**P**emrograman **Berorientasi** **O**bjek

*Tugas Kelas Abstrak & Interface*

A picture containing text, clipart

Description automatically generated

Dipersiapkan Oleh :

< Chandra Harkat Raharja> - < 233040089>

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

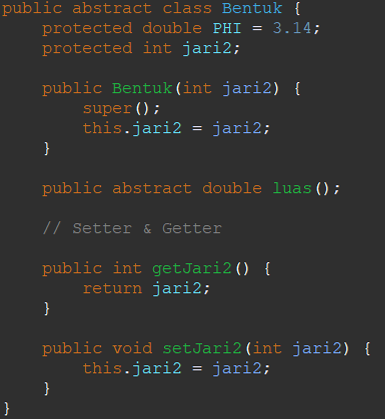
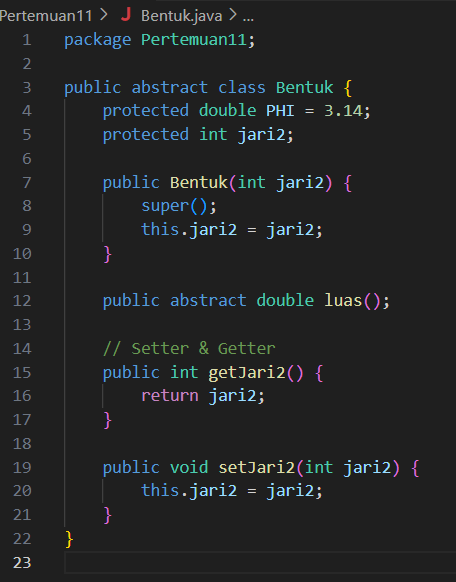
**UNIVERSITAS PASUNDAN**

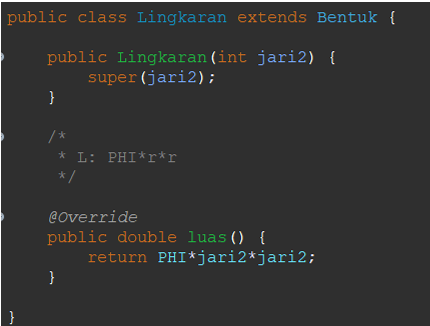
**BANDUNG**

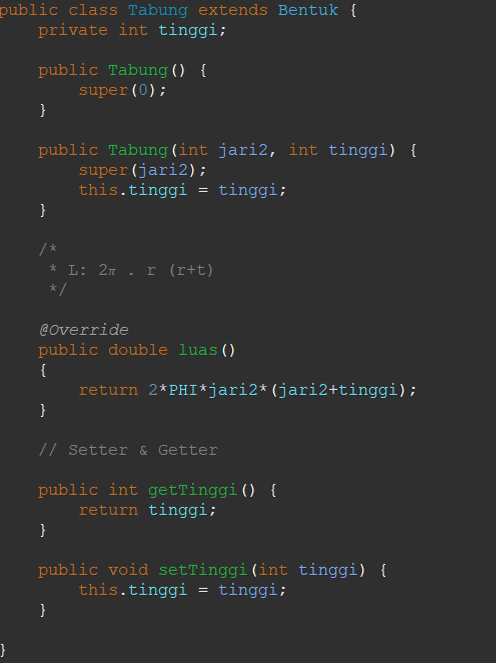
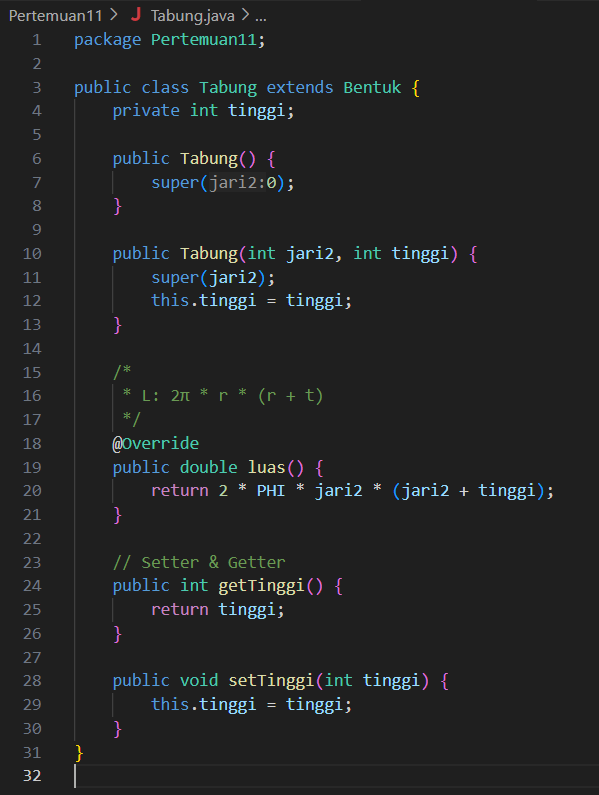
**2024**

