**Nama : Aura Rindatama Azzahra**

**Kelas : D3TI2A**

**NIM : 2003007**

**LAPORAN MODUL 6**

**6.1. Inheritance**

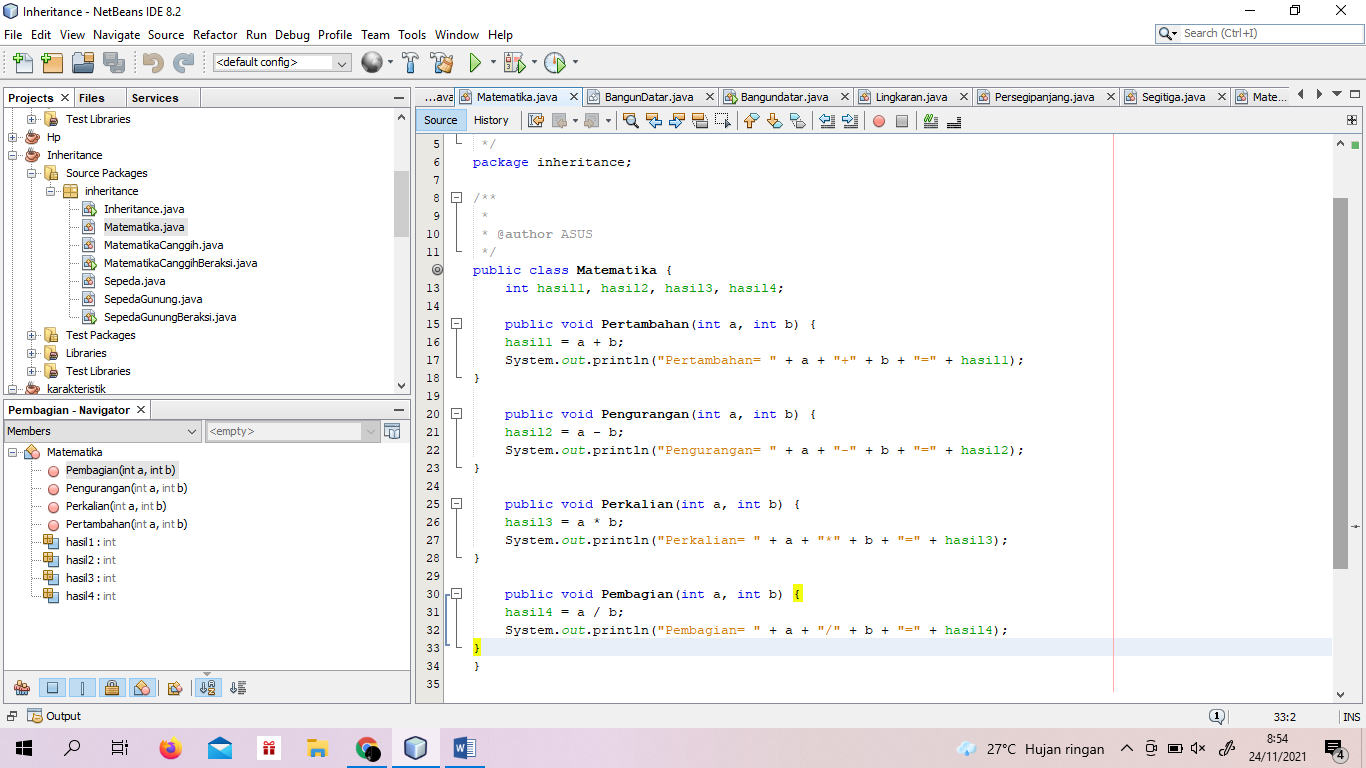
Konsep dari salah satu pilar Pemrograman Berorientasi Objek (PBO) untuk mewariskan property dan method dari parent class (super-class) ke child class (subclass). Terdapat beberapa jenis inheritance, yaitu single, multiple, multi-level, hierarchical, dan hybrid.

**A. Single inheritance**

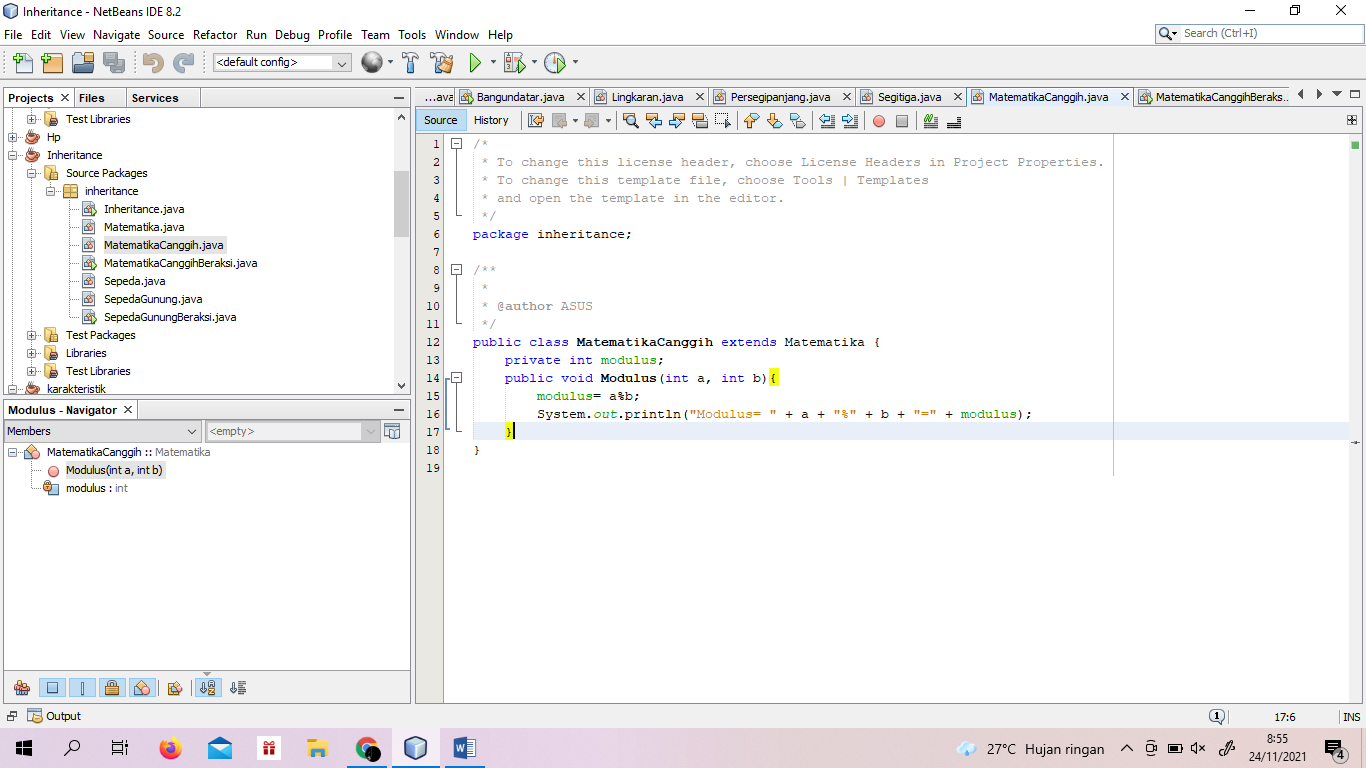
Konsep pewarisan pada satu parent class ke satu child class

