**Jobsheet 7**

**Interface**

**Nama : Rizqi Rohmatul Huda**

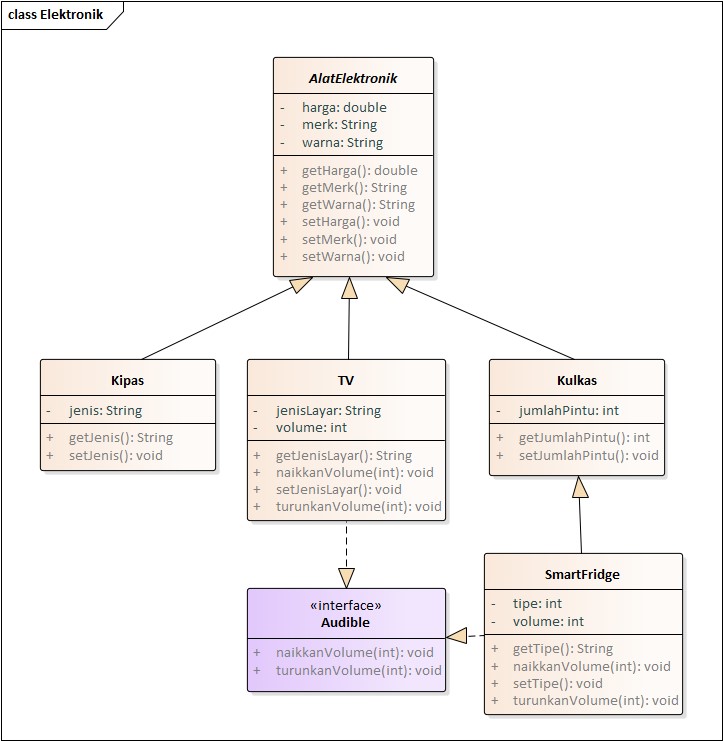
**Kelas : 2G – TI**

**No.Absen : 26**

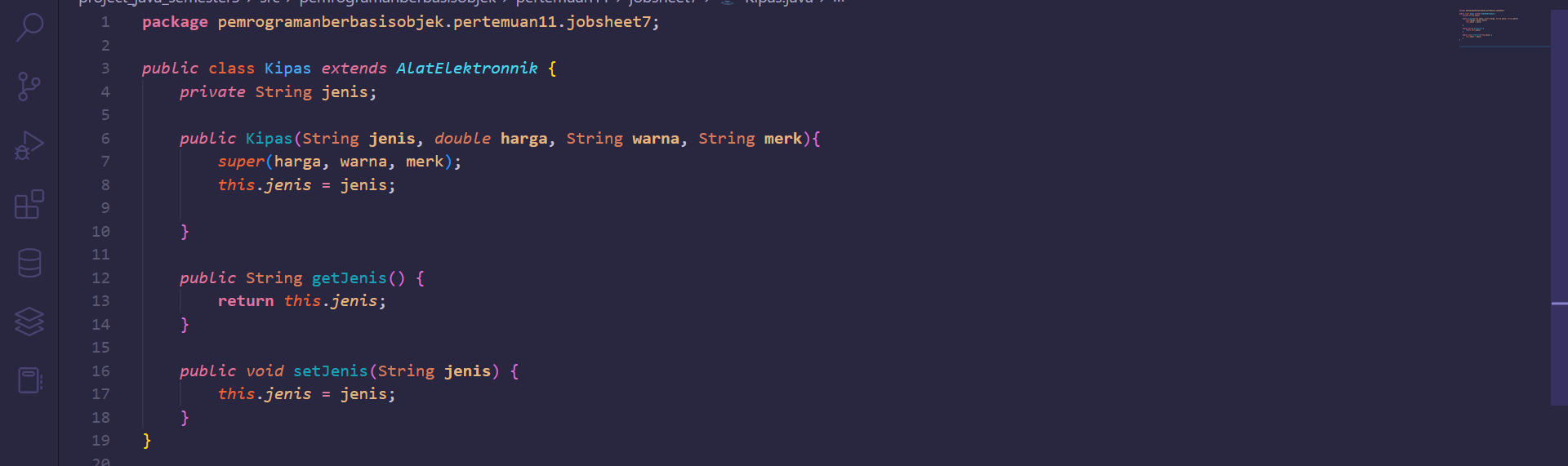
**NIM : 2141720264**

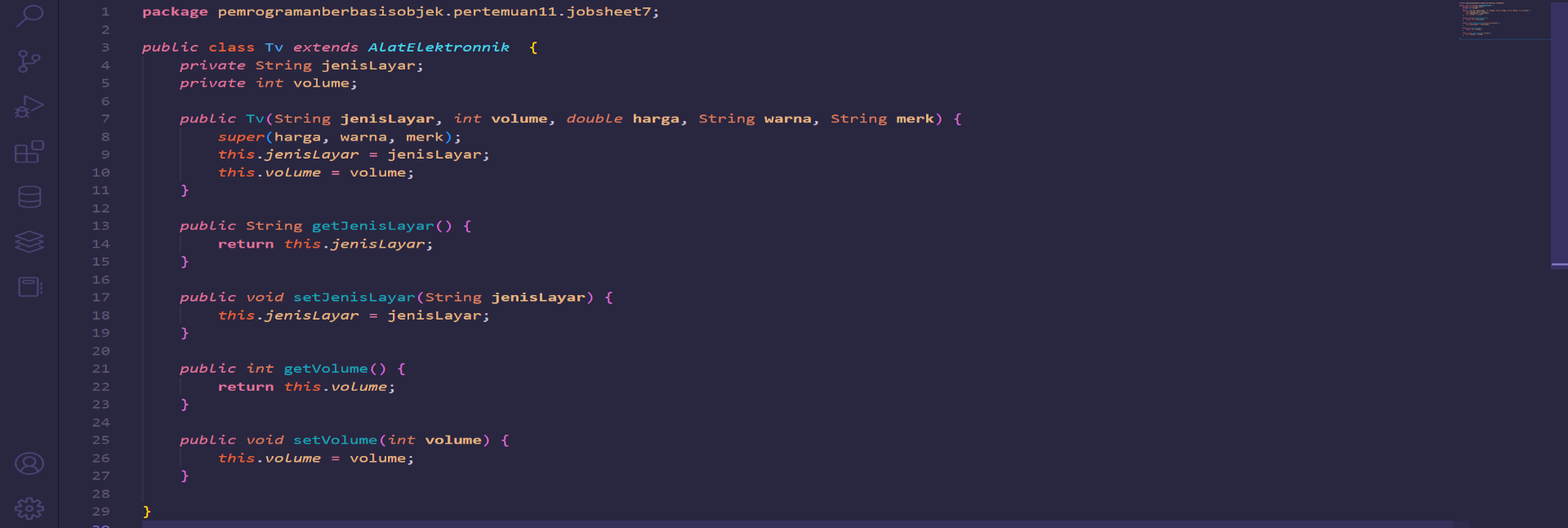
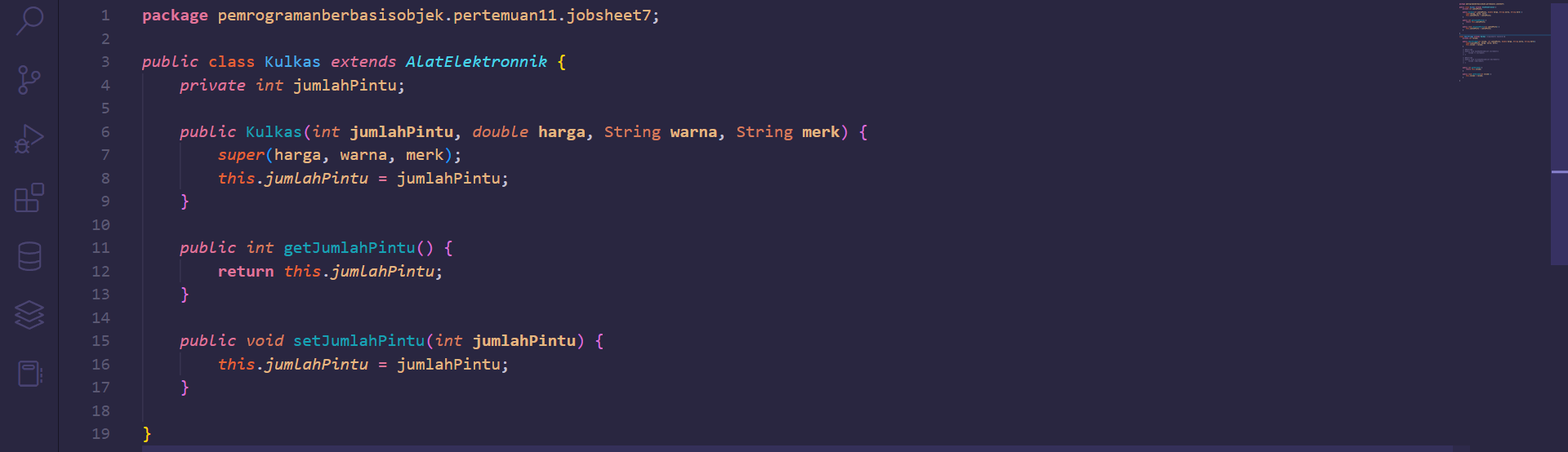
# C. PERCOBAAN

