**LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK**

**JOBSHEET 11: INTERFACE CLASS**

****

oleh :

Halim Teguh Saputro

2E

2141762122

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

Jl. Soekarno Hatta No .9, Jatimulyo, Kec. Lowokwaru, Kota Malang,

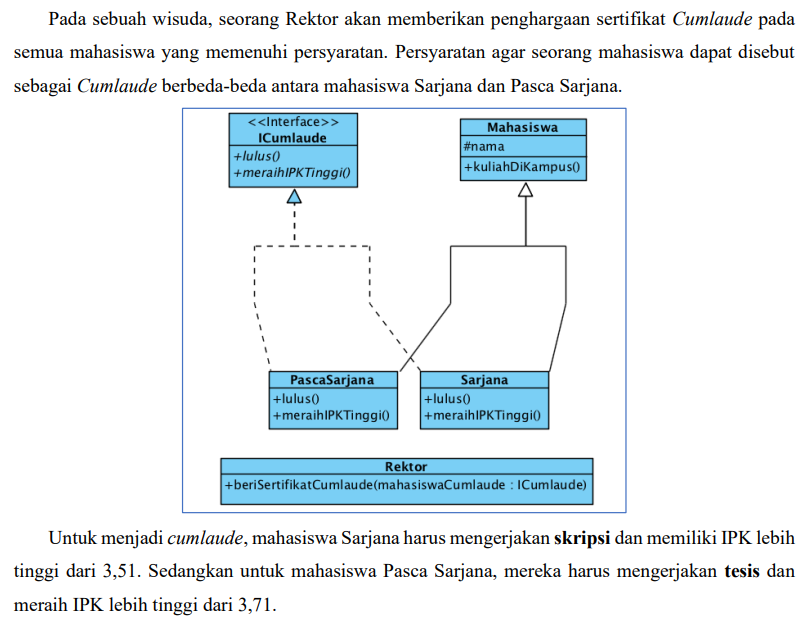
JawaTimur 65141

|  |
| --- |
| KOMPETENSI |

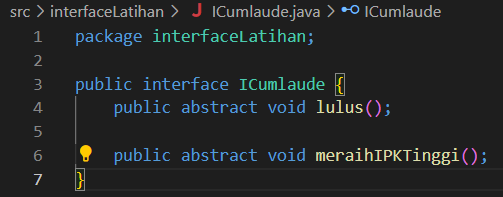
Setelah melakukan percobaan pada modul ini, mahasiswa memahami konsep:

1. Menjelaskan maksud dan tujuan penggunaan Interface
2. Menerapkan Interface di dalam pembuatan program

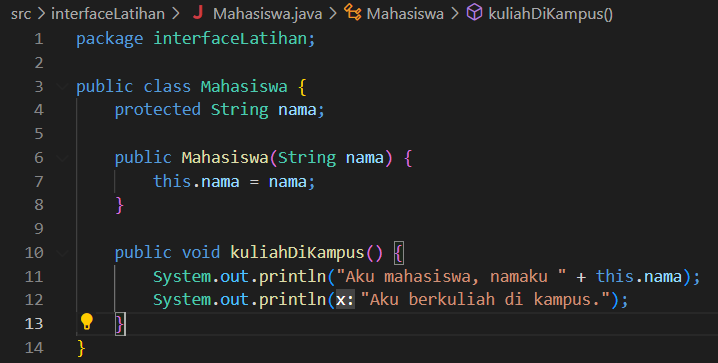
|  |
| --- |
| PRAKTIKUM 1. INTERFACE |



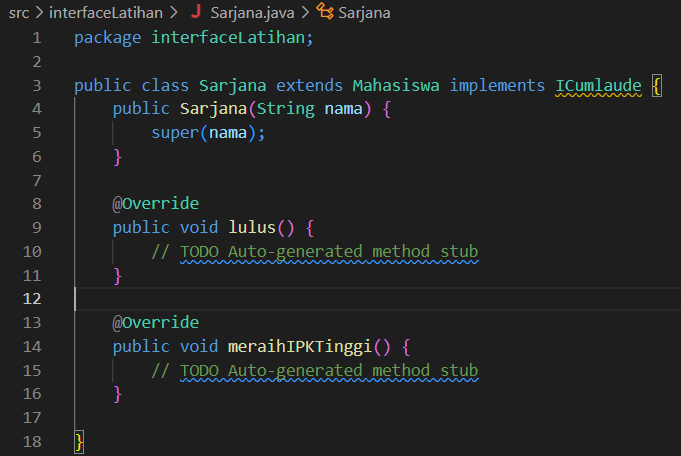
1. Pada project Latihan PBOMinggu10, buat sebuah package baru Bernama interfacelatihan. Kemudian klik Finish.
2. Pada package yang baru dibuat tersebut, tambahkan sabuah interface baru dengan cara klik kanan pada package > New > Java Interface… Beri nama interface baru tersebut dengan nama ICumlaude
3. Pada interface ICumlaude tersebut, tambahkan 2 abstract methods Bernama lulus() dan meraihIPKTinggi().

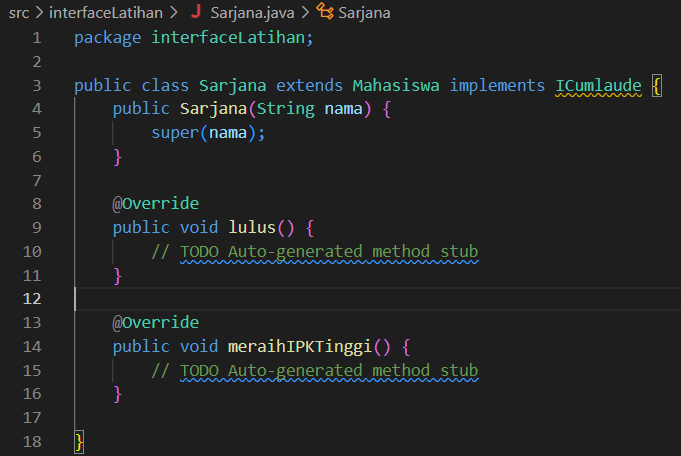


1. Berikutnya, buatlah sebuah class baru Bernama Mahasiswa dengan baris-baris kode seperti dibawah ini.

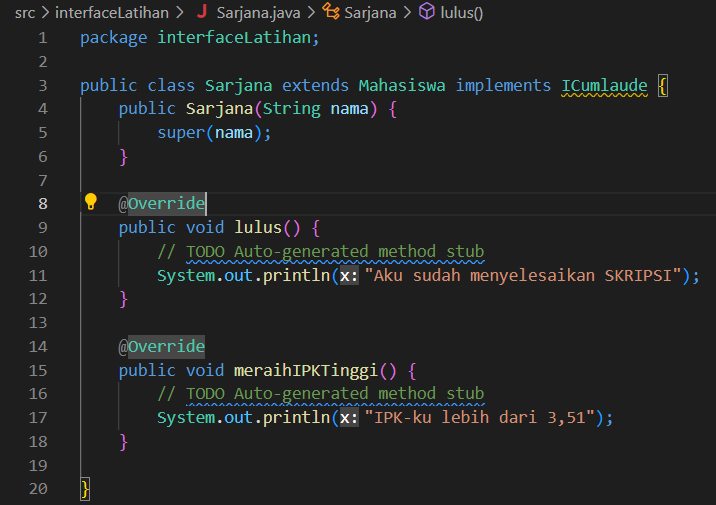


1. Selanjutnya, buatlah class baru Bernama Sarjana yang merupakan turunan dari class Mahasiswa. Class Sarjana tersebut dibuat meng-Implements interface ICumlaude yang sudah dibuat sebelumnya tadi. Ketikkan kode di bawah pada class tersebut. Tips: anda dapat menggunakan fasilitas override otomatis dengan cara yang sama yaitu dengan mengklik ikon lampu peringatan seperti pada percobaan 1.



