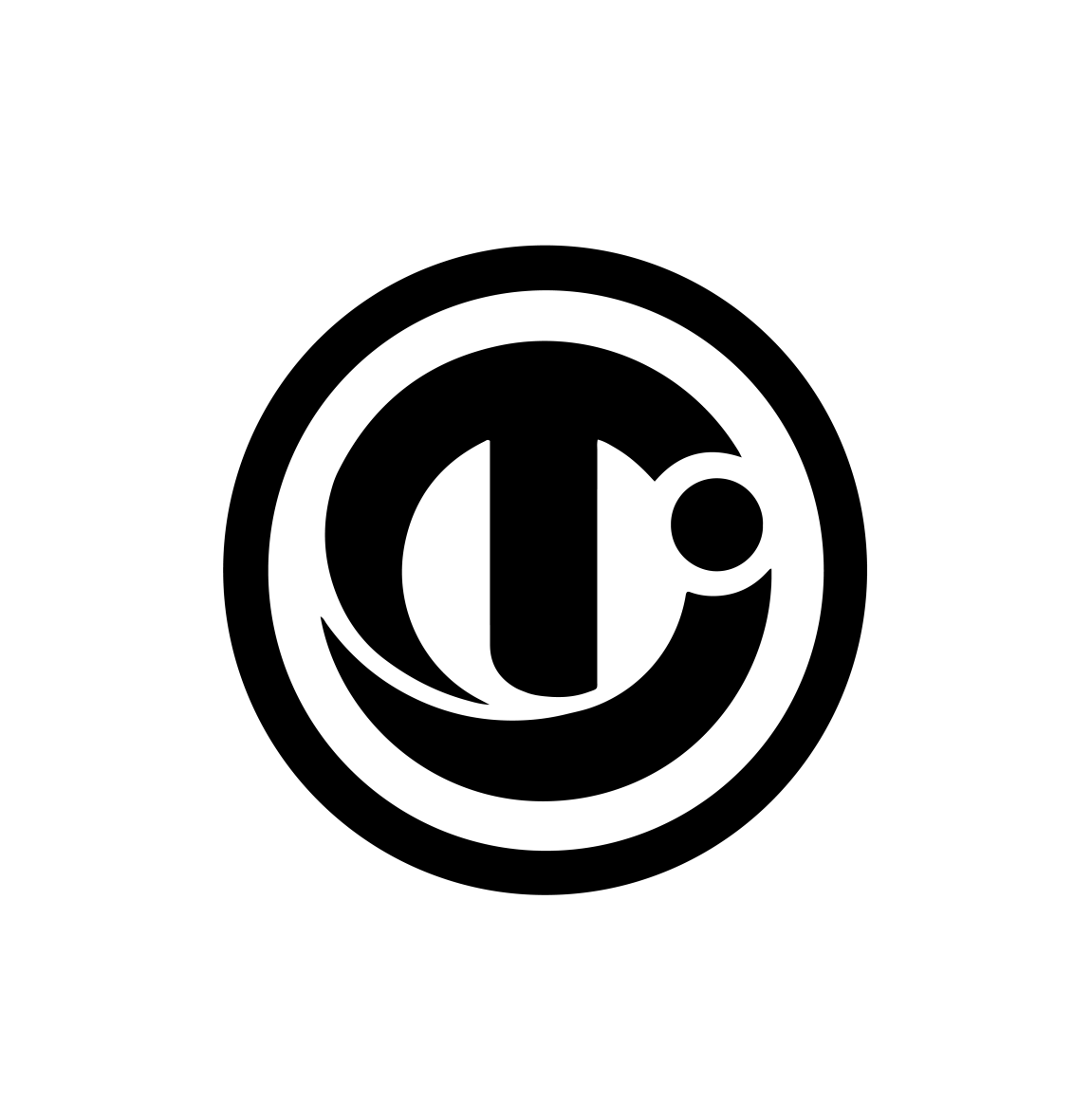
**LAPORAN PROJECT**

**PEMROGRAMAN BERBASIS OBJEK**

Penerapan Inheritance, Abstract Class, dan Interface

dalam OOP



**disusun oleh:**

Kelompok 11 : 17. Nauval Hilmi Hakim (2401082011)

1. Dimas Fachri Husaini (2401083004)

Kelas : Teknik Komputer 1 B

Dosen : Indri Rahmayuni, S.T., M.T

**JURUSAN TEKNOLOGI INFORMASI**

**PROGRAM STUDI D3 TEKNIK KOMPUTER**

**POLITEKNIK NEGERI PADANG**

2025

**DAFTAR ISI**

[Daftar Isi ii](#_Toc5034)