Kode Program

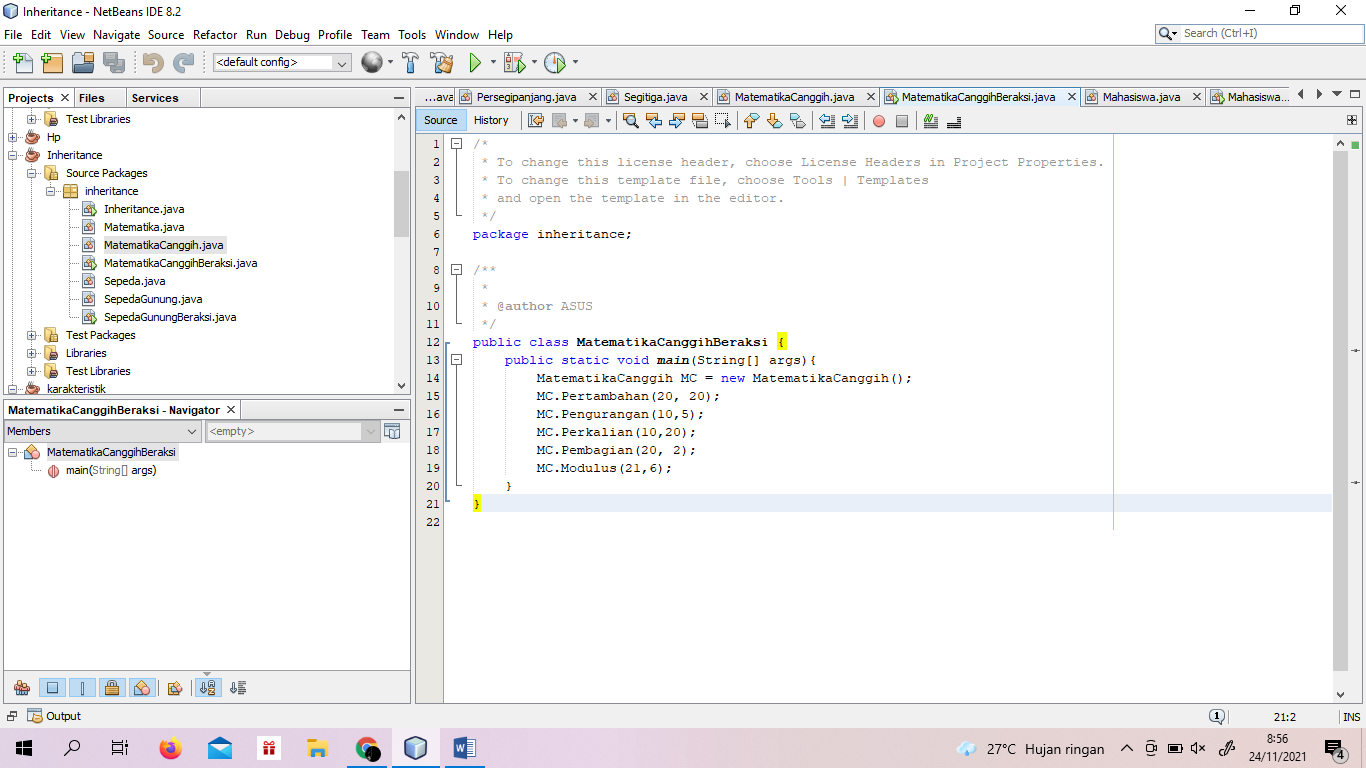
Matematika.java



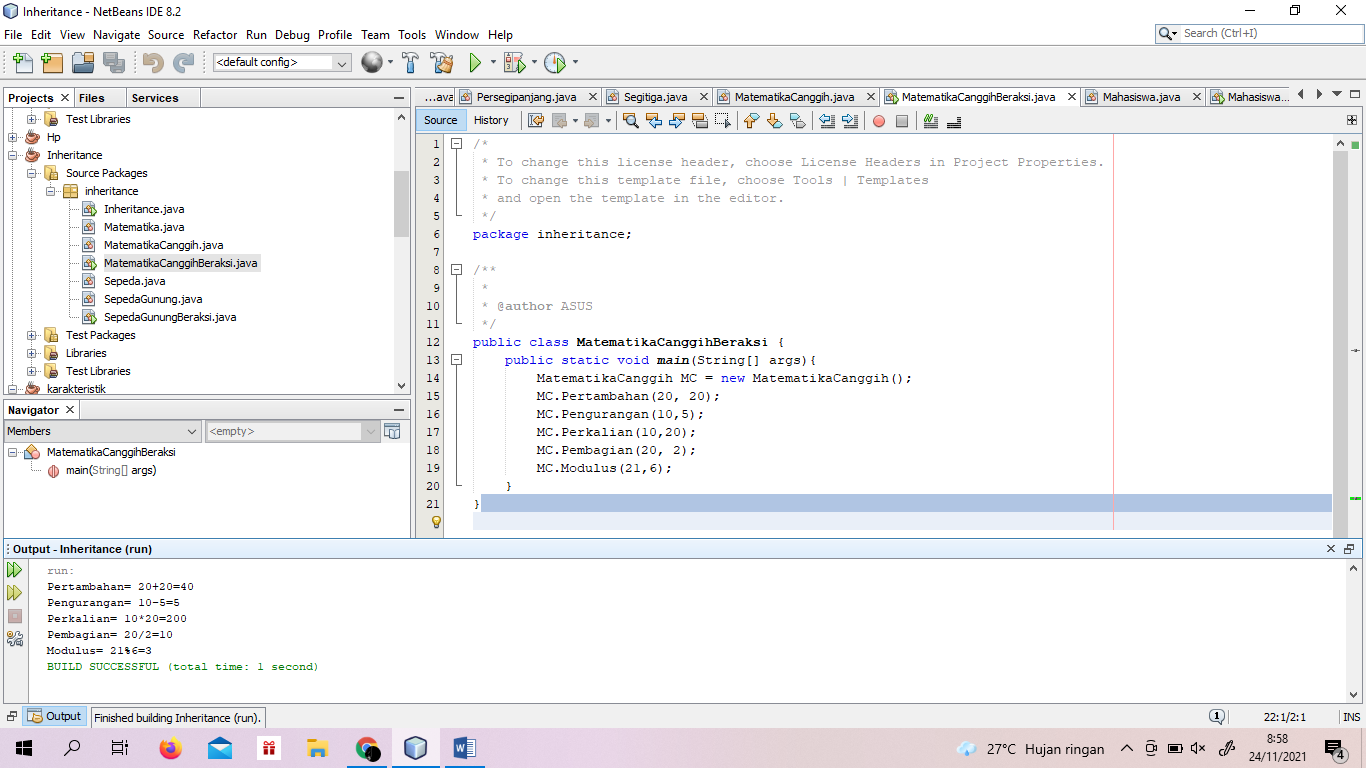
MatematikaCanggih.java



MatematikaCanggihBeraksi.java

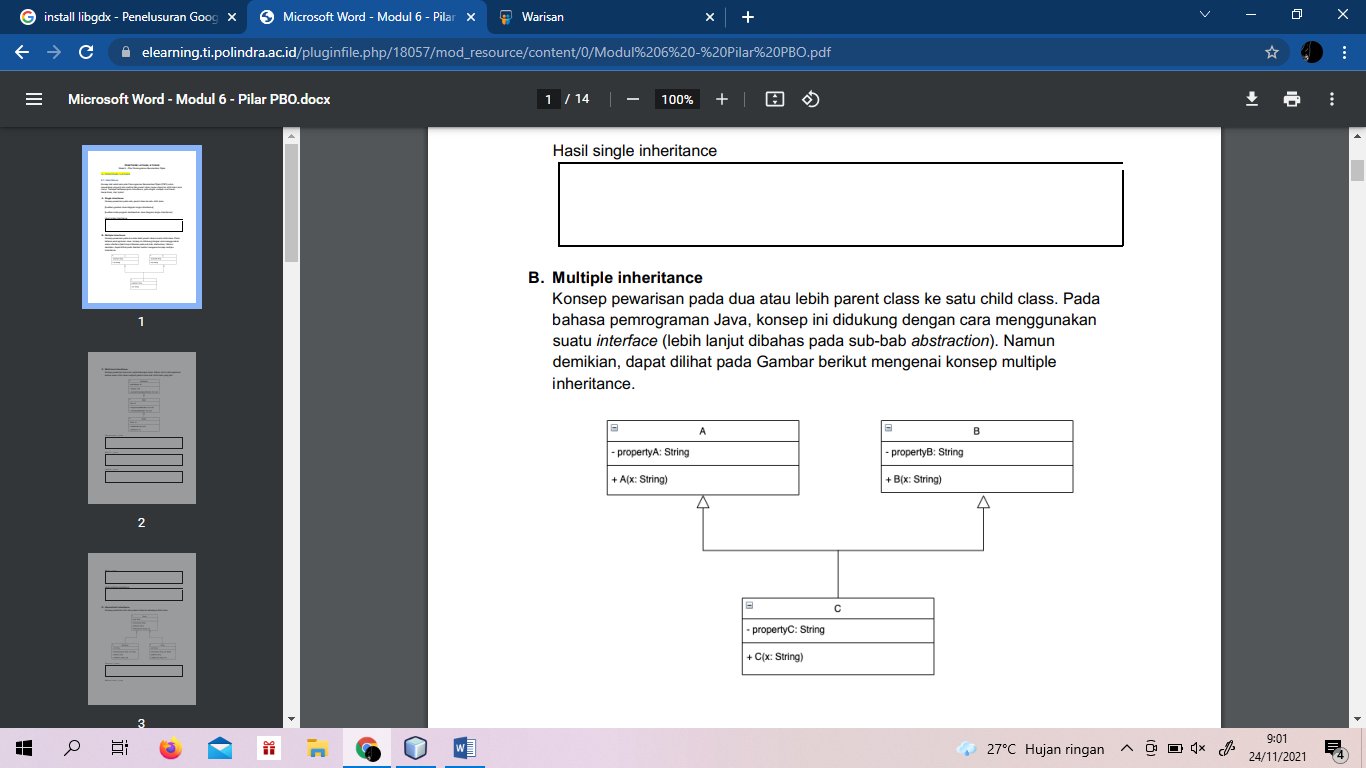


Hasil :



**B. Multiple Inheritance**

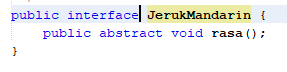
Konsep pewarisan pada dua atau lebih parent class ke satu child class. Pada bahasa pemrograman Java, konsep ini didukung dengan cara menggunakan suatu interface (lebih lanjut dibahas pada sub-bab abstraction). Namun demikian, dapat dilihat pada Gambar berikut mengenai konsep multiple inheritance.