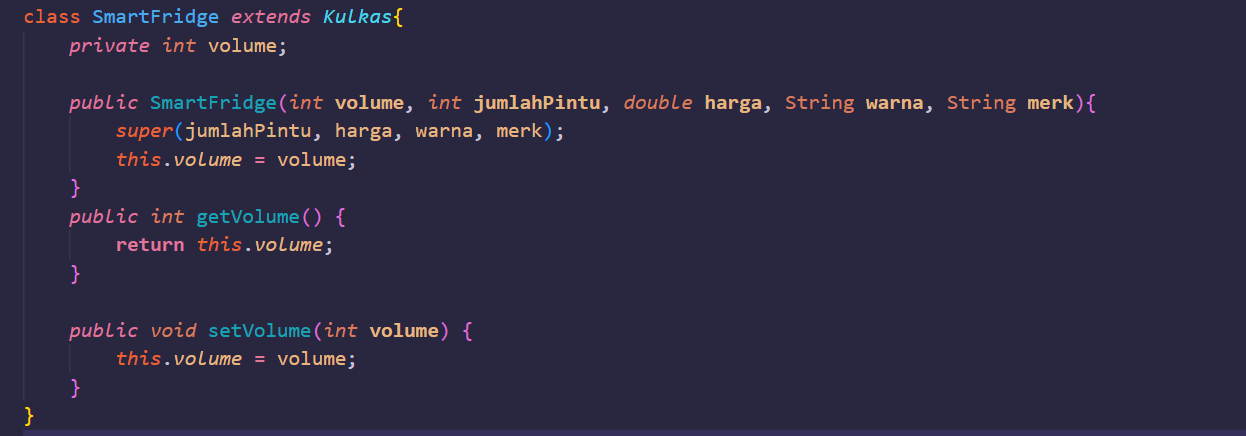
Implementasikan class diagram berikut ke dalam kode program.

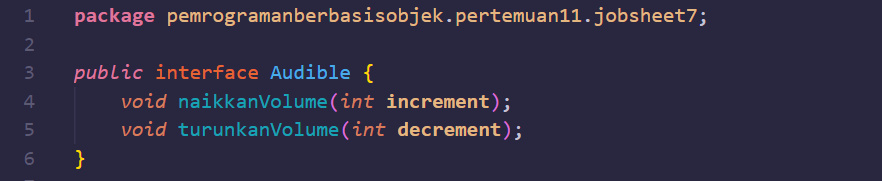


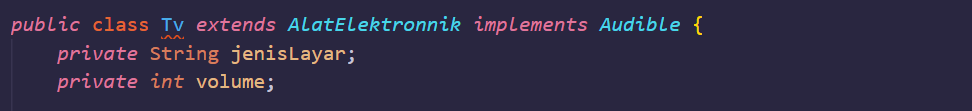
1. Buat project baru dengan nama InterfaceLatihan (boleh disesuaikan)
2. Pada sebuah package, buatlah abstract class AlatElektronik

1. Selanjutnya buatlah subclass dari AlatElektronik, yaitu Kipas, TV, dan Kulkas sebagai berikut.



1. Buatlah class SmartFridge yang merupakan sublass dari class Kulkas

1. Beberapa dari alat elektronik dapat mengeluarkan suara. Kapabilitas ini kita buat ke dalam kode program dengan interface Audible dengan method naikkanVolume() dan turunkanVolume() sebagai berikut

1. Ubah class TV untuk meng-impelement interface Audible

1. Implementasi abstract method pada interface Audible pada class TV

1. Lakukan hal yang sama pada class SmartFridge

# D. PERTANYAAN 2

1. Mengapa terjadi error pada langkah 5?

Jawab : Karena class Tv belum meng-implementasikan method dari interface Audible.

1. Mengapa Audible tidak dapat dibuat sebagai class?

Jawab : sebab Audible adalah interface. Interface bersifat abstrak sehingga tidak bisa dibuat kelas maupun diinstance dengan kata kunci new, selain itu interface hanya sebagai kontrak/syarat yang berisi method yang saling terkait untuk memenuhi kapabilitas.

1. Mengapa method dalam interface Audible tidak memiliki access level modifier?

Jawab : karena default method dalam interface adalah public sehingga walaupun tidak ditulis maka access level modifiernya tetap public.

1. Method naikkanVolume() dan turunkanVolume() memilki implementasi yang sama pada TV dan SmartFridge(), mengapa tidak langsung diimplementasikan pada interface Audible()?

Jawab : Karena interface tidak dapat diinstance dan hanya sebagai panduan bahwa method apa saja yang harus diimplementasikan, interface hanya bisa diinstansiasi melalui class yang mengimplment interface itu sendiri.

1. Method naikkanVolume() dan turunkanVolume() memilki implementasi yang sama pada TV dan SmartFridge(), mengapa tidak langsung diimplementasikan pada class AlatElektronik?

Jawab : Karena method naikkanVolume() dan turunkanVolume() bukan hanya dapat diimplementasikan pada alat elektronik saja dan tidak semua alat elektronik juga memiliki method naikkanVolume() dan turunkanVolume().

1. Semua yang Audible seharusnya memiliki nilai volume, mengapa atribut volume tidak dideklarasikan dalam interface Audible()?

Jawab : Karena jika dideklarasikan sebagai interface hanya akan memiliki variable yang bersifat final dan static sehingga nanti pada saat dilakukan instantsiasi pada kelas implements masing-masing objek akan memiliki nilai yang sama dan tidak dapat dirubah.