[Tugas 1: Program Dinamis Menghitung Luas & Keliling 1](#_Toc24125)

[A. Deskripsi Tugas 1](#_Toc27677)

[B. Diagram Class 1](#_Toc5601)

[C. Penjelasan Cara Kerja 2](#_Toc2520)

[1. Tampilan Menu, 2](#_Toc22489)

[2. Pengambilan Input, 3](#_Toc14105)

[3. Pembuatan Objek Dan Proses, 4](#_Toc10438)

[4. Pengulangan, 5](#_Toc9191)

[D. Kelebihannya 5](#_Toc1718)

[Tugas 2: Analisis Error Dan Implementasi Abstract Class 6](#_Toc21407)

[A. Deskripsi Tugas 6](#_Toc30487)

[B. Identifikasi Dan Penjelasan Error 6](#_Toc28552)

[C. Perbaikan 7](#_Toc6484)

[D. Hasil Output Perbaikan 9](#_Toc28549)

[Tugas 3: Penerapan Interface Pada Bangun Datar 10](#_Toc2917)

[A. Deskripsi Tugas 10](#_Toc20702)

[B. Diagram Class 10](#_Toc6102)

[C. Penjelasan Cara Kerja Program 11](#_Toc5295)

[1. Implementasi Interface 11](#_Toc8469)

[2. Penggunaan @Override 11](#_Toc19077)

[3. Cara Kerja Main Class 12](#_Toc7164)

[D. Perbandingan Abstract Class Dan Interface 13](#_Toc20784)

[Pembagian Tugas 15](#_Toc15355)

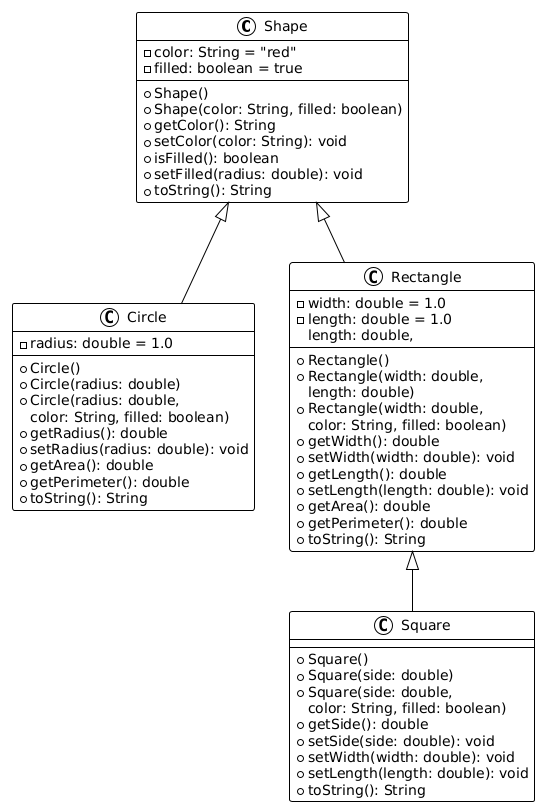
**TUGAS 1**

**Program Dinamis Menghitung Luas & Keliling**

1. **DESKRIPSI TUGAS**

Membuat program Java dengan main class dinamis yang mampu menerima input dari pengguna, menghitung luas dan keliling berbagai bangun datar (lingkaran, persegi, dan persegi panjang), serta menampilkan hasil perhitungan dengan output rapi.