ManggaCengkir.java



ManggaArumanis.java

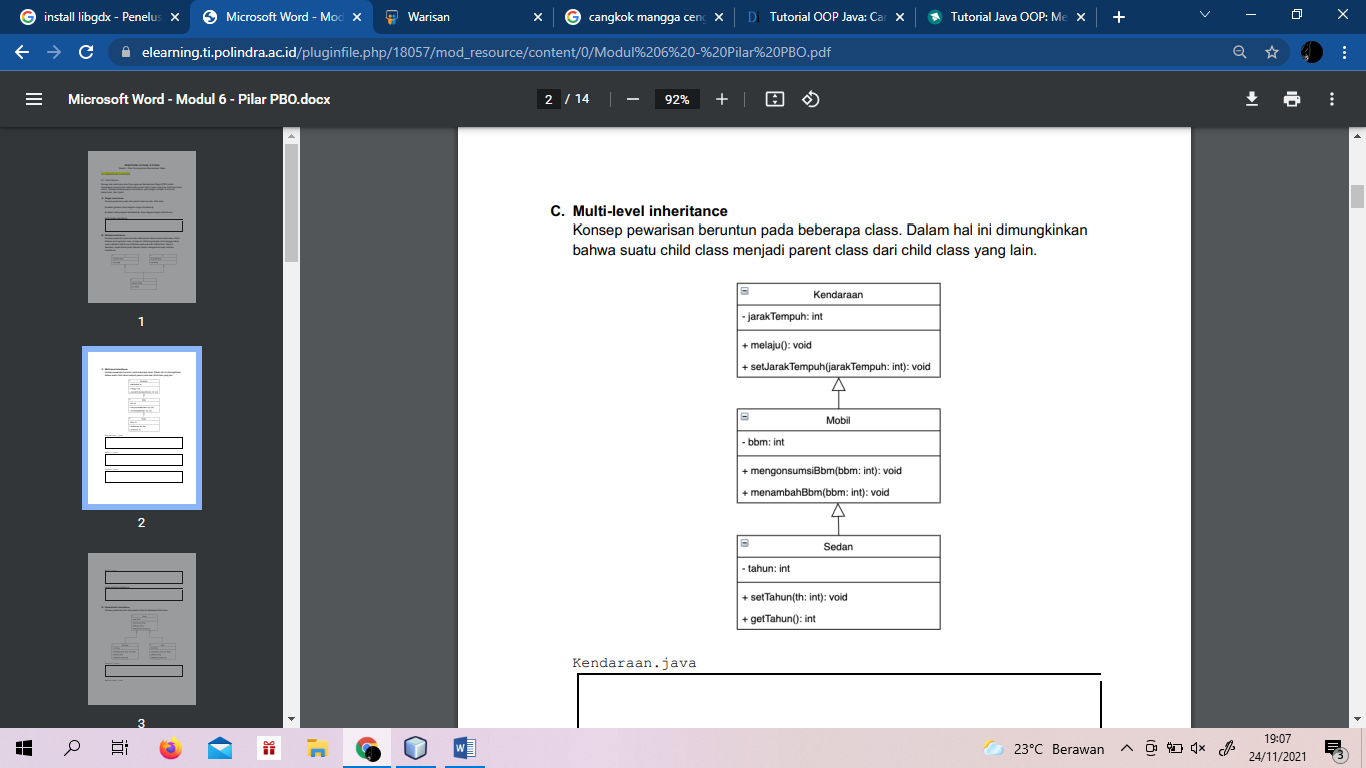


JerukCangkok.java

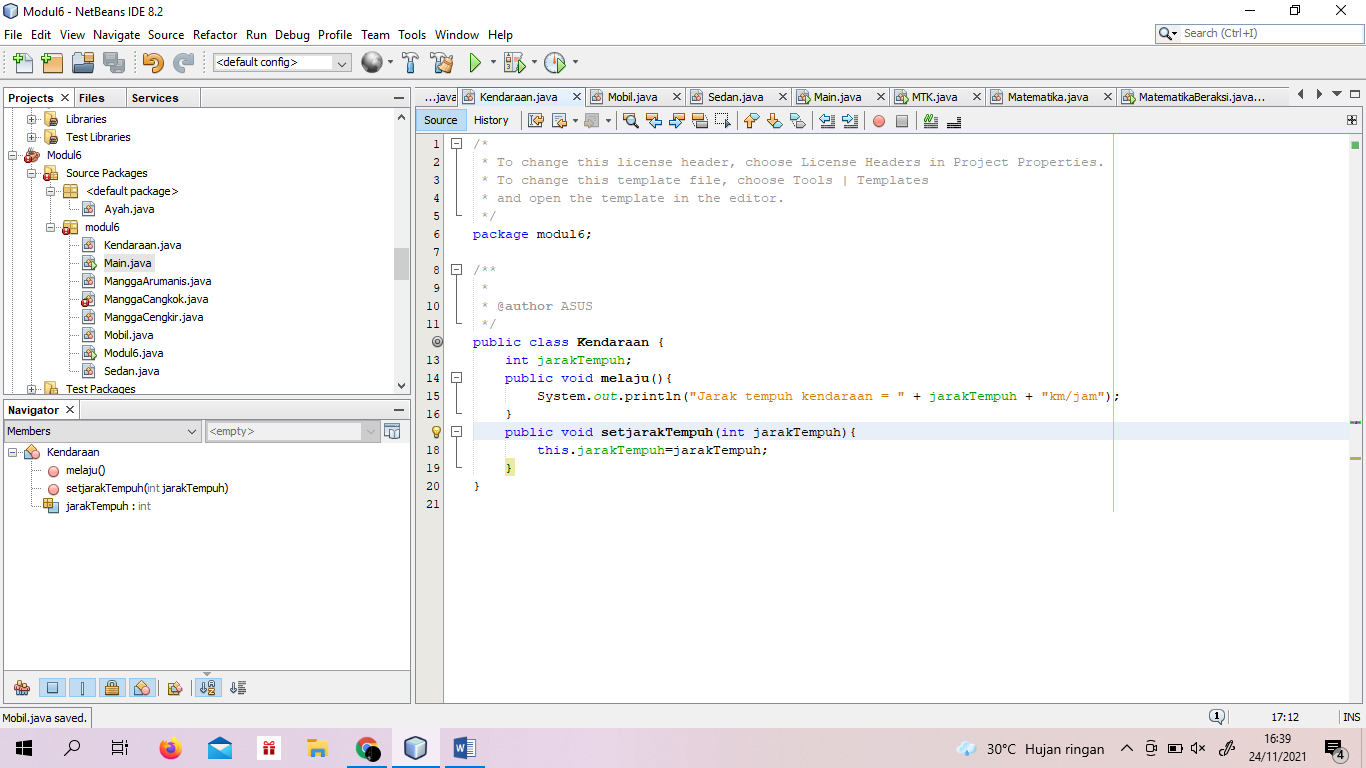


**C. Multilevel inheritance**

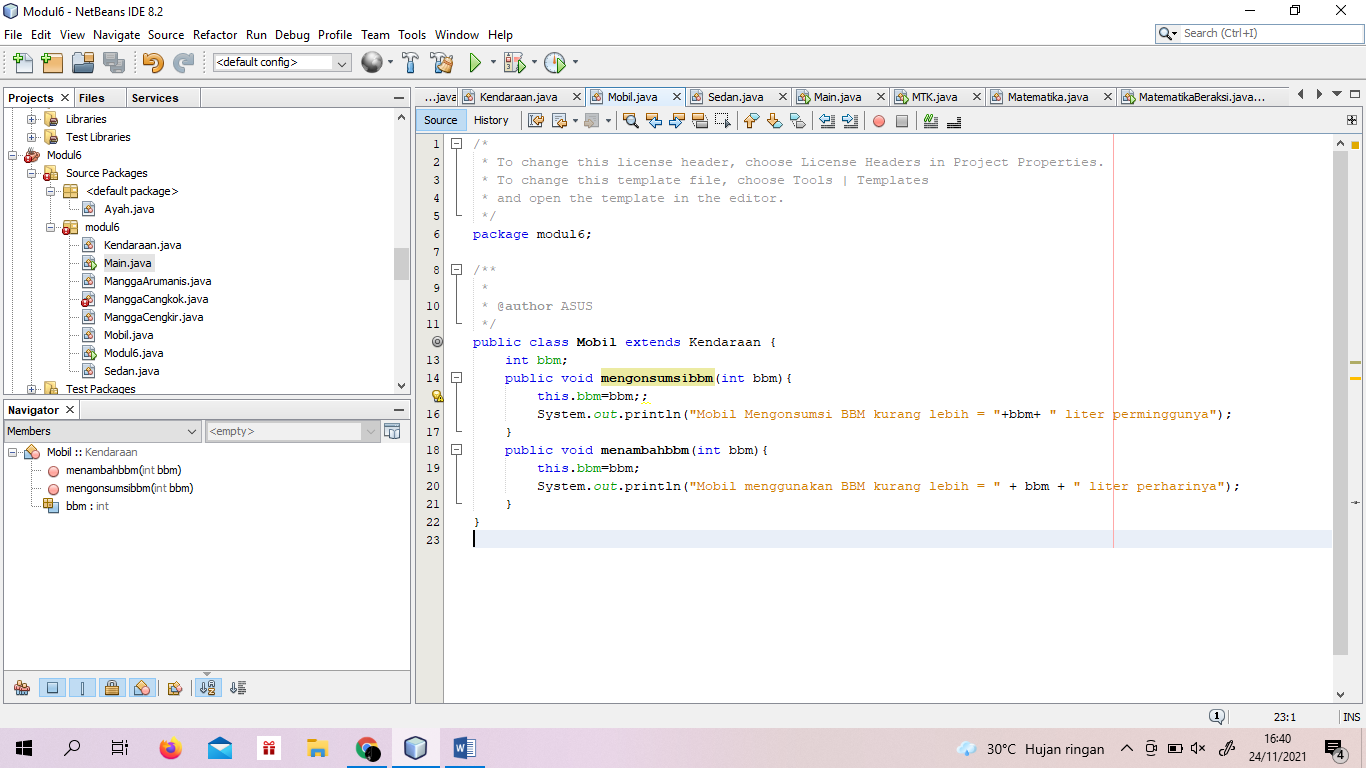
Konsep pewarisan beruntun pada beberapa class. Dalam hal ini dimungkinkan bahwa suatu child class menjadi parent class dari child class yang lain.