1. Apa fungsi dari interface?

Jawab :

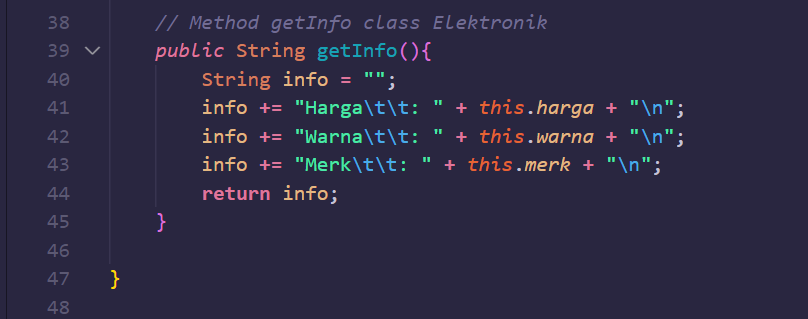
Interface berfungsi untuk :

* menyeragamkan method
* bertindak sebagai kontrak/syarat yang berisi sekumpulan behavior/method yang saling terkait untuk memenuhi suatu kapabilitas
* memberikan panduan mengenai method apa saja yang perlu diimplementasikan untuk memenuhi kapabilitas tertentu

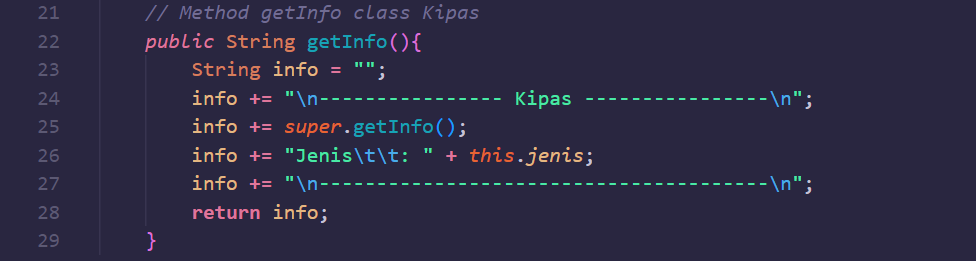
1. Buat method getInfo() untuk setiap class. Instansiasi objek dari setiap concrete class pada main class, kemudian tampilkan infonya.

Jawab :