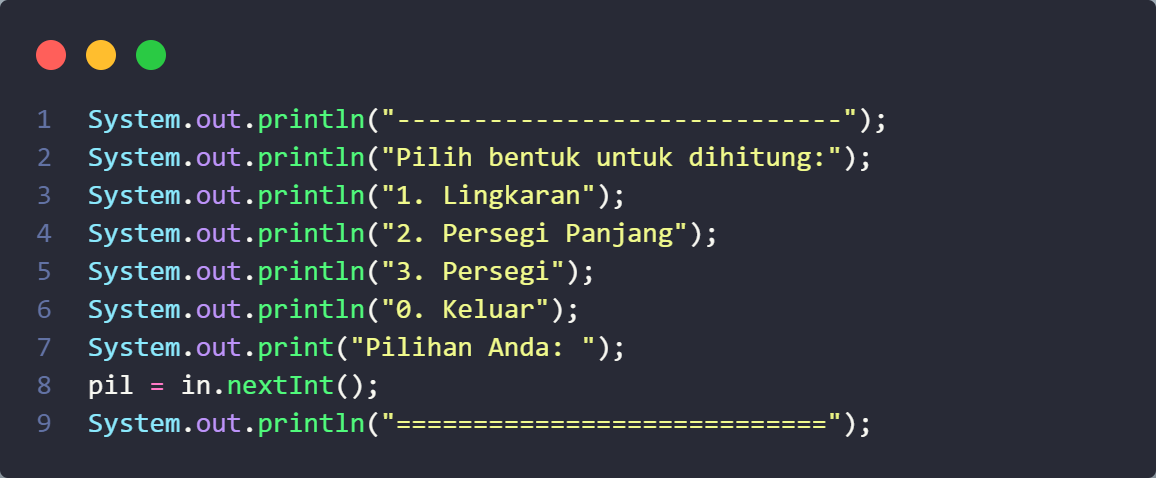
1. **DIAGRAM CLASS**



1. **PENJELASAN CARA KERJA**
2. Tampilan menu,

- Program meminta memilih Lingkaran, Persegi Panjang atau Persegi.

- User memilih bentuk (1–3) atau keluar (0).



1. Pengambilan input,

- Setelah user memilih bentuk:

- Untuk Lingkaran , user diminta nilai jari-jari.

- Untuk Persegi Panjang, user diminta panjang dan lebar.

- Untuk Persegi, user diminta panjang sisi.

- Dan untuk semua bentuk ditanyakan warna dan terisi atau tidaknya.



1. Pembuatan Objek dan Proses,

- Program membuat objek dari kelas sesuai bentuk yang dipilih.

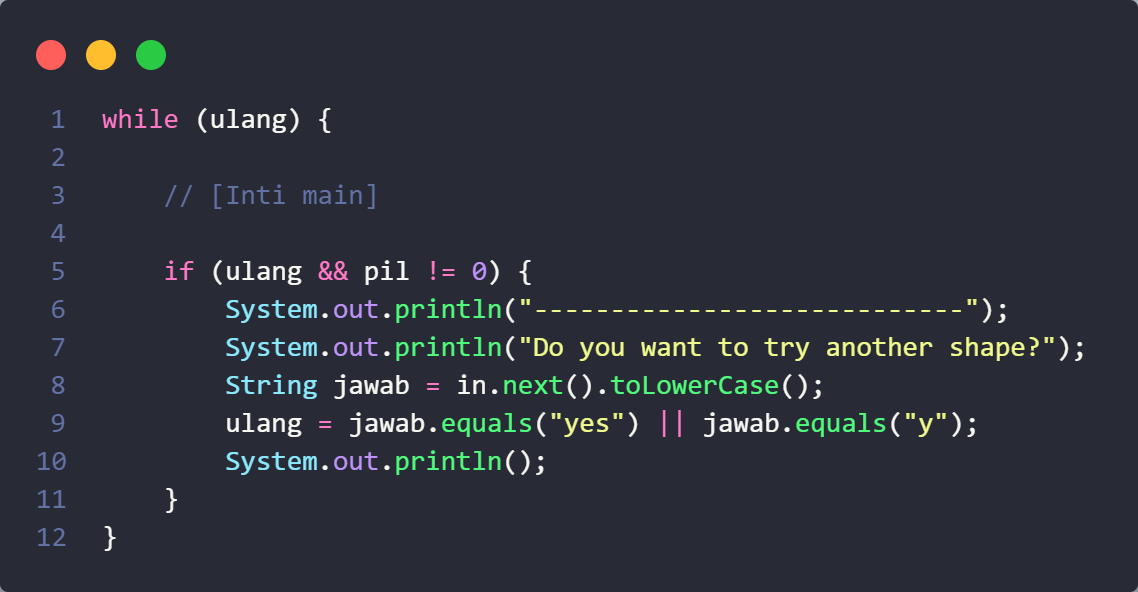
- Objek menghitung luas dan keliling melalui method yang diwariskan.

- Hasil ditampung dan ditampilkan oleh method toString di tiap - tipa kelas.



1. Pengulangan,

- Setelah hasil ditampilkan, akan muncul menu untuk user mengulang atau tidak (opsi 0).



1. **KELEBIHANNYA**
2. Dinamis & User-Friendly

- Pemilihan bentuk dan input nilai disesuaikan dari input langsung user, sehingga tidak statis dan terasa lebih nyata dan fleksibel.