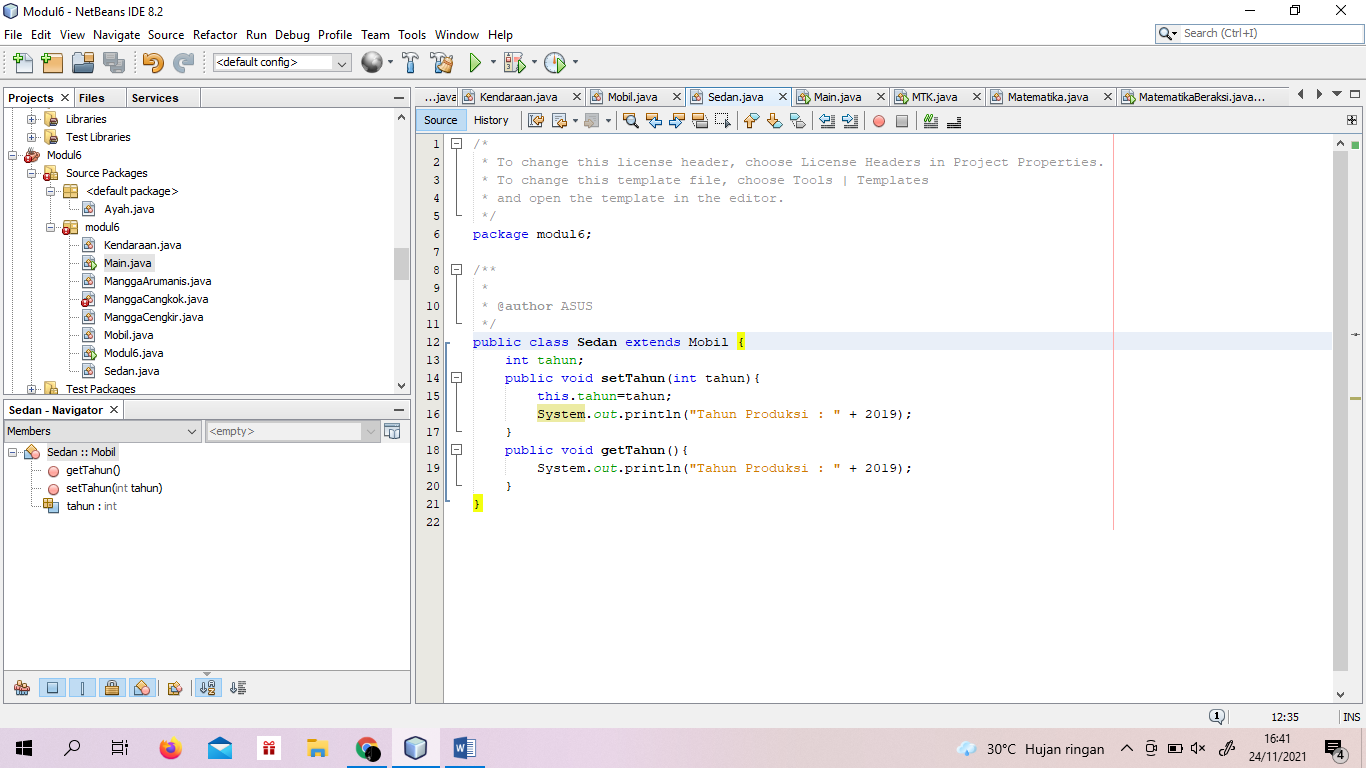
Kendaraan.java



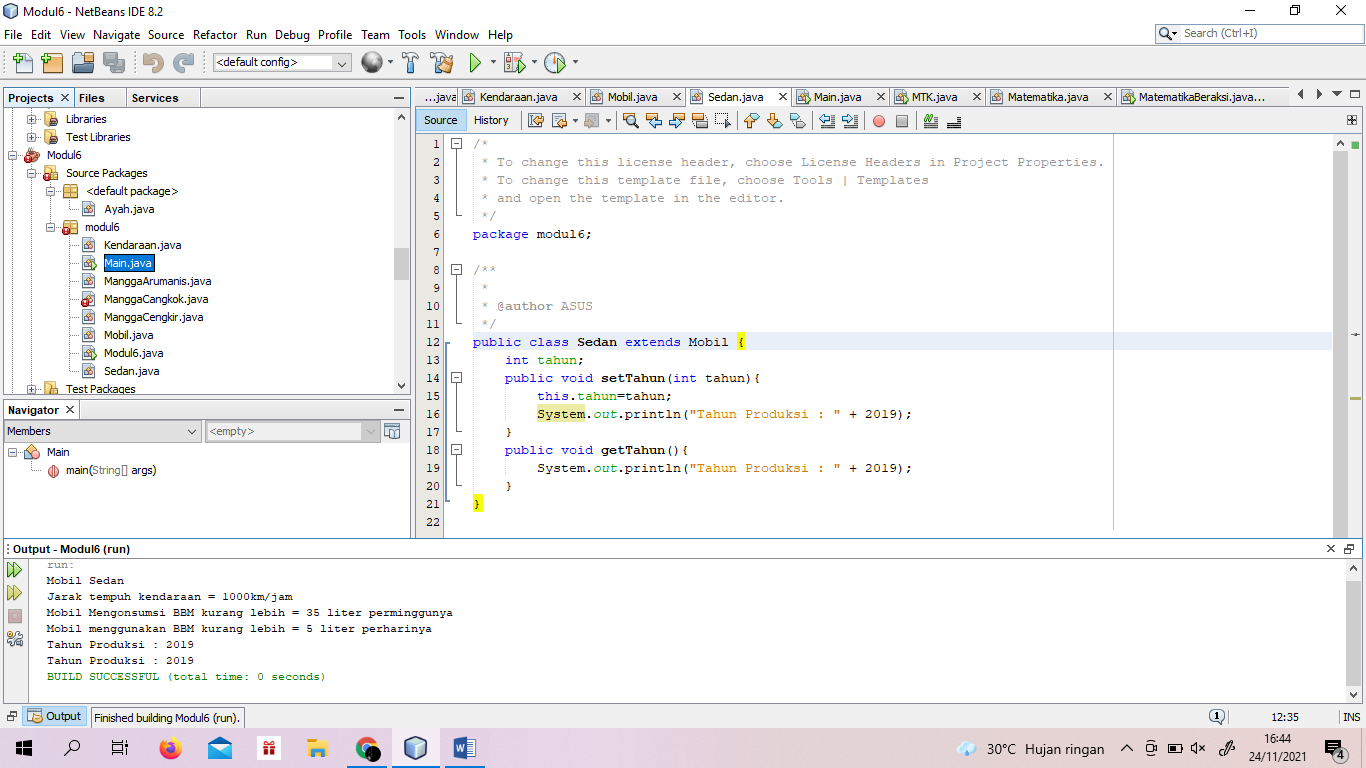
Mobil.java



Sedan.java

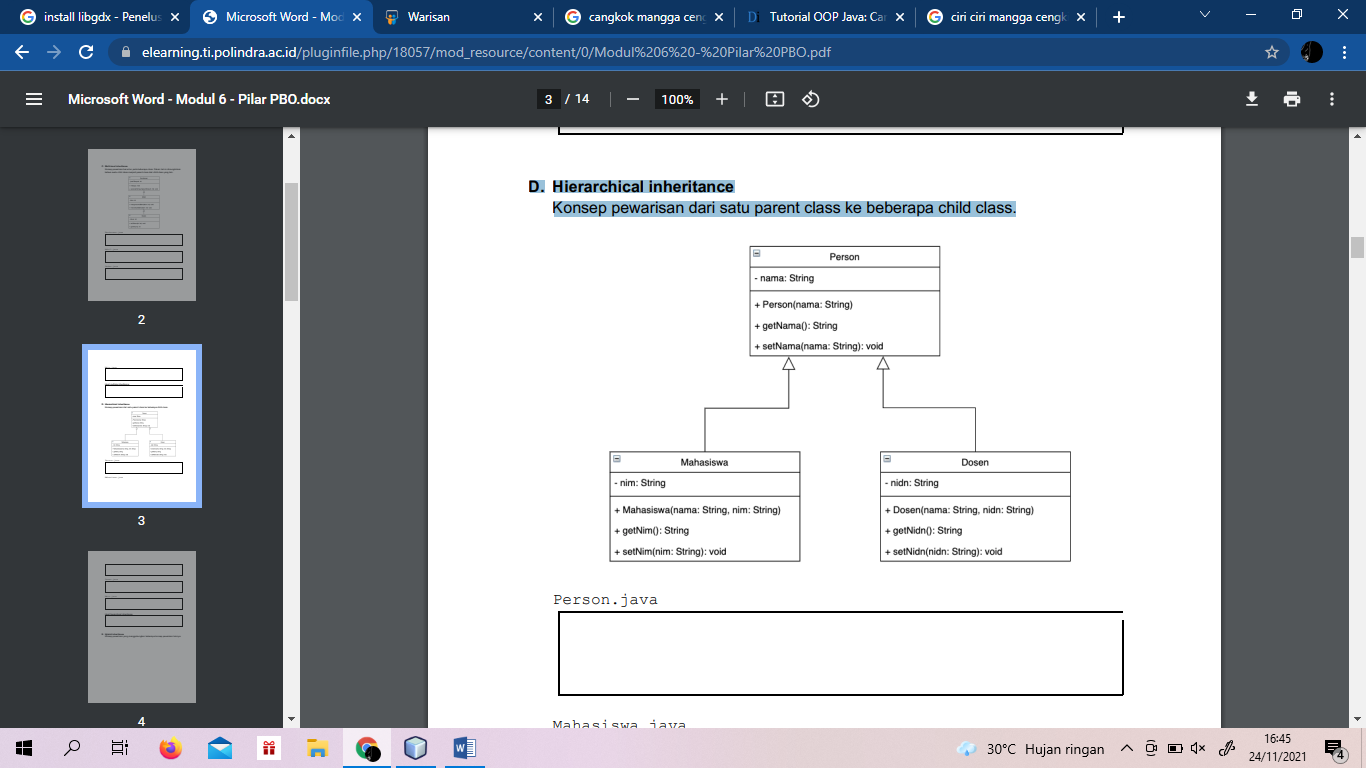


Hasil :