1. Method getInfo() class AlatElektronik



1. Method getinfo() class Kipas



1. Method getinfo() class Kulkas



1. Method getinfo() class SmartFridge



1. Method getinfo() class Tv

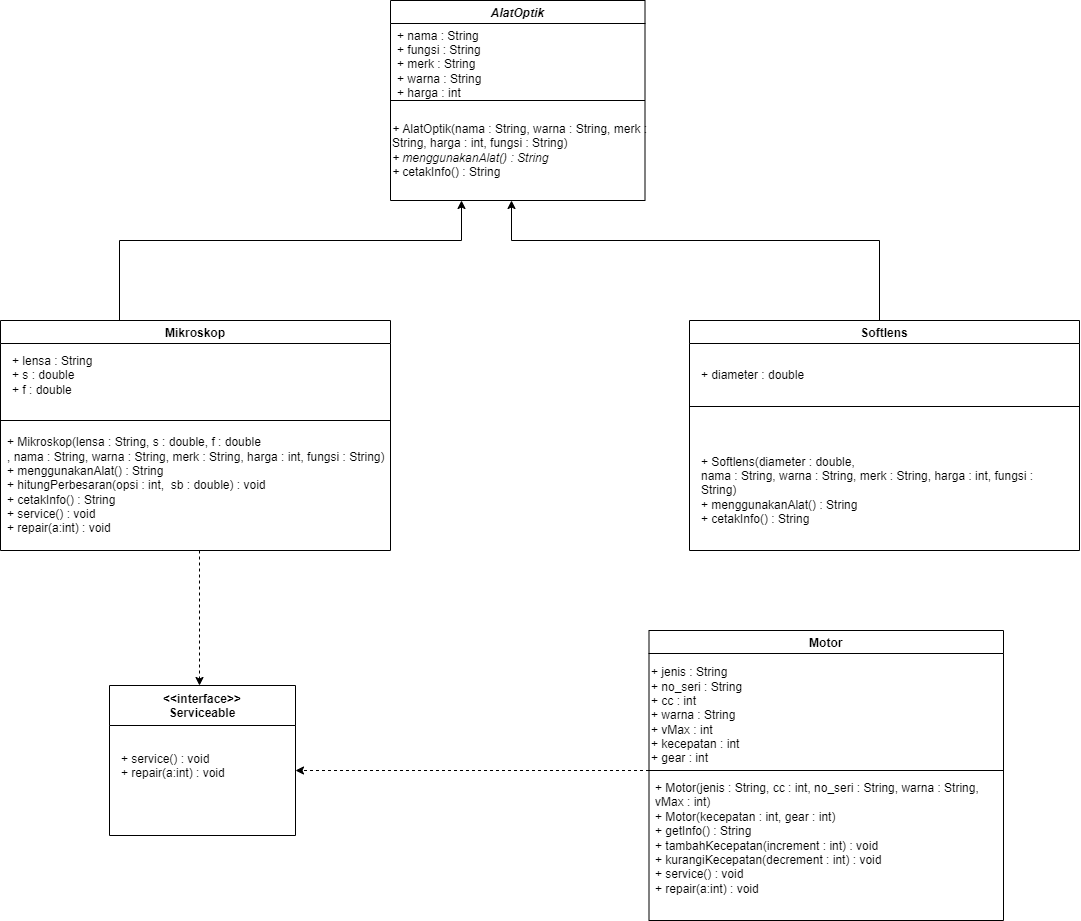


1. Instansiasi pada class DemoElektronik

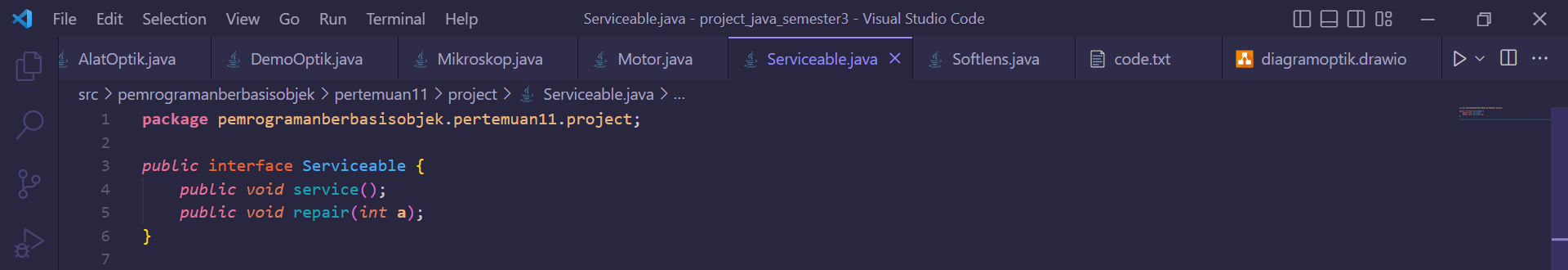
# Hasil running

# E. TUGAS

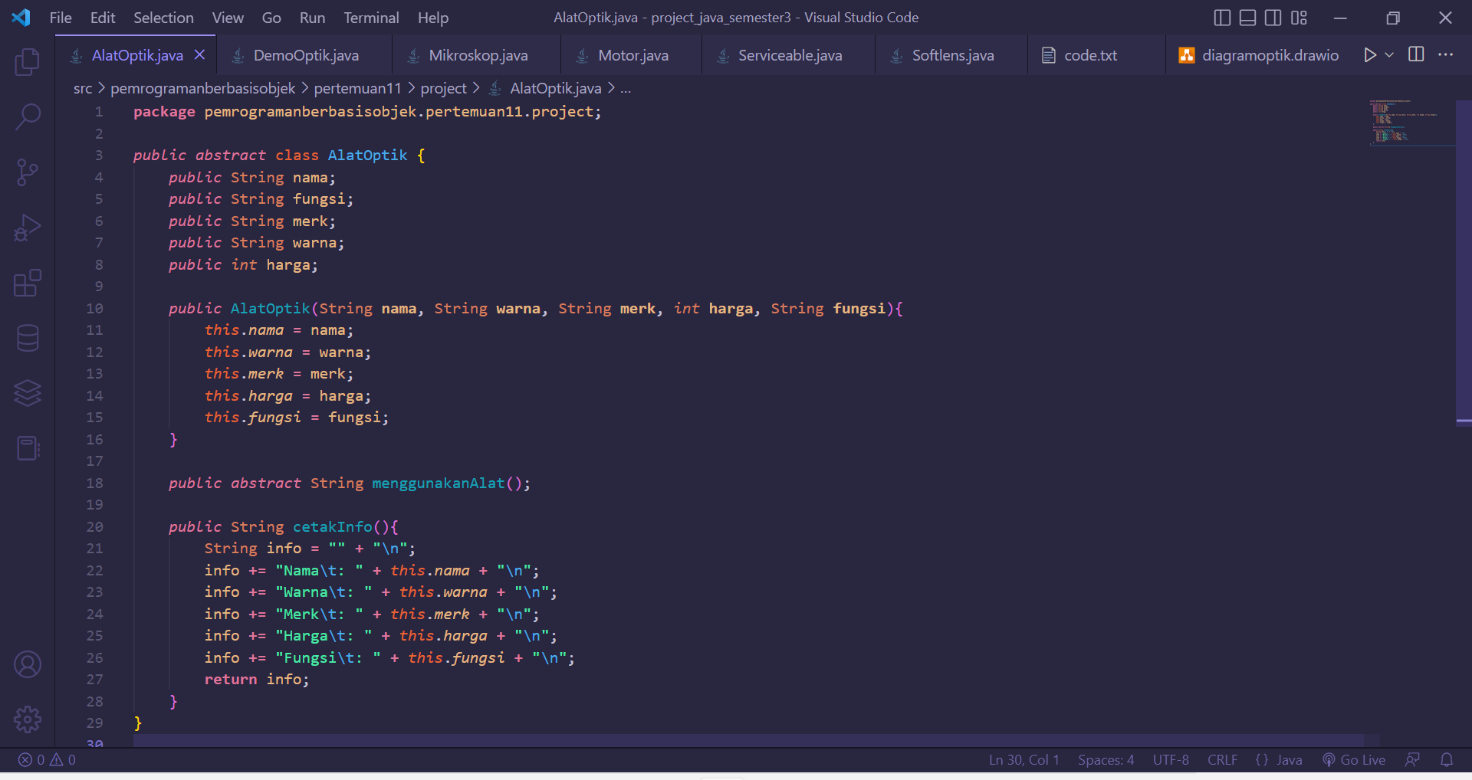
Implementasikan class diagram yang dibuat pada tugas PBO ke dalam kode program.