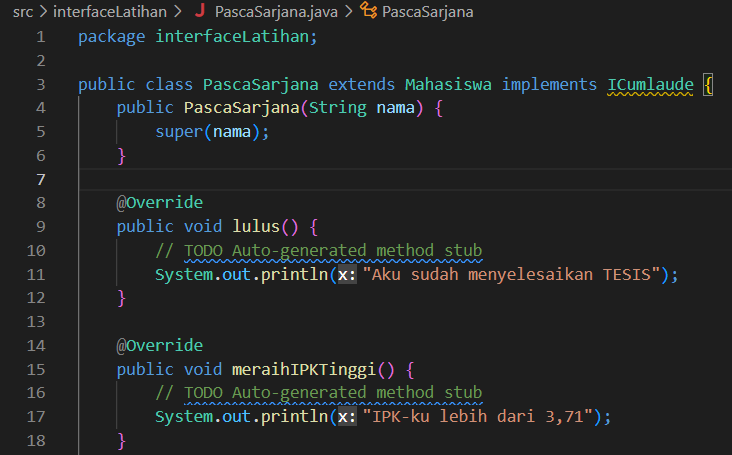


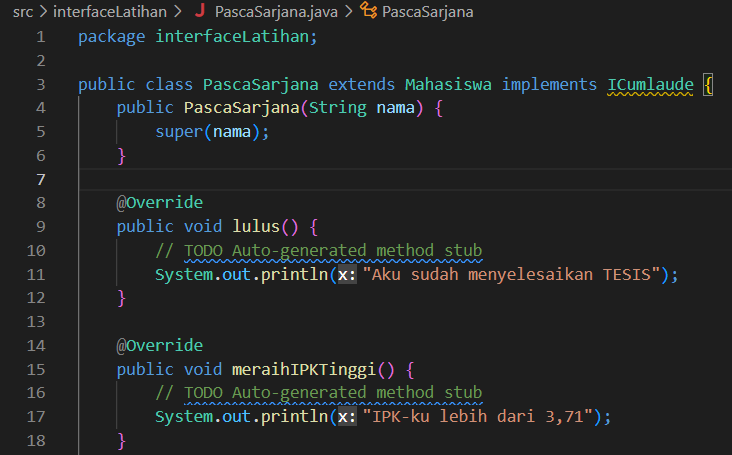
1. Selanjutnya sesuaikan isi dari method lulus() dan meraihIPKTinggi() agar sama dengan baris kode di bawah.



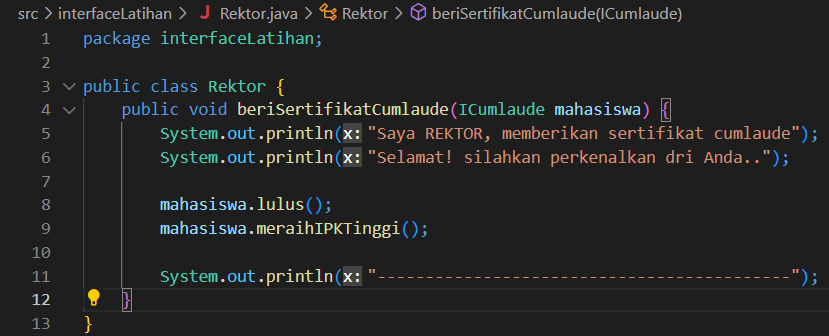
Perhatikan pada baris kode di atas, class Sarjana meng-extend class Mahasiswa, ini berarti, Sarjana adalah Mahasiswa sementara itu agar semua objek dari class Sarjana ini nantinya dapat disebut sebagai Cumlaude makai a harus meng-Implements interface ICumlaude.

1. Kemudian dengan cara yang sama buatlah class baru Bernama PascaSarjana dengan baris kode seperti di bawah ini.





1. Lalu buatlah sebuah class baru Bernama Rektor. Class ini adalah class yang memanfaatkan class-class Mahasiswa yang telah dibuat sebelumnya.