1. Menggunakan Konsep Inheritance

- Kelas-kelas bentuk kemungkinan diturunkan dari kelas dasar (misalnya BangunDatar) sehingga menunjukkan pewarisan (inheritance) yang baik dan mempercepat pengembangan.

1. Terstruktur & Modular

- Masing-masing bentuk geometri dibuat dalam kelasnya sendiri.

- Tugas perhitungan diserahkan pada kelas masing-masing, menjadikan kode lebih bersih dan mudah dikelola.

**TUGAS 2**

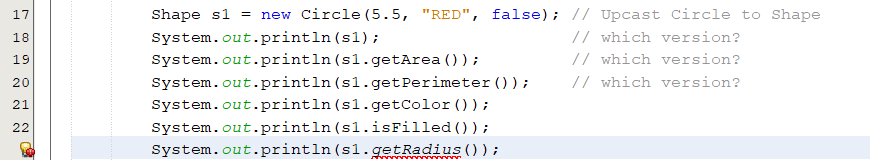
**Analisis Error dan Implementasi Abstract Class**

1. **DESKRIPSI TUGAS**

Di tugas ini, kita melanjutkan dari tugas sebelumnya tapi dengan tambahan konsep abstract class. Kelas Shape diubah jadi abstract, lalu ditambah method abstract getArea() dan getPerimeter() yang harus di-*override* di kelas turunannya.

Kita juga diminta menganalisis kode main yang sudah disiapkan: mencari error-nya, menjelaskan kenapa bisa error, dan memperbaikinya. Setelah itu, kita buat versi main class yang sudah diperbaiki agar bisa jalan lancar dan hasil output-nya benar.

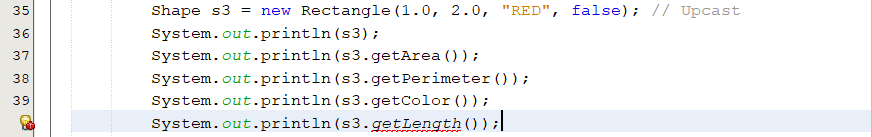
1. **IDENTIFIKASI DAN PENJELASAN ERROR**
2. Bagian Error: Penjelasan = Kompiler hanya melihat tipe referensi (Shape) dan tidak mengetahui bahwa objek di dalamnya adalah Circle saat kompilasi.



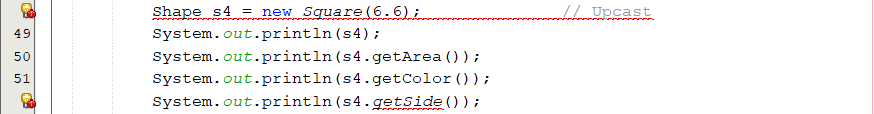
1. Bagian Error: Penjelasan = Kelas Shape merupakan abstract class kita tidak dapat membuat instance atau objek langsung dari kelas abstrak.



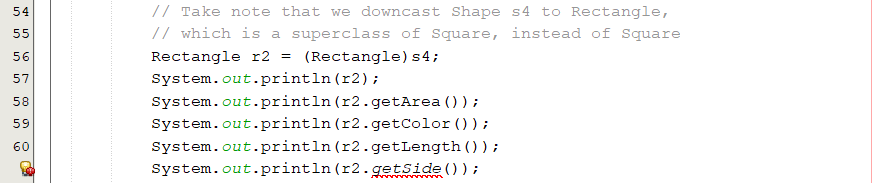
1. Bagian Error: Penjelasan = Sama seperti Error 1. Tipe referensi s3 adalah Shape, sedangkan Shape tidak memiliki method getLength().



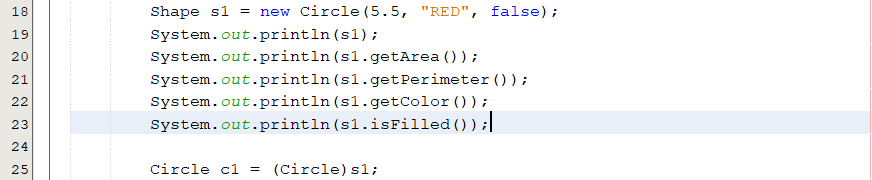
1. Bagian Error: Penjelasan = Apabila konstruktor Square adalah new Square(double side, String color, boolean filled) maka baris ini akan error karena kurang 2 argumen. Dan tipe referensi s4 adalah Shape, dan Shape tidak memiliki metode getSide().