Link Github: [Pemrograman-Berbasis-Objek\_233040089/Pertemuan11 at main · ComradeChandra/Pemrograman-Berbasis-Objek\_233040089](https://github.com/ComradeChandra/Pemrograman-Berbasis-Objek_233040089/tree/main/Pertemuan11)  
-LATIHAN 1

Latihan ke-1 ini memberikan contoh program kelas abstrak. Kelas Bentuk adalah kelas abstrak yang salah satu methodnya abstrak. Method abstrak tersebut akan diimplementastikan oleh kelas Lingkaran dan kelas Tabung.

- Buatlah Kelas Bentuk Seperti di Bawah ini: -Jawab:

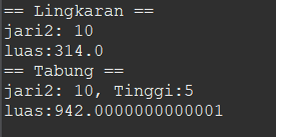
  
- Buatlah Kelas Lingkaran Seperti Di Bawah Ini: -Jawab:

-Buatlah Kelas Tabung Seperti Di bawah Ini: -Jawab:

PERTANYAAN:  
1. Sebutkan method abstrak yang terdapat di kelas abstrak Bentuk !

2. Jelaskan apa perbedaannya method luas di kelas Bentuk dan di kelas Lingkaran dan kelas Tabung!

3. Jelaskan bagaimana jika method luas di kelas Tabung dihilangkan atau tidak mengimplementasikannya!

4. Buatlah kelas main sehingga outputnya seperti dibawah ini:   


5. Buatlah diagram kelas dari contoh program kelas abstrak!

JAWABAN:  
1. Di dalam kelas Bentuk, terdapat satu method abstrak yaitu:

public abstract double luas();

Method ini hanya dideklarasikan tanpa isi atau implementasi, dan harus diimplementasikan oleh semua kelas yang mewarisi Bentuk.

2. Di kelas Bentuk, method luas() hanya didefinisikan sebagai method abstrak. Artinya, method ini **tidak memiliki isi** dan hanya berfungsi sebagai "kontrak" bagi kelas turunan.

2.a. Di kelas Lingkaran, method luas() di-*override* dan berisi rumus luas lingkaran, yaitu:  
PHI × jari-jari × jari-jari

2.b. Di kelas Tabung, method luas() juga di-*override* dan berisi rumus luas permukaan tabung, yaitu:  
2 × PHI × jari-jari × (jari-jari + tinggi)  
  
Jadi perbedaannya adalah:

* Di Bentuk, method ini hanya berupa deklarasi.
* Di Lingkaran dan Tabung, method ini memiliki isi yang sesuai dengan bentuknya masing-masing.

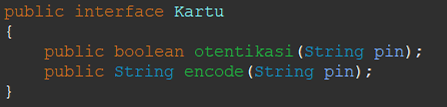
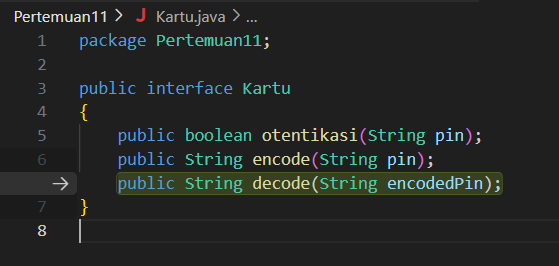
3. Jika method luas() tidak diimplementasikan di kelas Tabung, maka akan muncul error saat kompilasi. Hal ini karena Tabung adalah turunan dari Bentuk, dan kelas Bentuk memiliki method abstrak luas().