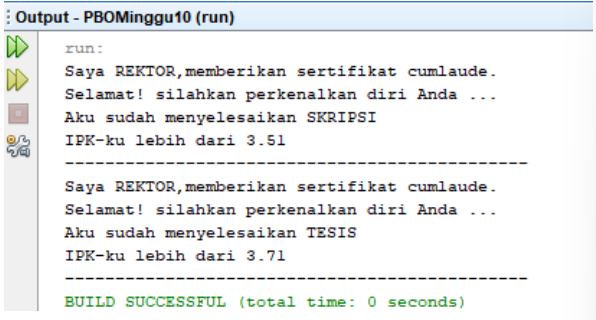


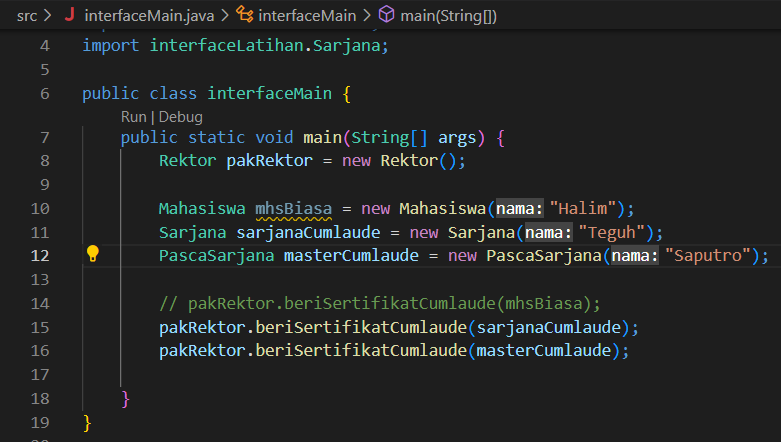
1. Terakhir, buatlah sebuah class baru Bernama interfaceMain yang diletakkan pada package yang sama dengan class-class percobaan 2. Tambahkan baris kode berikut ini:



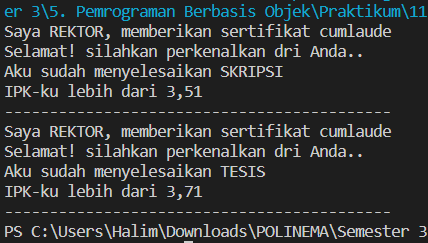


1. Pada baris kode tersebut, apabila anda mengetikkan semua class dengan benar, maka akan terdapat error dan class Program tidak dapat dieksekusi. Perbaikilah kode anda agar program yang anda buat mengeluarkan output seperti berikut ini:





Keterangan: saya mengubah baris ke-14 yaitu sintaks yang memanggil method beriSertifikatCumlaude yang memanggil mhsBiasa menjadi sintaks komentar.



|  |
| --- |
| PERTANYAAN PRAKTIKUM 1 |

1. Pada Langkah ke 9, pada baris program ke 3 terdapat warning pada script tersebut. Jelaskan penyebab terjadinya hal tersebut?

Jawab: terjadi warning karena program interfaceMain tidak menggunakan file ICumlaude. Sehingga seharusnya program tidak memerlukan import dari ICumlaude tersebut.

1. Pada Langkah ke 9, pada baris program ke 3. Apa yang terjadi jika script tersebut dihilangkan? Jelaskan pemahaman anda

Jawab: tidak terjadi masalah atau error, karena file ICumlaude tidak digunakan pada interfaceMain.

1. Mengapa pada langka nomor 9 terjadi error? Jelaskan!

Jawab: menurut saya, terjadi error karena method beriSertifikatCumlaude() memerlukan atau akan menjalankan method lulus() dan meraihIPKTinggi(). Namun pada class Mahasiswa tidak memiliki method tersebut.