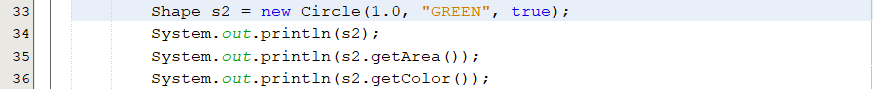
1. Bagian Error: Penjelasan = Kelas Rectangle tidak memiliki metode getSide().



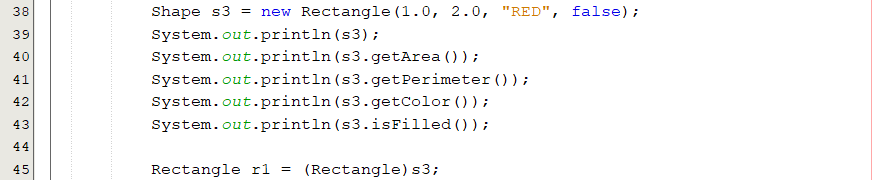
1. **PERBAIKAN**
2. Perbaikan: Untuk mengakses getRadius(), kita perlu melakukan downcasting s1 ke Circle terlebih dahulu, seperti yang dilakukan pada baris berikutnya Circle c1 = (Circle)s1;.



1. Perbaikan: Anda harus membuat instance dari salah satu subclass konkret dari Shape, seperti Circle, Rectangle, atau Square.



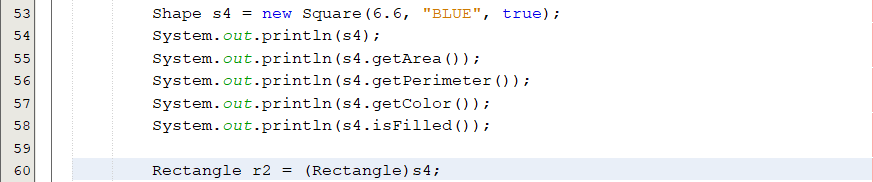
1. Perbaikan: kita perlu meakukan downcasting s3 ke Rectangle untuk dapat mengakses method getLength().



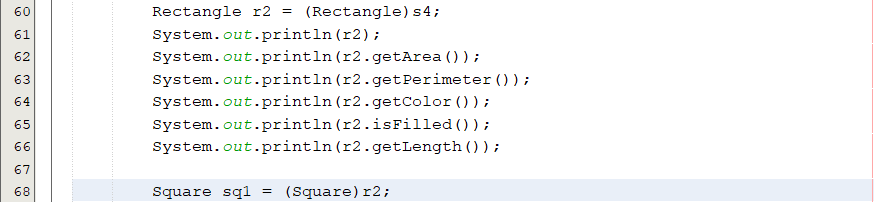
1. Perbaikan: Maka seharusnya perbaikannya seperti:

Shape s4 = new Square(6.6, "BLUE", true);.

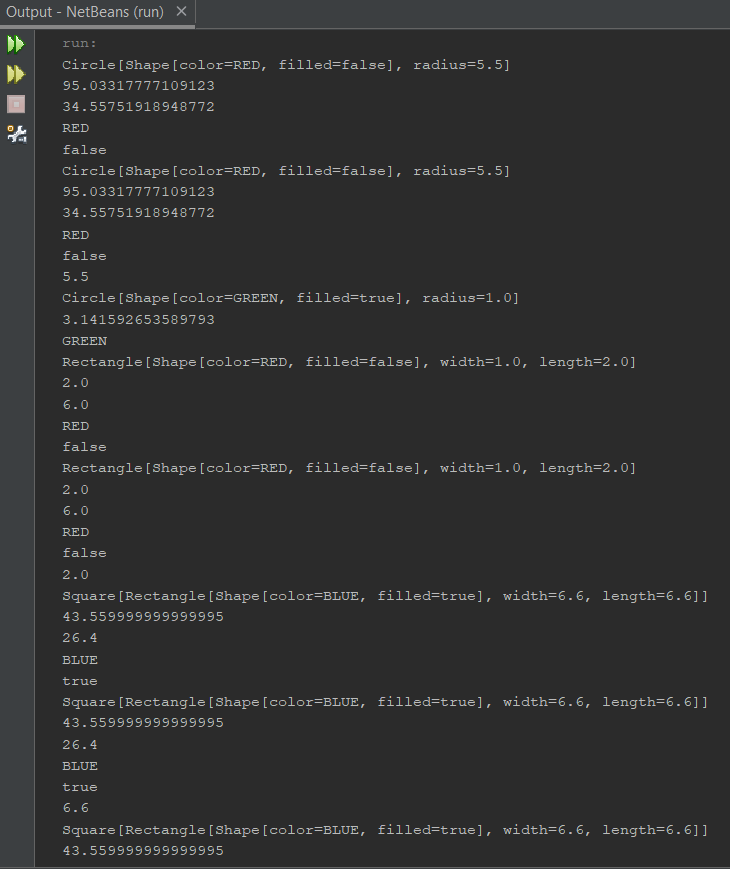
Dan melaakukan Downcasting s4 ke Square (atau Rectangle terlebih dahulu, lalu Squere) untuk mengakses getSide().