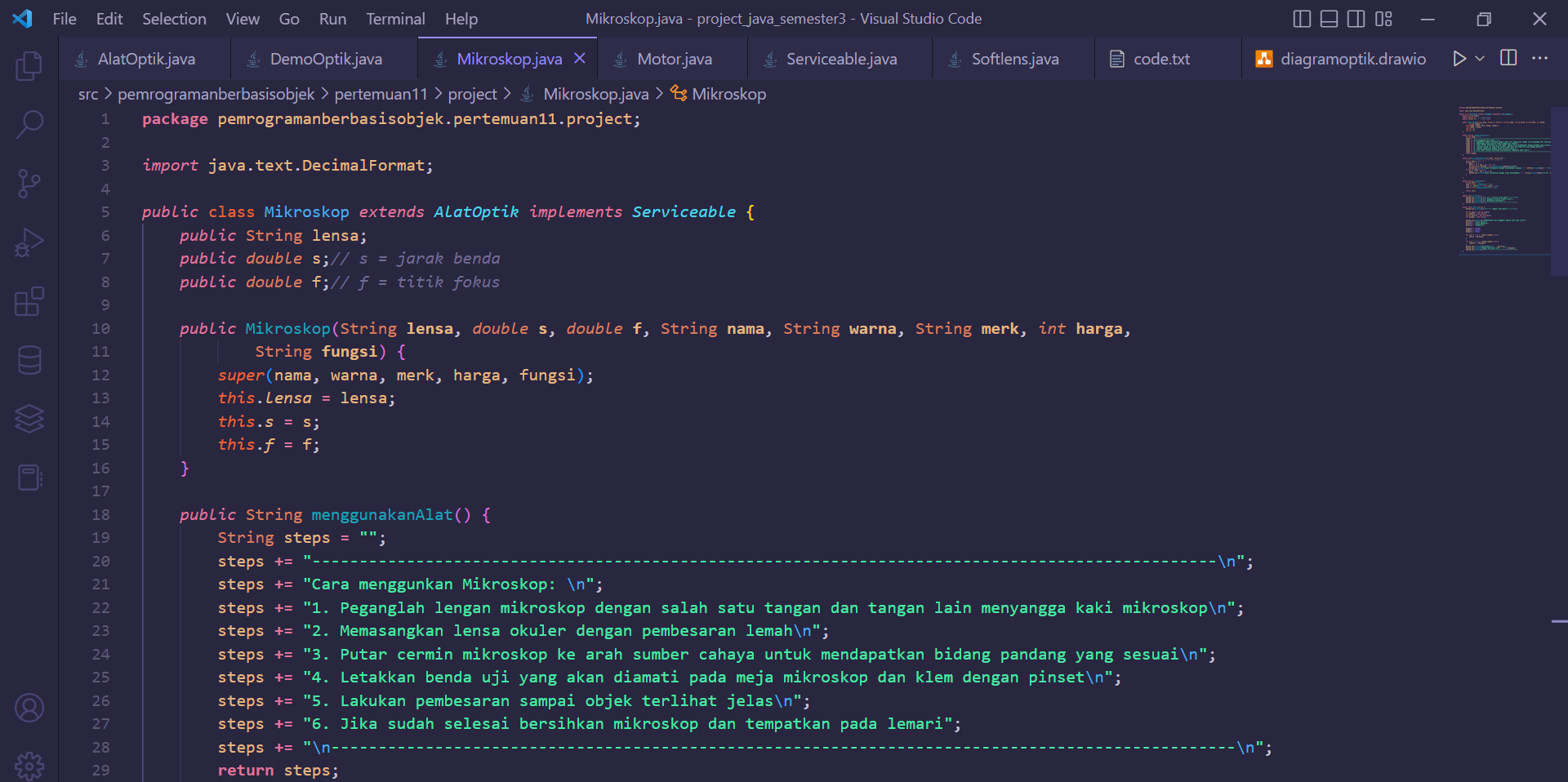
**Code interface Serviceable**

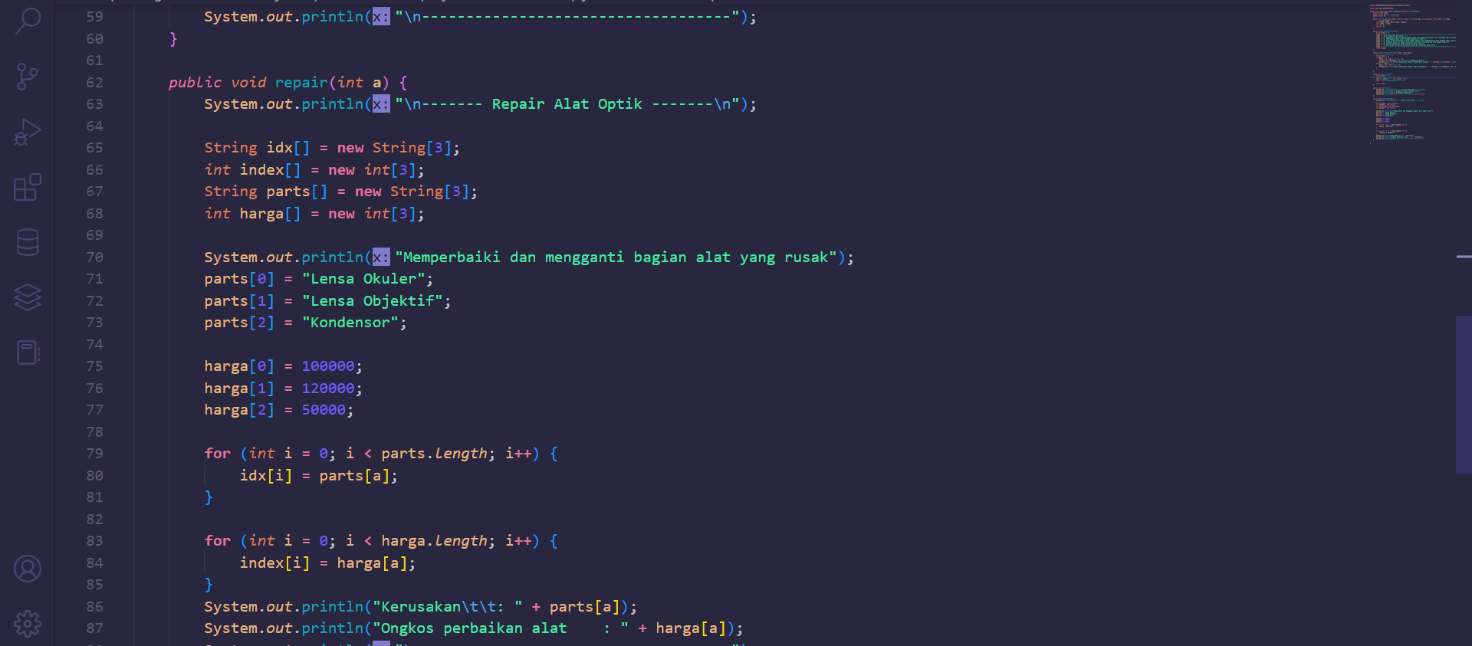
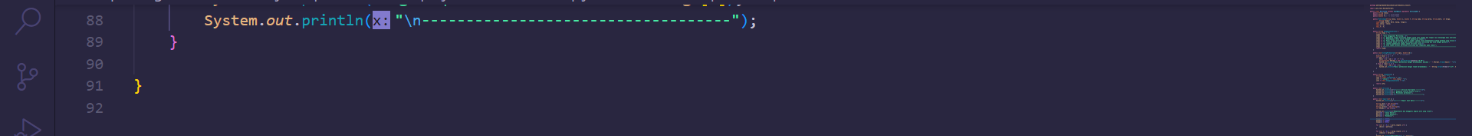
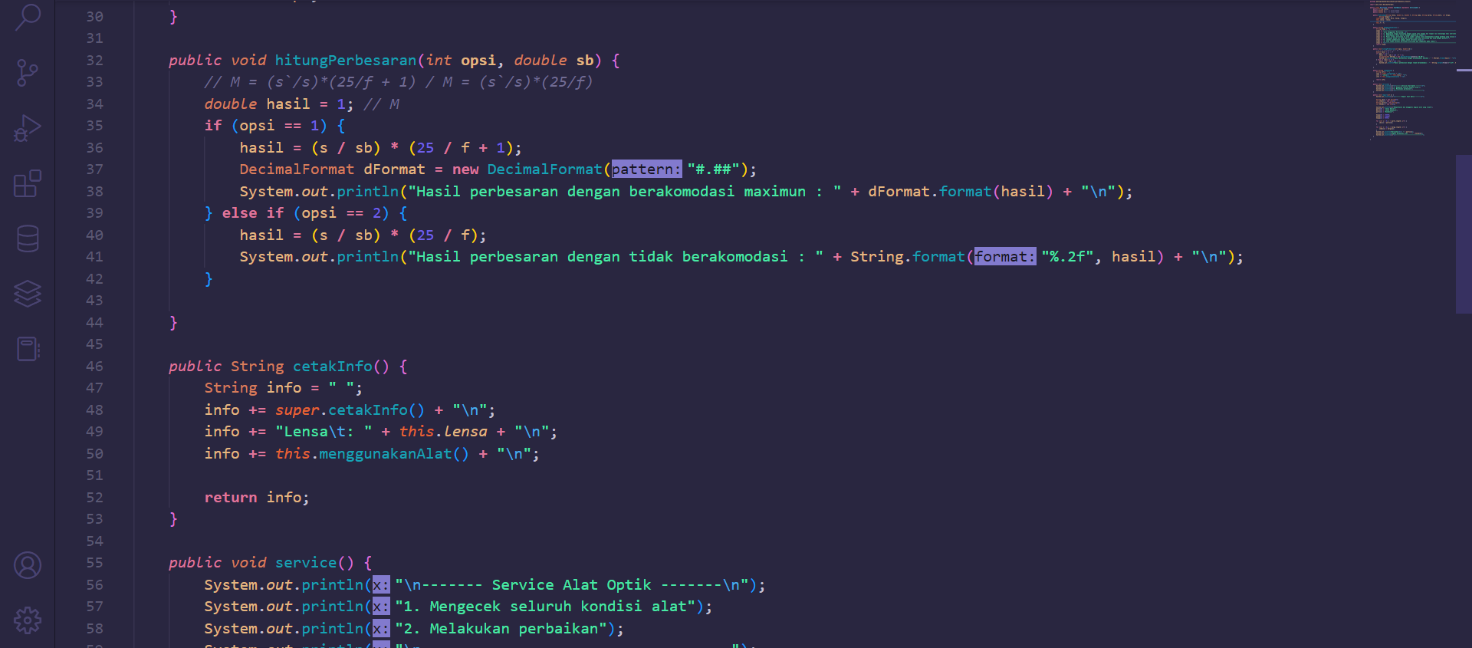