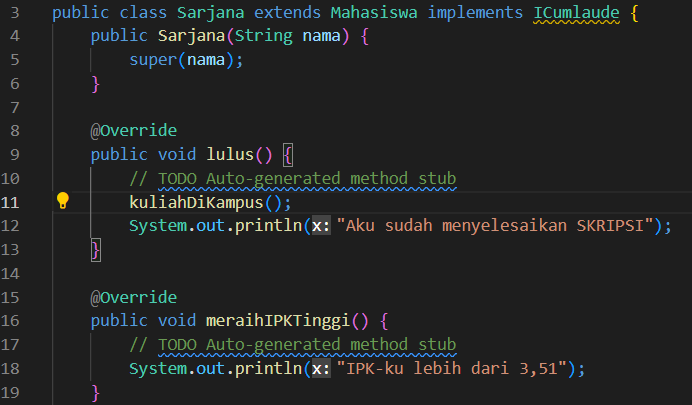
1. Dapatkah method kuliahDiKampus() dipanggil dari objek sarjanaCumlaude di class Program? Mengapa demikan?

Jawab: Bisa, karena sarjanaCumlaude merupakan objek dari class Sarjana dan class Sarjana meng-extend ke class Mahasiswa yang memiliki method kuliahDiKampus().

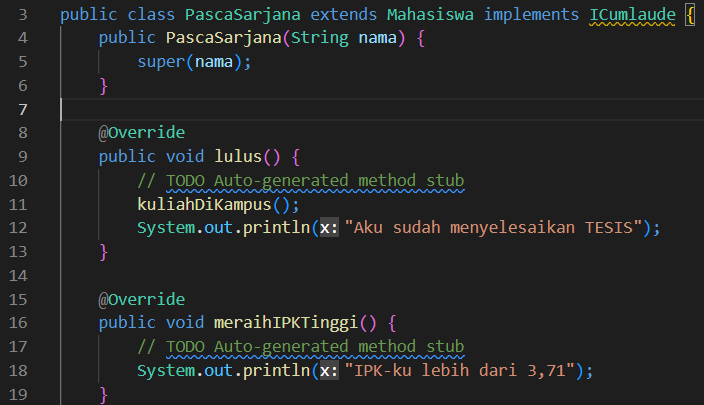
1. Dapatkah method kuliahDiKampus() dipanggil dari parameter mahasiswa di method beriSertifikatCumlaude() pada class Rektor? mengapa demikian?

Jawab: Tidak bisa, karena parameter mahasiswa di method beriSertifikatCumlaude() itu berasal dari interface ICumlaude yang tidak memiliki method kuliahDiKampus(). Method kuliahDiKampus() berada pada class Mahasiswa bukan ICumlaude

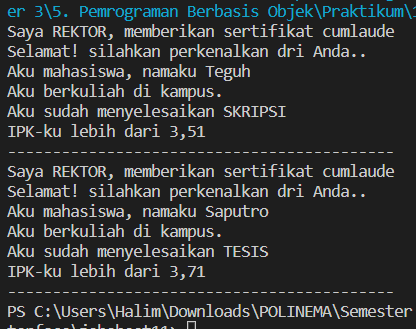
1. Modifikasilah method berisertifikatCumlaude() pada class Rektor agar hasil eksekusi class Program menjadi seperti berikut ini:
2. Sarjana