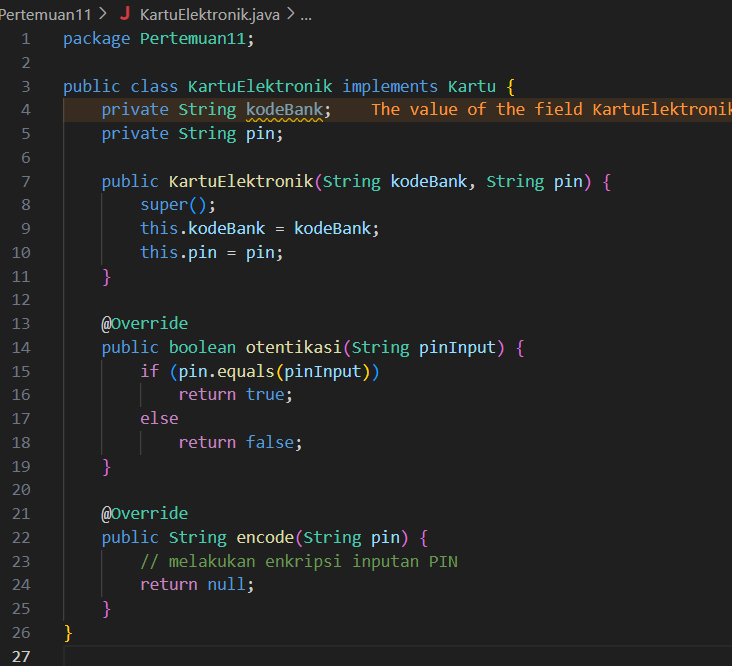
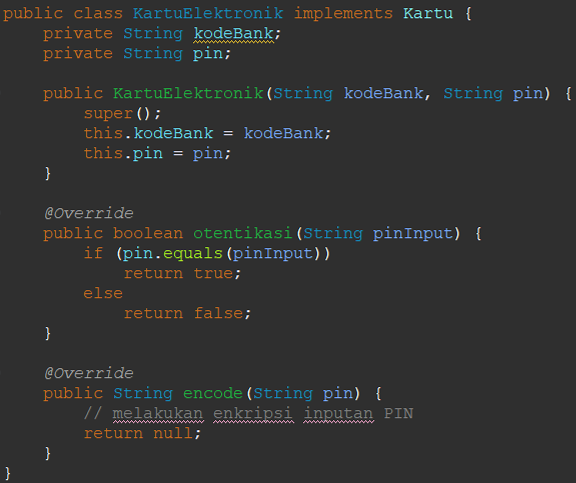
4. Jika method luas() tidak diimplementasikan di kelas Tabung, maka akan muncul **error saat kompilasi**. Hal ini karena Tabung adalah turunan dari Bentuk, dan kelas Bentuk memiliki method abstrak luas().

Dalam Java, setiap kelas turunan dari kelas abstrak **wajib mengimplementasikan semua method abstrak**. Jika tidak, maka kelas turunan tersebut juga harus ditandai sebagai abstract. Karena Tabung tidak ditandai sebagai abstract, maka method luas() wajib ditulis.

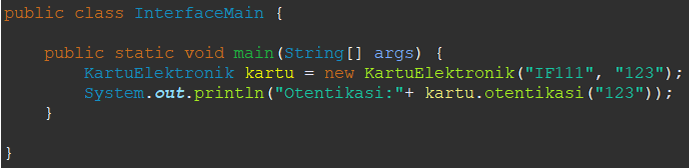
**5.**

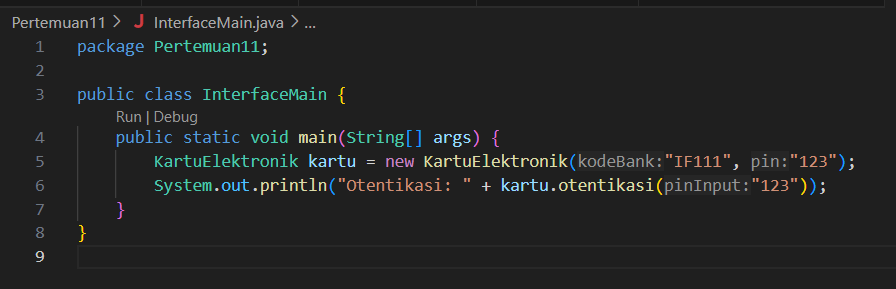
**LATIHAN-2**Latihan ke-2 ini memberikan contoh program interface. Interface Kartu mempunyai semua method abstrak yaitu otentikasi dan encode. Interface Kartu harus implementasikan oleh kelas konkret yaitu kelas KartuElektronik. Kartu elektronik menggunakan interface Kartu menggunakan keyword implements.