**Code class AlatOptik**

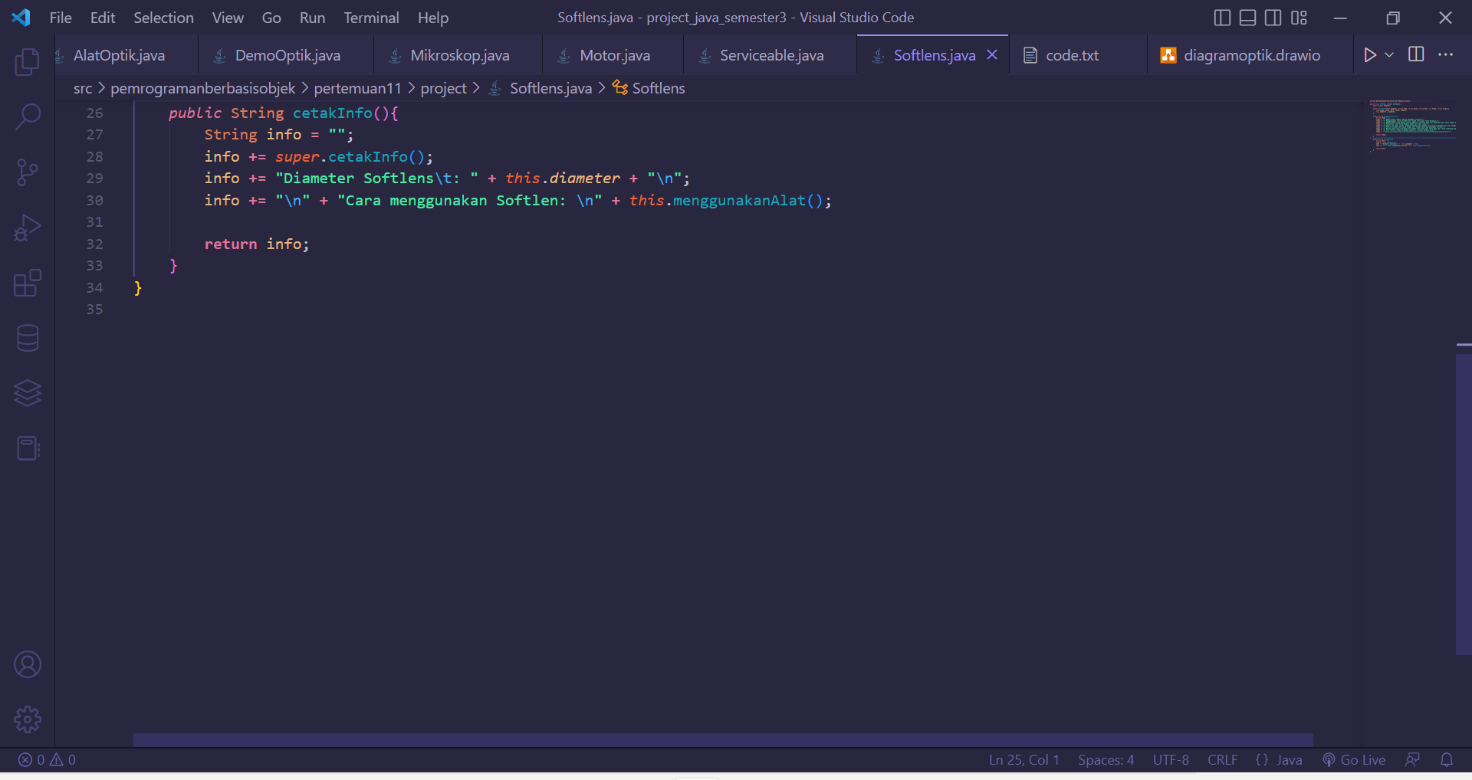


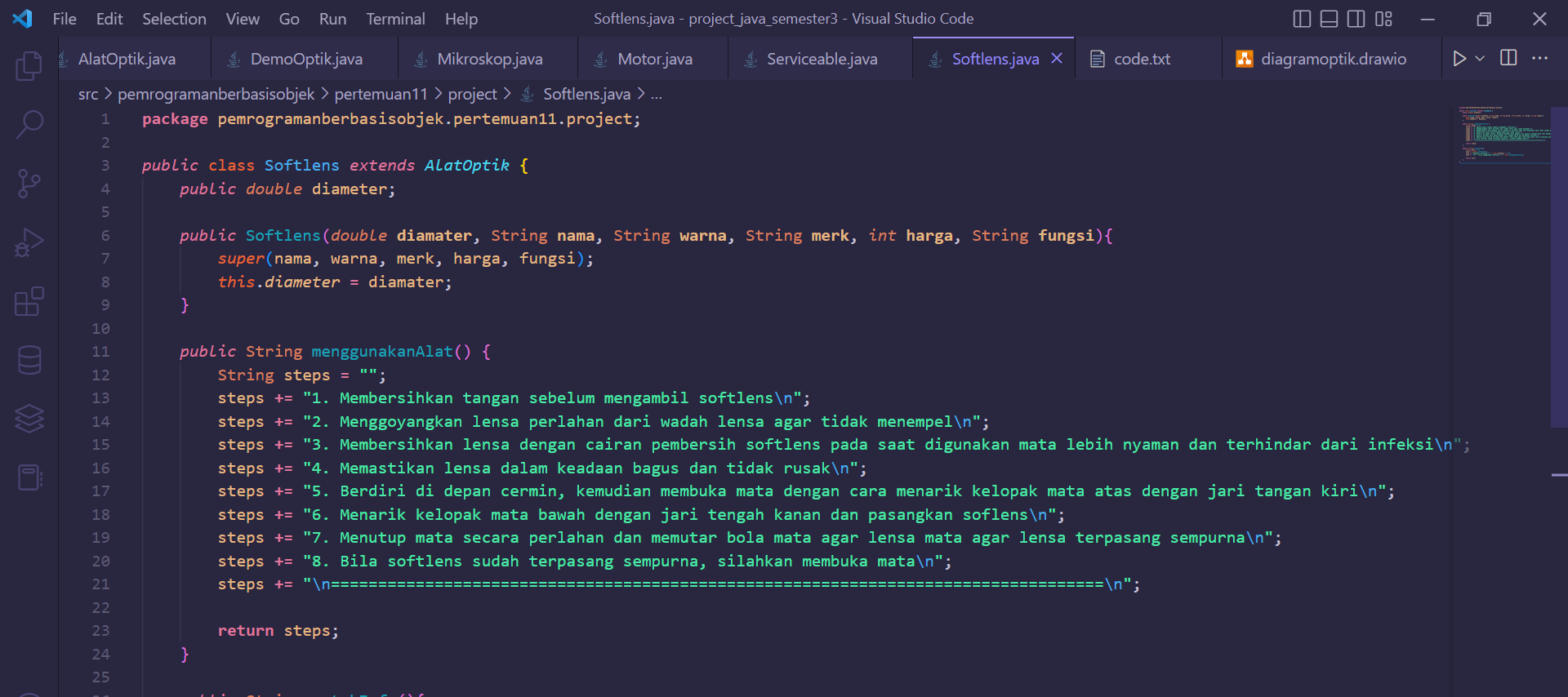
**Code class Mikroskop**



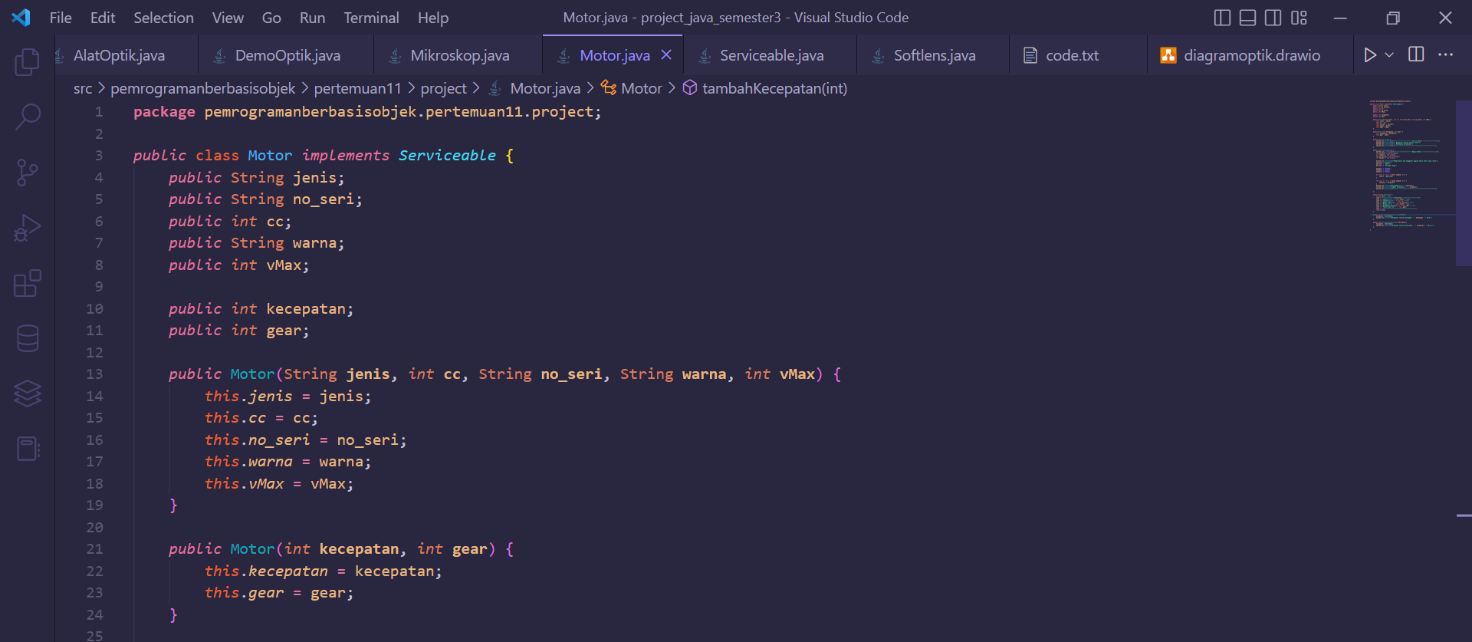