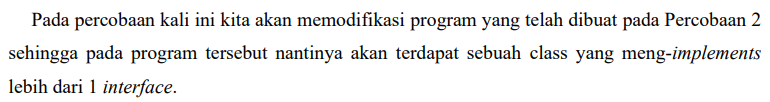
1. PascaSarjana

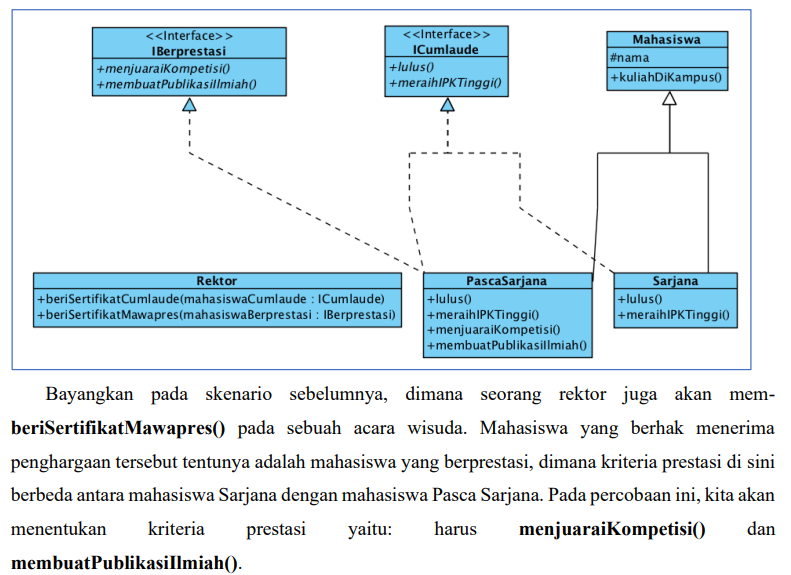


1. Output

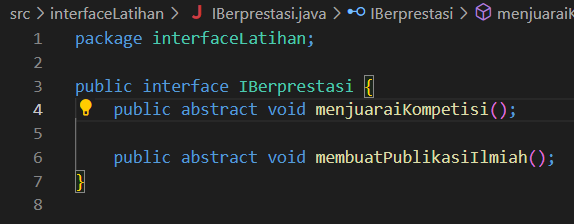


|  |
| --- |
| PRAKTIKUM 2. MUTIPLE INTERFACES IMPLEMENTATION |

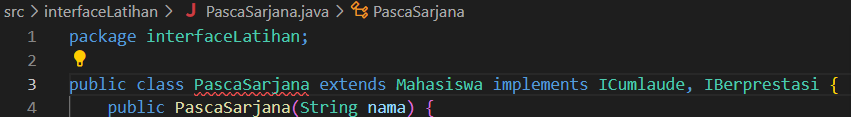




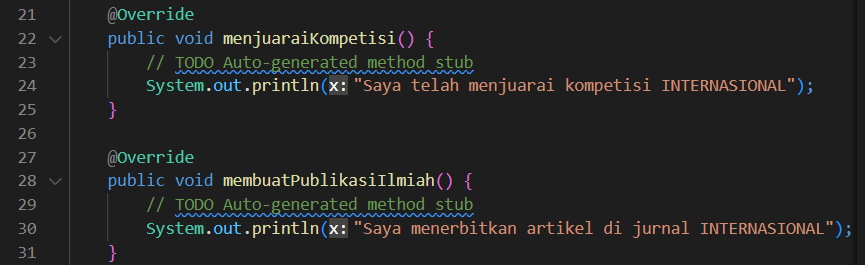
1. Pada package yang sama dengan package pada Percobaan Sebelumnya, tambahkan sebuah interface baru Bernama IBerprestasi. Tambahkan baris kode seperti berikut didalamnya.



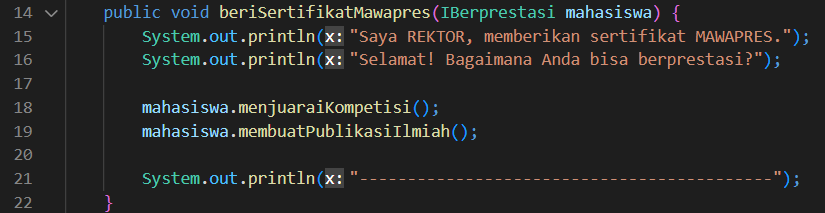
1. Selanjutnya, modifikasilah class PascaSarjana dengan menambahkan interfave baru IBerprestasi dibelakang kata kunci implements. Lalu dengan cara yang sama seperti sebelumnya, kliklah ikon lampu peringatan untuk meng-generate semua method abstract dari interface pada class PascaSarjana.



