengurangi recognizability bentuk * Konsistensi style across berbagai bentuk geometri penting untuk estetika keseluruhan |

Use the "Insert Citation" button to add citations to this document.