****

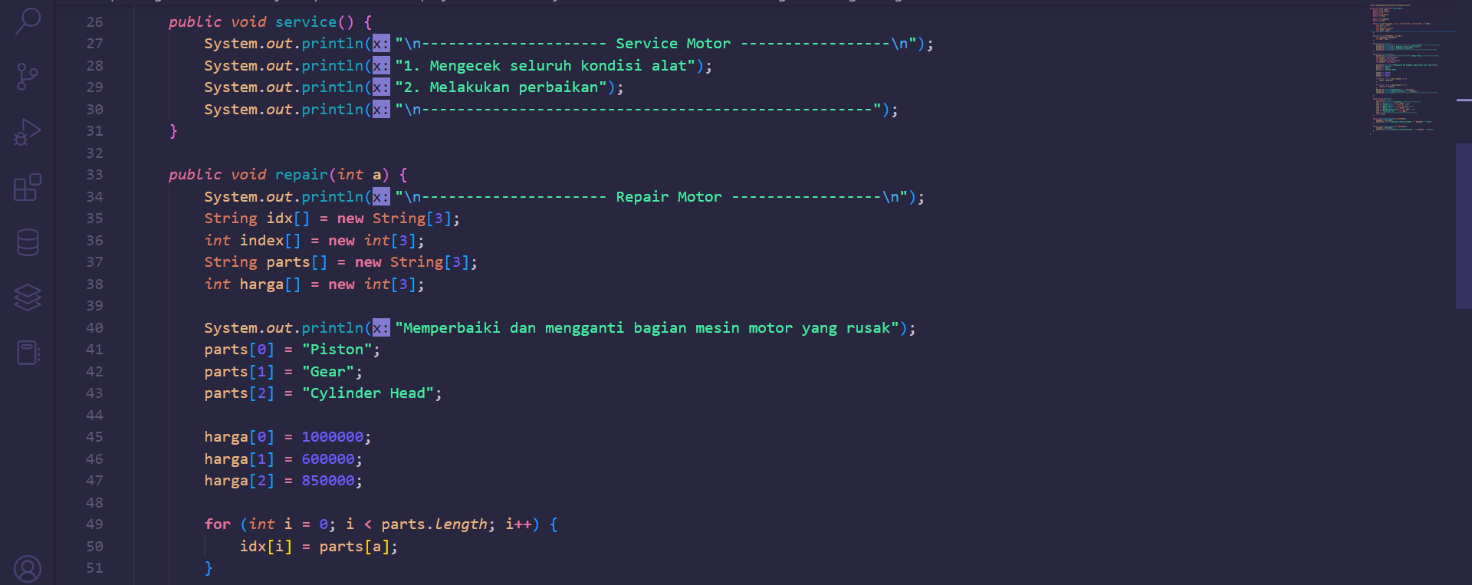
**Code class Soflens**

****

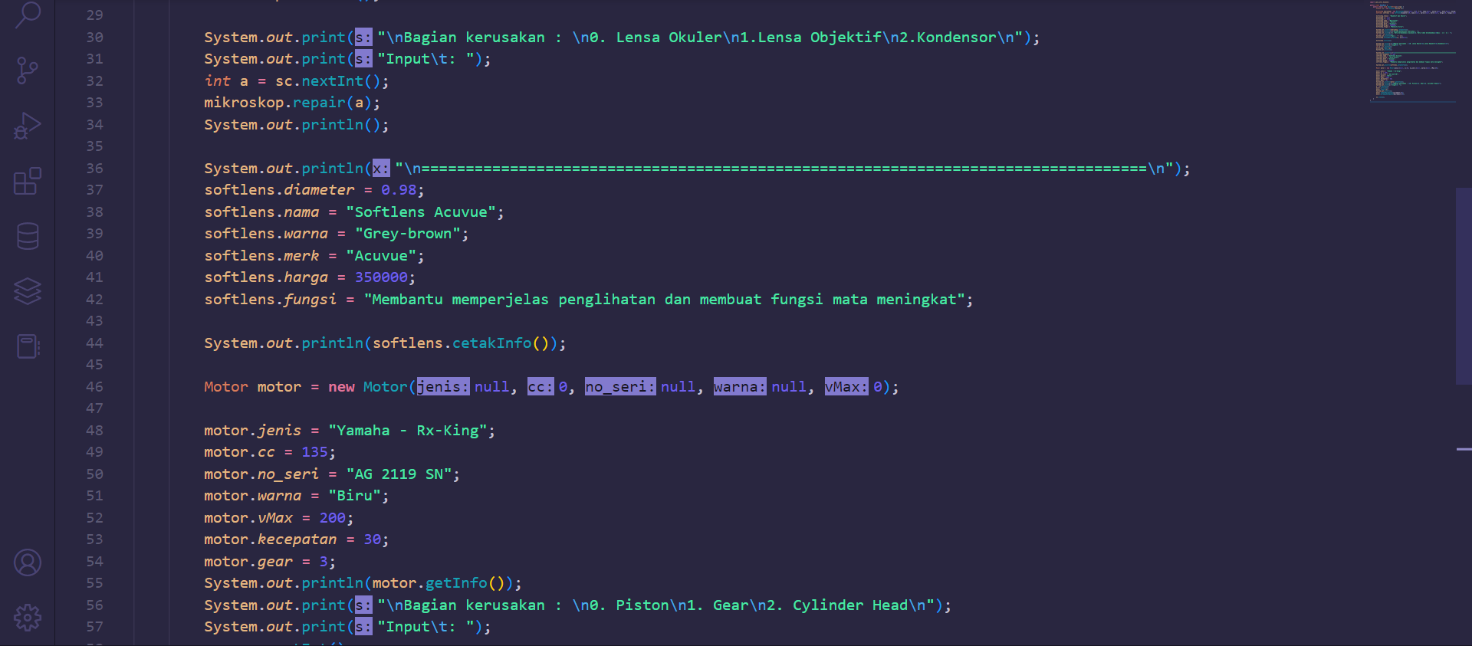
****

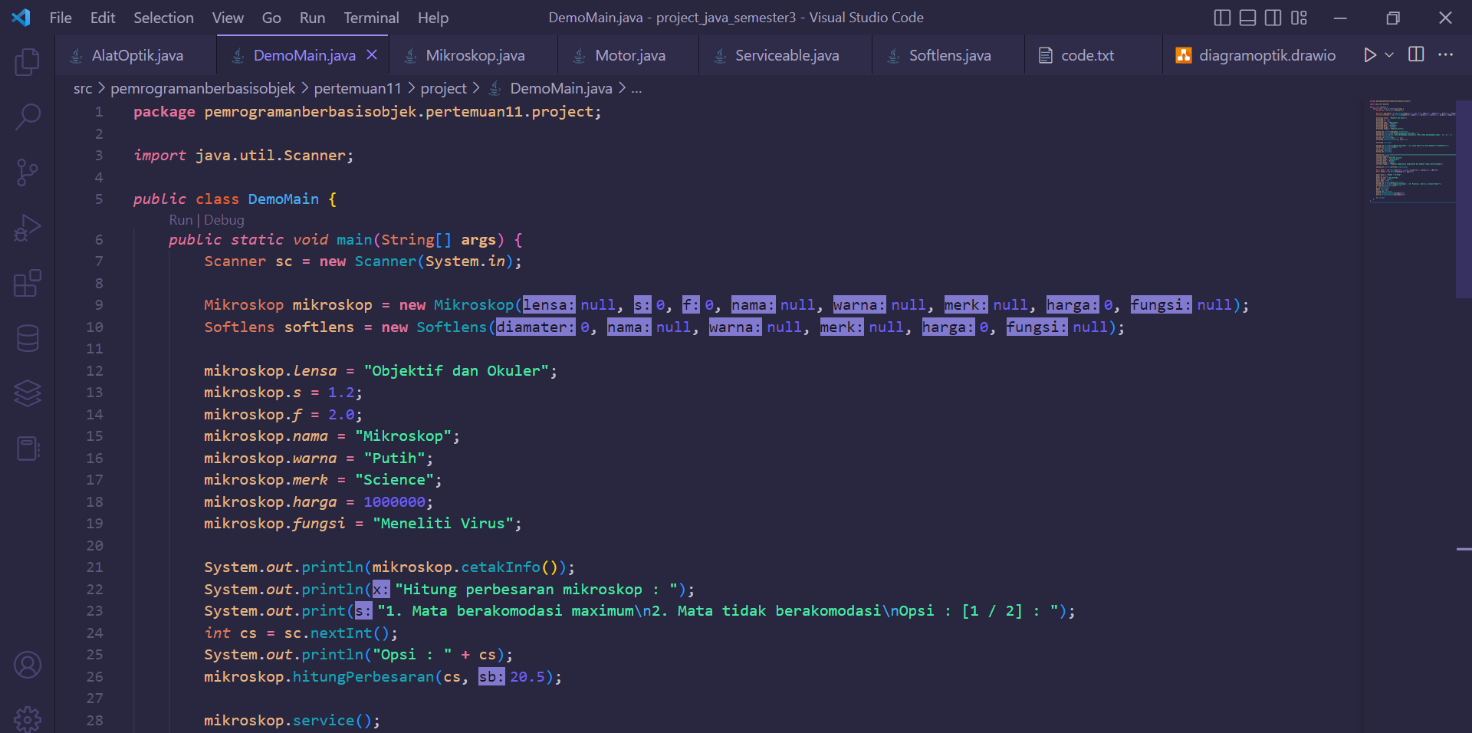