1. Perbaikan: malakukan downcasting r2 ke Square untuk mengakses getSide().



1. **HASIL OUTPUT PERBAIKAN**



**TUGAS 3**

**Penerapan Interface pada Bangun Datar**

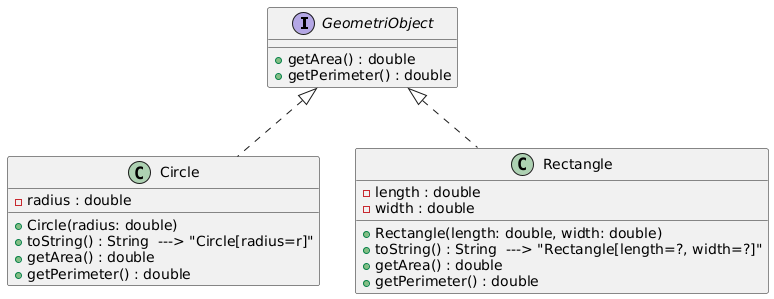
1. **DESKRIPSI TUGAS**

Di tugas ini, kita diminta buat sesuatu yang mirip kayak tugas sebelumnya, tapi kali ini pakai interface, bukan abstract class. Kita harus bikin interface bernama GeometriObject yang isinya dua method: *getArea()* dan *getPerimeter()*.

Nah, interface itu nanti di-implement sama dua kelas: Circle dan Rectangle. Jadi dua kelas itu wajib punya versi mereka sendiri dari *getArea()* dan *getPerimeter()*, dan semuanya harus dikasih anotasi *@Override* biar jelas.

Terakhir, kita juga harus bikin main class buat nguji apakah semuanya jalan atau enggak. Intinya, kita bikin dan ngetes lagi kayak sebelumnya, tapi pakai pendekatan interface. Di akhir tugas juga disuruh ngebandingin, sih — gimana bedanya waktu kita pakai abstract class di tugas 2, dan interface di tugas 3 ini.

1. **DIAGRAM CLASS**

****

1. **PENJELASAN CARA KERJA PROGRAM**
2. IMPLEMENTASI INTERFACE

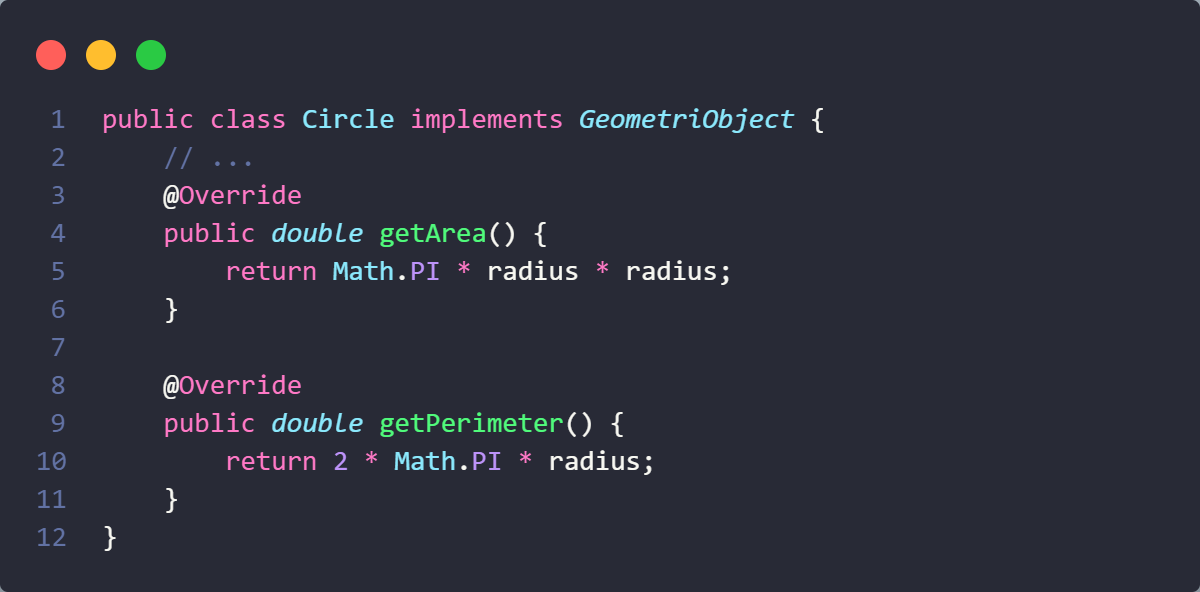
Interface GeometriObject berfungsi sebagai aturan dasar yang harus diikuti oleh setiap kelas yang menggunakannya. Dalam hal ini, interface tersebut memiliki dua metode:

- getArea() : digunakan untuk menghitung luas suatu objek geometri.

- getPerimeter() : digunakan untuk menghitung keliling suatu objek geometri.

Kelas Circle dan Rectangle mengimplementasikan interface ini. Artinya, mereka harus menyediakan sendiri cara kerja dari kedua metode tersebut sesuai bentuk masing - masing.

Contoh nya seperti ini:



1. PENGGUNAAN @OVERRIDE

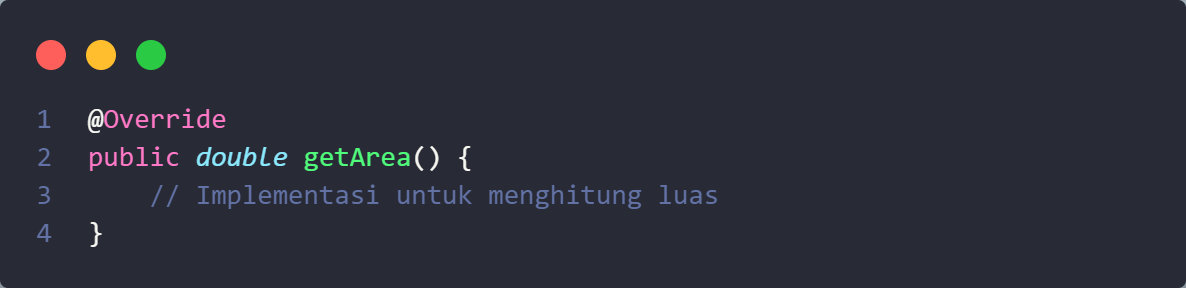
Anotasi @Override digunakan untuk menandai bahwa sebuah metode adalah versi yang dimodifikasi (override) dari metode yang sudah ada di interface atau kelas induk.

Manfaat dari penggunaan anotasi ini antara lain:

- Membantu pengecekan: Jika ternyata metode yang ditulis tidak cocok atau salah eja, program akan menampilkan kesalahan saat dikompilasi.

- Meningkatkan keterbacaan kode: Dengan anotasi ini, pembaca kode akan lebih mudah memahami bahwa metode tersebut merupakan turunan atau pengganti dari metode yang sudah ada sebelumnya.

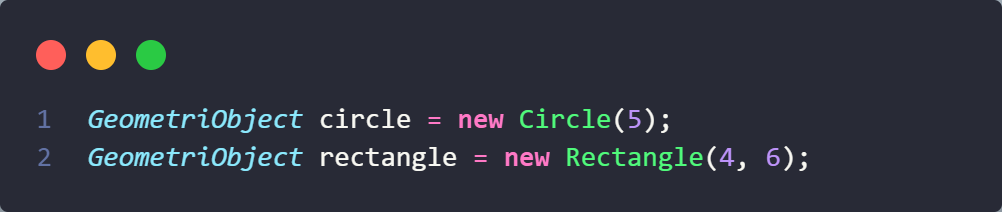
Contoh penggunaan @Override dalam kelas Circle dan Rectangle adalah saat mengimplementasikan getArea(), getPerimeter(), dan toString().



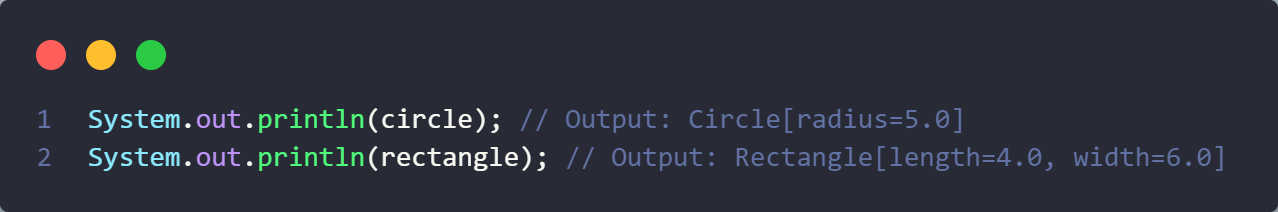
1. CARA KERJA MAIN CLASS

Kelas main berfungsi sebagai titik awal program dijalankan. Di dalam metode main, ada beberapa langkah utama yang dilakukan:

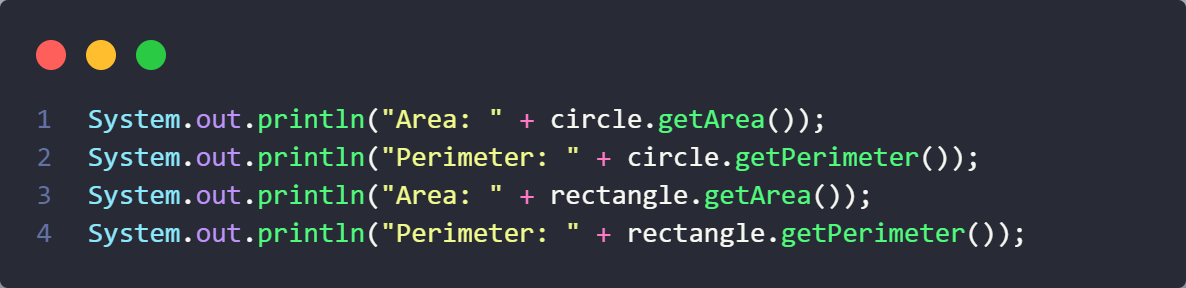
1. *Membuat Objek:* Objek dari kelas Circle dan Rectangle dibuat dengan memberikan nilai-nilai yang dibutuhkan, seperti jari-jari untuk lingkaran atau panjang dan lebar untuk persegi panjang.



1. *Menampilkan Informasi Objek:* Metode toString() yang sudah dioverride digunakan untuk menampilkan informasi tentang objek tersebut dalam bentuk teks.



1. *Menghitung luas dan keliling:* Metode getArea() dan getPerimeter() dipanggil untuk menghitung nilai luas dan keliling, kemudian hasilnya ditampilkan.



1. **PERBANDINGAN ABSTRACT CLASS DAN INTERFACE**

Apa Itu Abstract Class dan Interface?

- Abstract class itu kayak cetakan dasar sebuah kelas yang nggak bisa langsung dipakai, tapi bisa punya isi (method) dan juga tempat kosong (abstract method) yang harus diisi oleh anak-anaknya (kelas turunannya).

- Interface itu lebih kayak daftar aturan atau kontrak: "kalau kamu mau implement saya, kamu wajib punya method ini dan itu". Isinya cuma method kosong, yang baru nanti diisi di kelas yang pakai.

Soal Isi Method

- Di abstract class, kita boleh punya campuran: ada method yang udah jadi, dan ada juga yang masih abstract (belum diisi).

- Di interface, dulu (sebelum Java 8) semua method harus kosong. Tapi sejak Java 8, kita boleh punya method default juga, meskipun itu nggak umum banget.

Bisa Turun dari Berapa?

- Abstract class cuma bisa diturunin satu. Jadi satu kelas cuma bisa extend satu abstract class.

- Tapi kalau interface, bebas! Satu kelas bisa implement banyak interface sekaligus.

Kapan Dipakai?

- Pakai abstract class kalau kamu mau bikin struktur pewarisan yang kuat — misalnya semua bentuk geometri itu “adalah” Shape.

- Pakai interface kalau kamu cuma mau bilang: “kelas ini harus bisa melakukan hal tertentu”, tanpa peduli dia anak dari siapa.

Akses dan Aturan

- Di abstract class, kamu bisa atur akses method dan atributnya pakai public, protected, atau private.

- Di interface, semua method itu otomatis public dan abstract. Dan kalau kamu bikin variabel, dia otomatis public static final alias konstanta.

**PEMBAGIAN TUGAS**

Dalam pengerjaan proyek ini, kami membagi tugas sebagai berikut:

- Nauval Hilmi Hakim bertanggung jawab mengerjakan Tugas 1 dan Tugas 3, termasuk penulisan laporannya.

- Dimas Fachri Husaini mengerjakan Tugas 2 beserta laporannya.

Setelah semua bagian selesai, penyusunan laporan akhir dilakukan bersama-sama oleh kami berdua, agar hasilnya utuh dan rapi sebagai satu kesatuan.