- Buatlah interface Kartu seperti dibawah ini  
  
Jawab:  


-Buatlah kelas KartuElektronik seperti dibawah ini -Jawab  


-Buatlah kelas InterfaceMain seperti dibawah ini



-Jawab:  


**-Tugas**   
Buatlah contoh kasus lainnya program kelas abstrak dan interface seperti latihan yang sudah anda kerjakan dalam Bahasa pemrograman Java. Laporan yang dikumpulkan adalah rancangan kelas diagram dan sintaks kode programnya serta outputnya dari kode program tersebut.

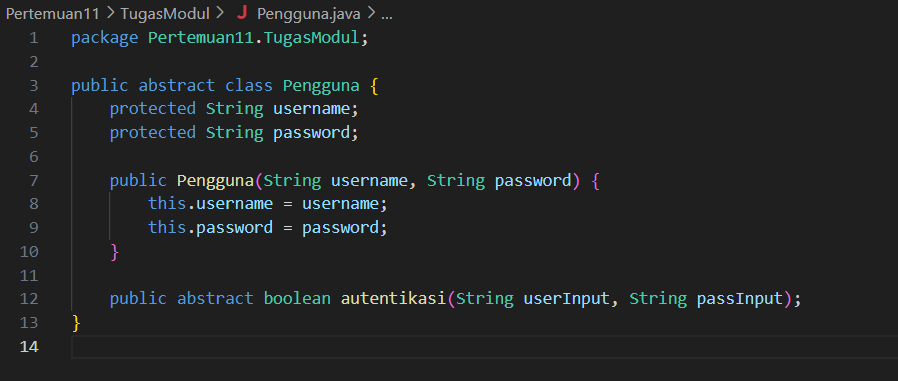
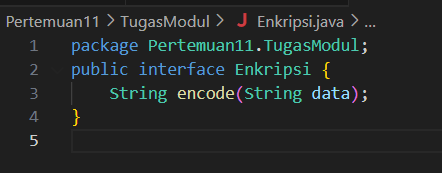
Jawab:  
Judul kasus: Sistem Autentikasi Pengguna di Aplikasi Digital

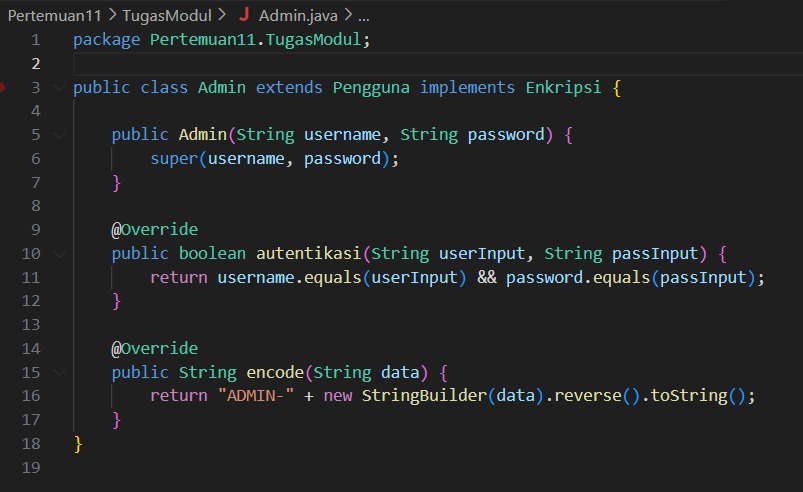
Penjelasan:  
Pada program ini, dibuat sebuah simulasi sederhana mengenai sistem otentikasi pengguna yang terdiri dari dua jenis akun: Admin dan User Biasa. Setiap pengguna memiliki username dan password, serta harus bisa melakukan proses otentikasi dan enkripsi data.

Untuk menerapkan konsep abstraksi, digunakan sebuah kelas abstrak bernama Pengguna yang memuat struktur dasar pengguna dan method abstrak autentikasi(). Lalu, digunakan juga sebuah interface bernama Enkripsi yang berisi deklarasi method encode() untuk proses enkripsi data.