**Code class Motor**

****

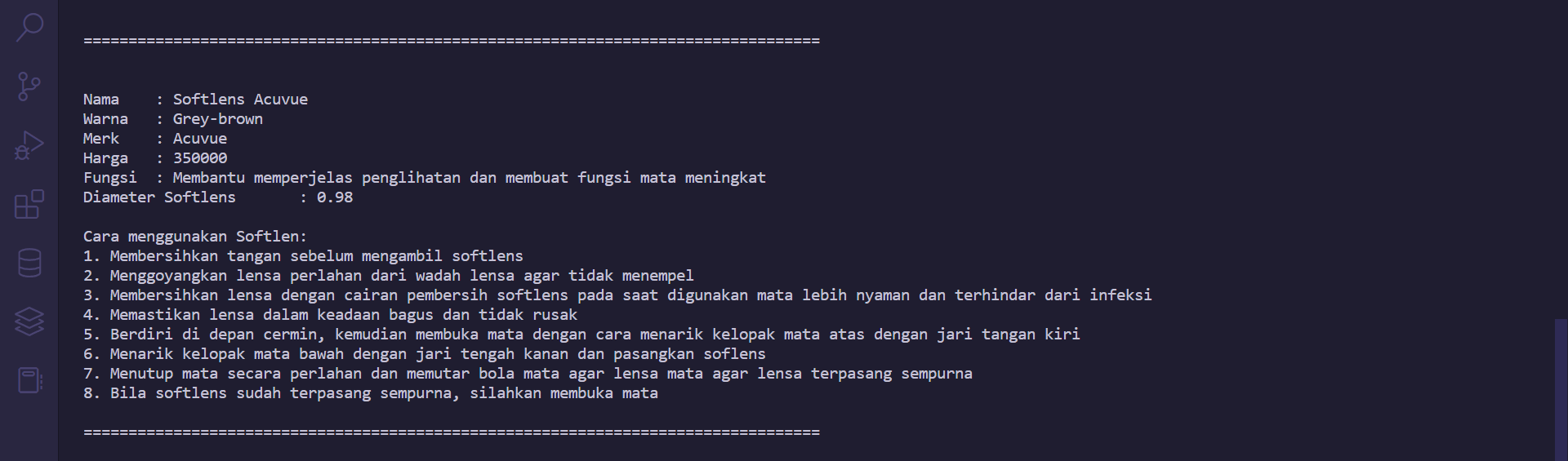
****

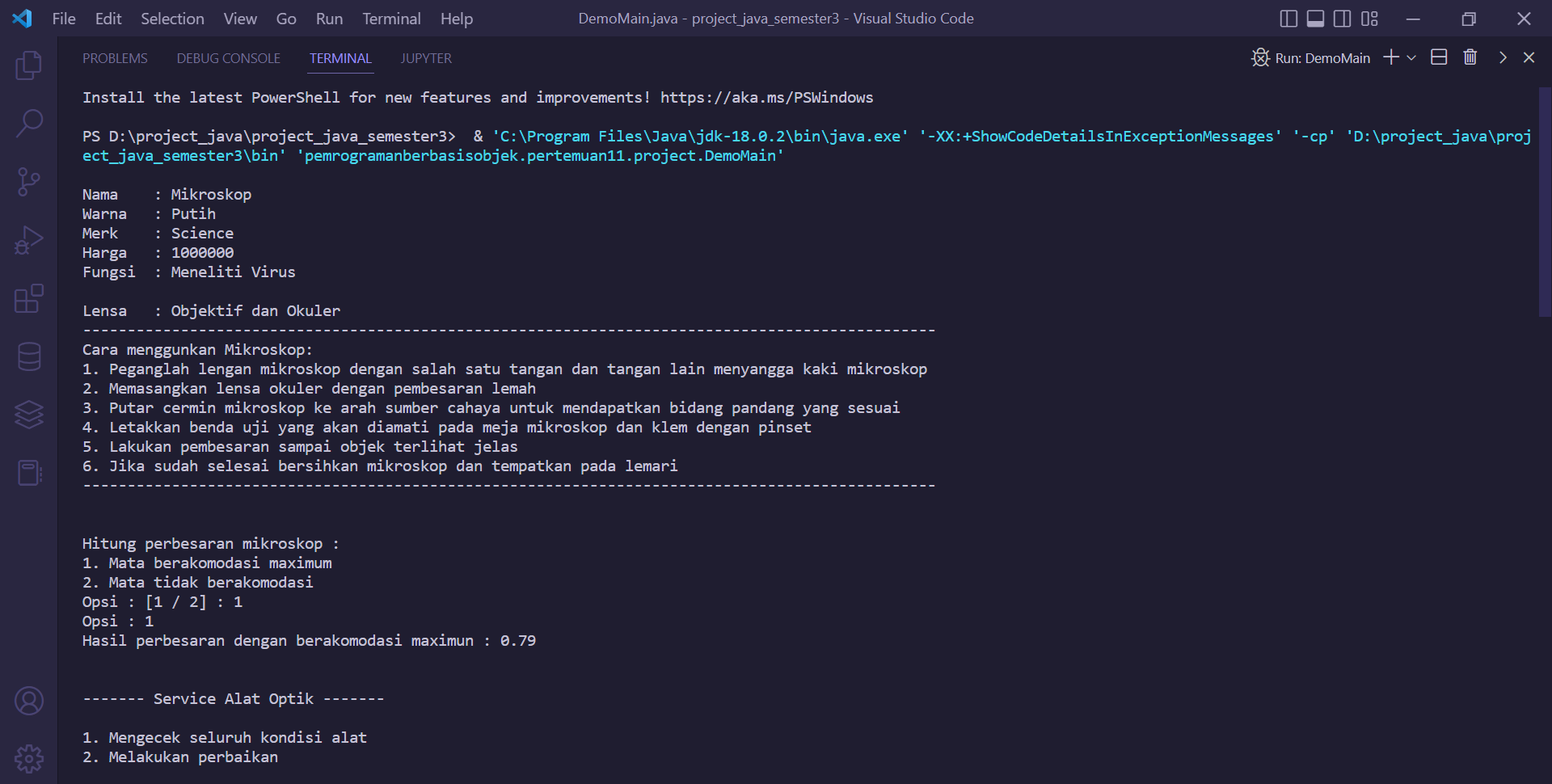
**Code class DemoMain**

****

****

**Hasil running**

****



****