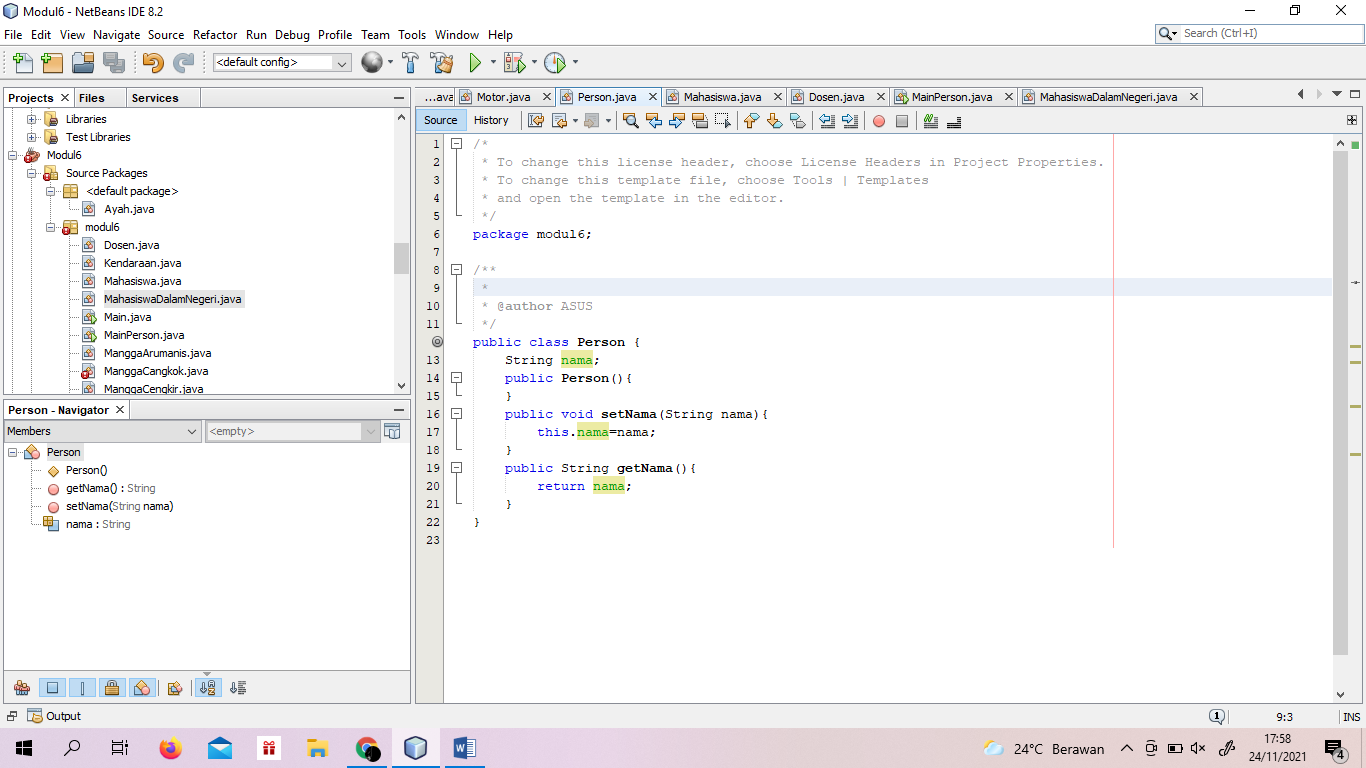


**D. Hierarchical inheritance**

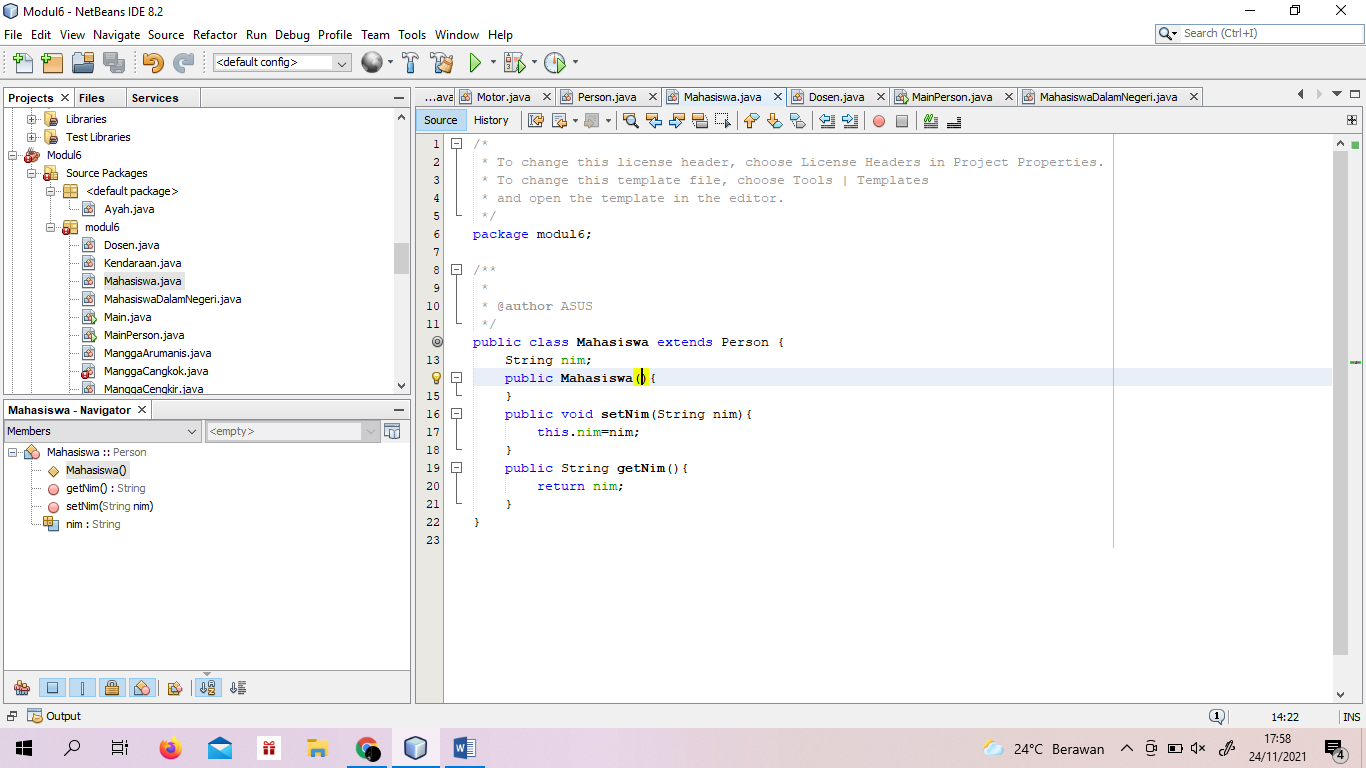
Konsep pewarisan dari satu parent class ke beberapa child class.