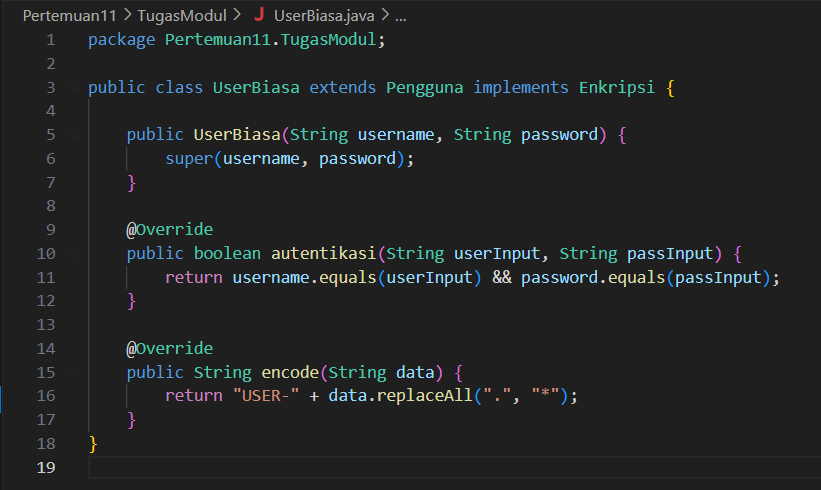
Dua kelas konkret yaitu Admin dan UserBiasa kemudian mewarisi dari Pengguna dan mengimplementasikan interface Enkripsi.

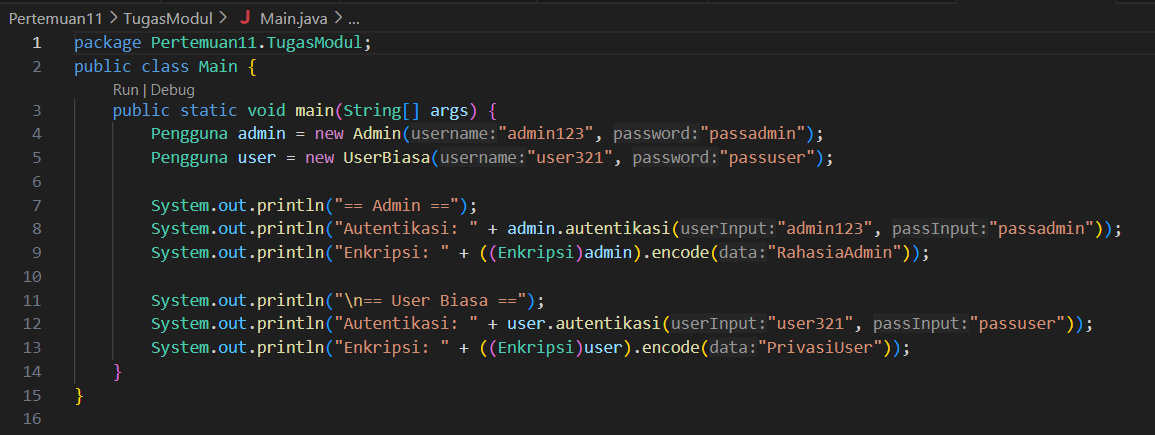
**Penjelasan Struktur Program**

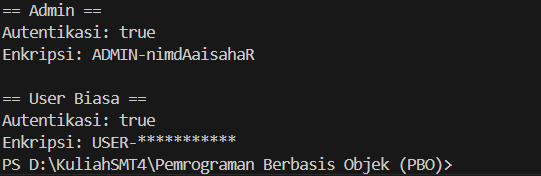
1. **Kelas abstrak Pengguna**  
   Merupakan kelas dasar yang menyimpan atribut username dan password. Di dalamnya terdapat method abstrak autentikasi() yang harus diimplementasikan oleh kelas turunannya.
2. **Interface Enkripsi**  
   Interface ini mendefinisikan method encode() untuk melakukan enkripsi data. Tujuannya agar setiap kelas yang membutuhkan fitur enkripsi dapat mengimplementasikan interface ini sesuai kebutuhan masing-masing.  
     
   
3. **Kelas Admin**  
   Kelas ini mewarisi dari Pengguna dan mengimplementasikan Enkripsi. Metode autentikasi() akan memverifikasi username dan password, sedangkan encode() akan membalik string (simulasi enkripsi).



1. **Kelas UserBiasa**  
   Mirip seperti Admin, namun metode encode() di kelas ini akan menyamarkan karakter string dengan tanda \* sebagai bentuk penyembunyian informasi.



1. **Kelas Main**  
   Berfungsi sebagai kelas utama untuk mengeksekusi program. Di sini dibuat objek Admin dan UserBiasa, lalu ditampilkan hasil dari proses otentikasi dan enkripsi datanya.  
   

**Output:  
**