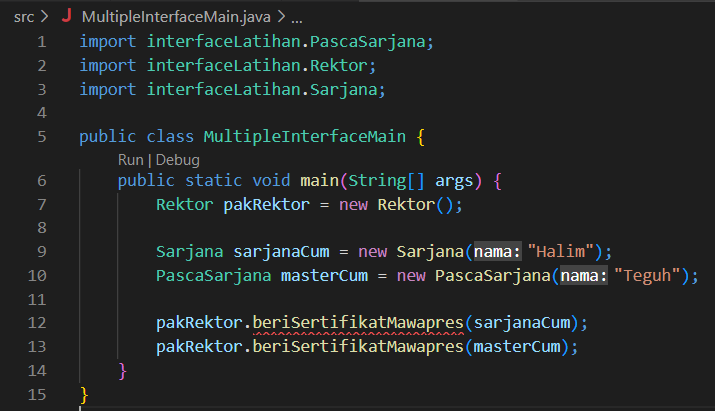
1. Modifikasilah method yang telah di-generate oleh NetBeans menjadi seperti berikut



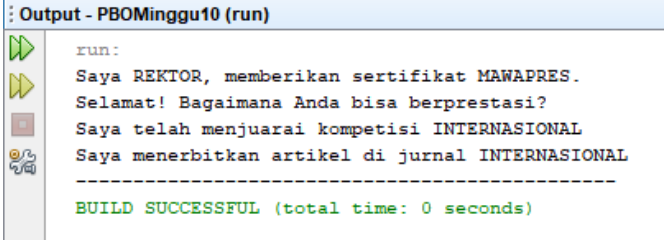
1. Tambahkan method beriSertifikatMawapres() dengan baris kode seperti di bawah, pada class Rektor.

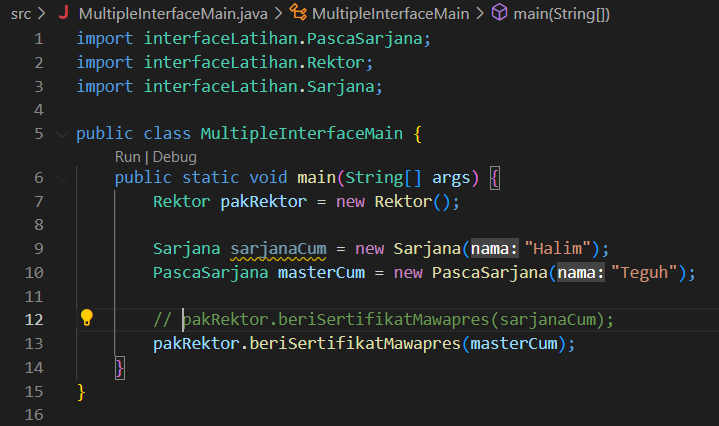


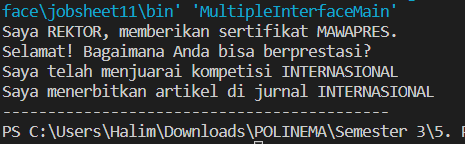
1. Terakhir, modifikasilah method main() pada class MutipleInterfaceMain anda. Comment-lah semua baris yang terdapat method beriSertifikatCumlaude(), lalu tambahkan baris kode baru seperti pada gambar dibawah ini.



1. akan terdapat error pada Langkah 5, sehingga program tidak dapat dieksekusi. Perbaikilah kode programnya, sehingga hasil eksekusi menjadi sama seperti pada screenshot dibawah ini.