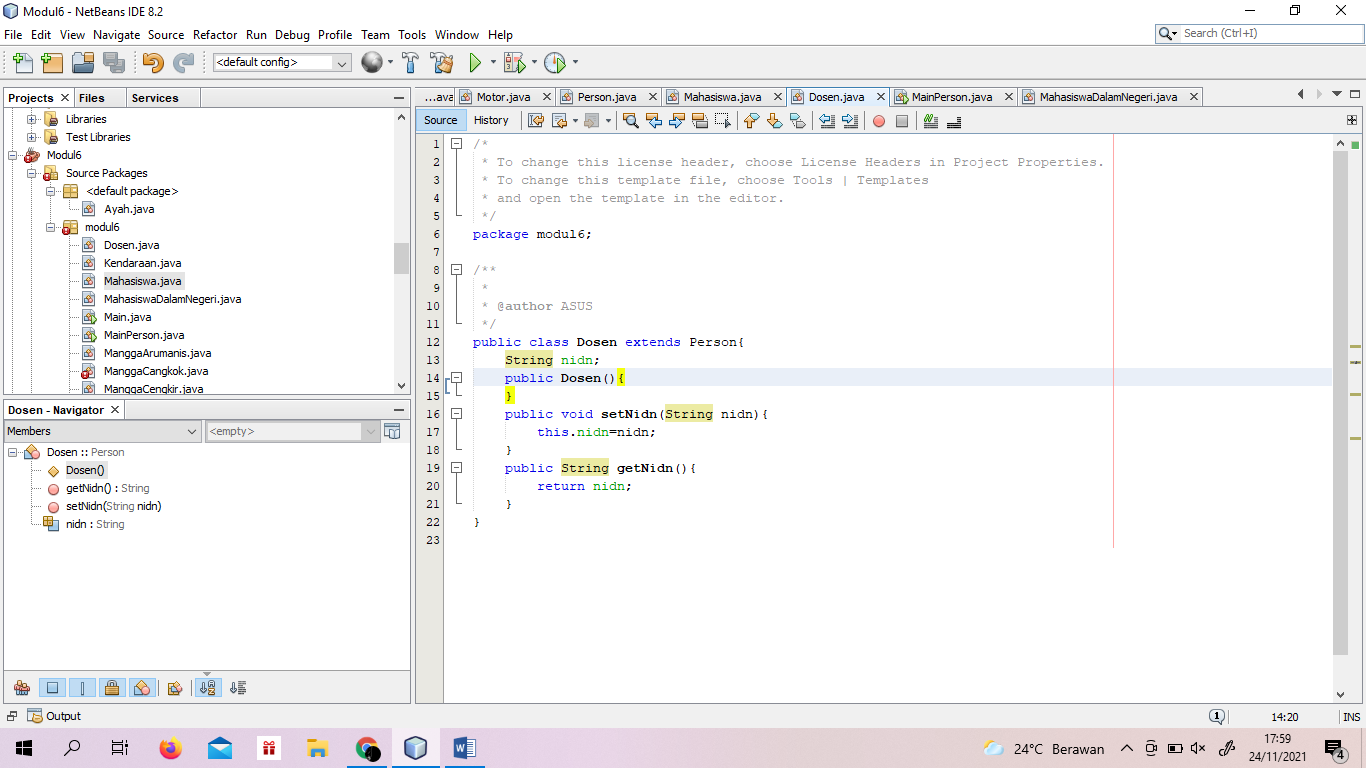
Person.java



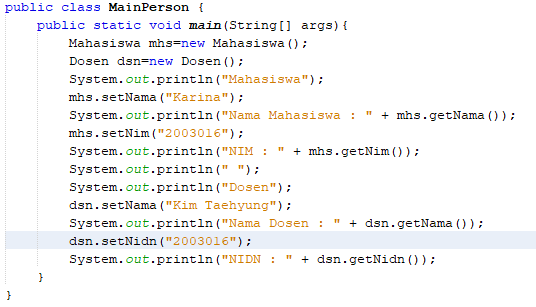
Mahasiswa.java



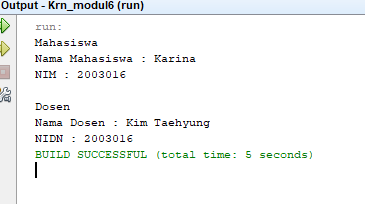