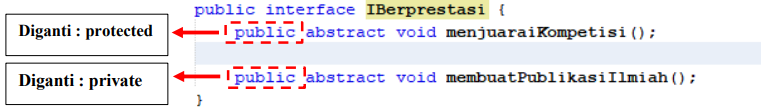






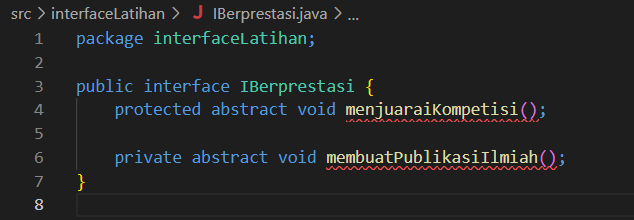
|  |
| --- |
| PERTANYAAN PRAKTIKUM 2 |

1. Pada script code interface IBerprestasi, modifikasi script tersebut sesuai dengan gambar dibawah ini:



Dari perubahan script diatas, apa yang terjadi? Serta jelaskan alasannya (capture hasilnya)

Jawab:



Keterangan: terjadi Error karena abstrak tidak bisa menggunakan acces modifier private karena pasti class abstrak tersebut akan di panggil diluar class tersebut dan abstrak tidak dianjurkan menggunakan acces modifier protected dikarenakan access modifier protected hanya dapat dilihat oleh clas itu sendiri, class yang ada di package itu sendiri, dan class yang berada pada subclass itu sendiri, jika menggunakan protected pada abstract, ini akan membuat program bingung akan mewariskan ke dalam packagenya atau ke bagian subclassnya. Jika keduanya bisa diwariskan, itu termasuk access modifier public.

1. Perhatikan script code dibawah ini:

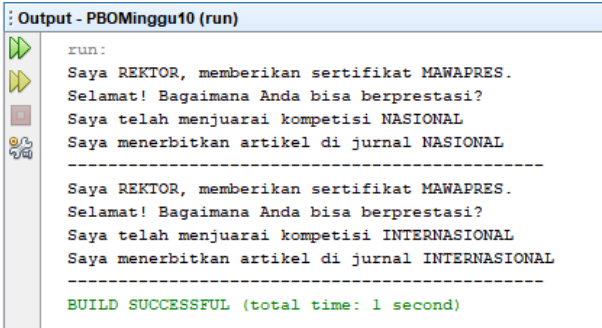


Jelaskan menurut anda, mengapa hasil dari script code tersebut error?

Jawab: terjadi error karena class IBerprestasi adalah class interface yang merupakan penghubung antar objek. class interface tidak dapat dibuat objek menggunakan new karena interface bersifat abstrak.

1. Apabila Sarjana Berprestasi harus menjuarai kompetisi Nasional dan menerbitkan artike di jurnal Nasional, maka modifikasilah class-class yang terkait pada aplikasi anda agar di class Program objek pak Rektor dapat memberikan sertifikat mawapres pada objek sarjana Cumlaude.

Jawab:



1. Sarjana

package interfaceLatihan;

public class Sarjana extends Mahasiswa implements ICumlaude, IBerprestasi {

    public Sarjana(String nama) {

        super(nama);

    }

    @Override

    public void lulus() {

        // TODO Auto-generated method stub

        kuliahDiKampus();

        System.out.println("Aku sudah menyelesaikan SKRIPSI");

    }

    @Override

    public void meraihIPKTinggi() {

        // TODO Auto-generated method stub

        System.out.println("IPK-ku lebih dari 3,51");

    }

    @Override

    public void menjuaraiKompetisi() {

        // TODO Auto-generated method stub

        System.out.println("Saya telah menjuarai kompetisi NASIONAL");

    }

    @Override

    public void membuatPublikasiIlmiah() {

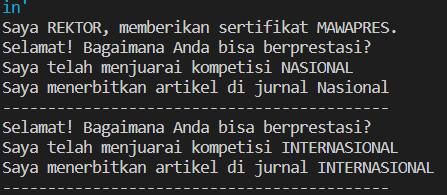
        // TODO Auto-generated method stub

        System.out.println("Saya menerbitkan artikel di jurnal NASIONAL");

    }

}

1. Output



LINK GITHUB: <https://github.com/HalimTeguh/Praktikum/tree/master/11.%20Interface>