Dosen.java



MainPerson.java

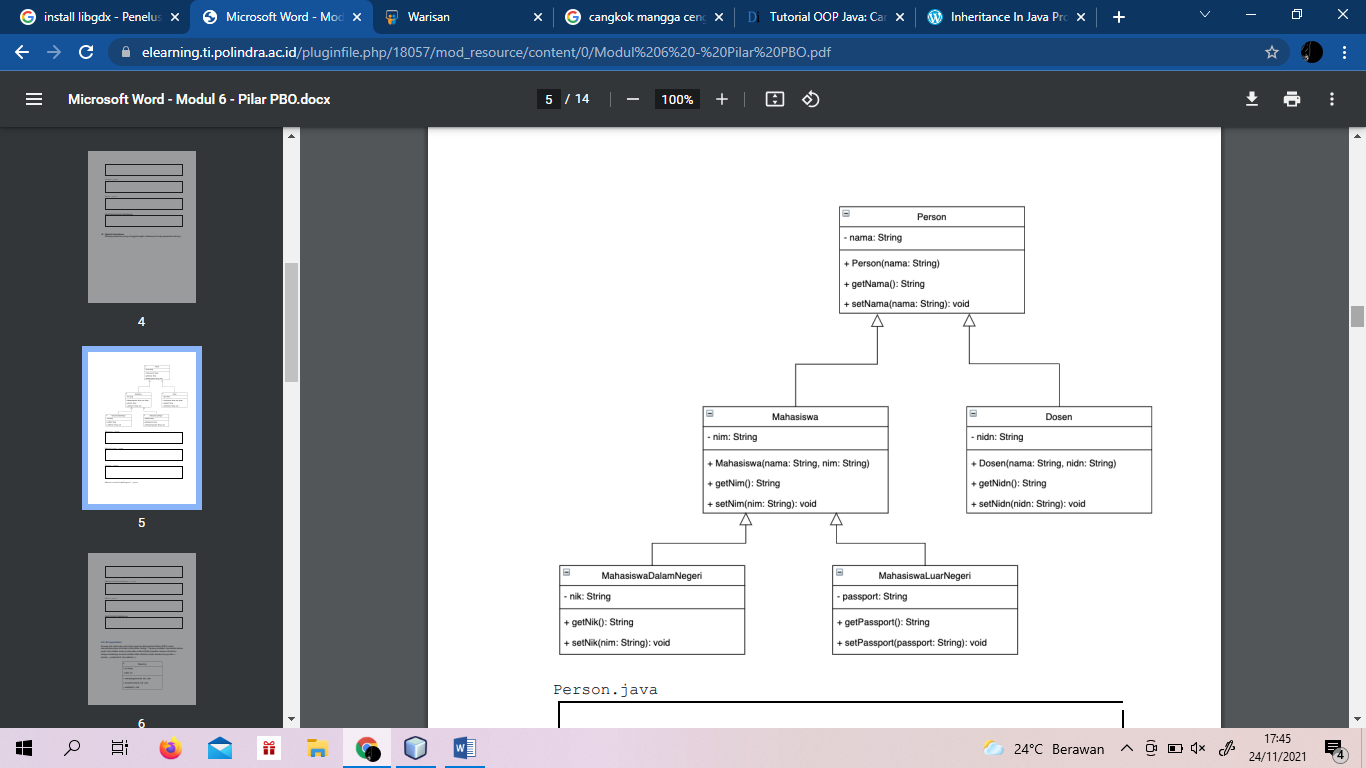


Hasil :

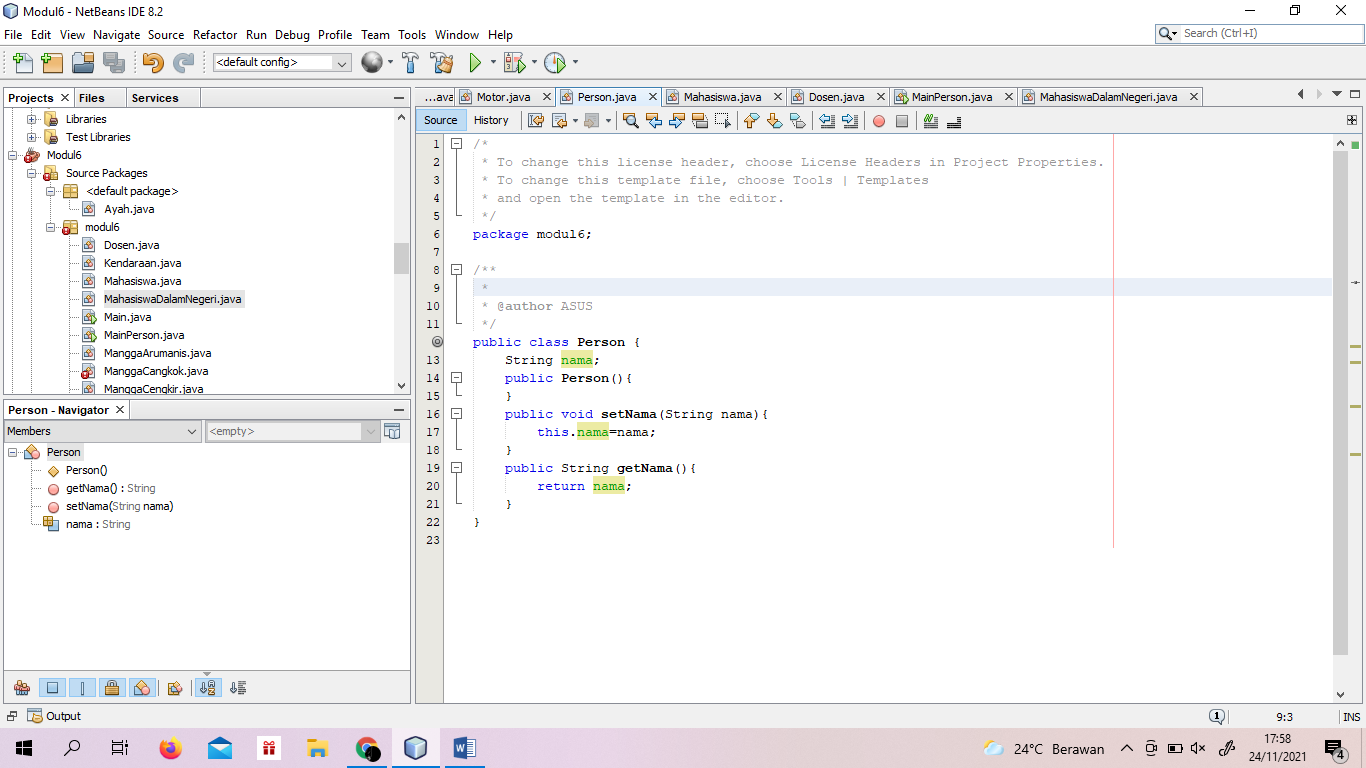


**E. Hybrid inheritance**

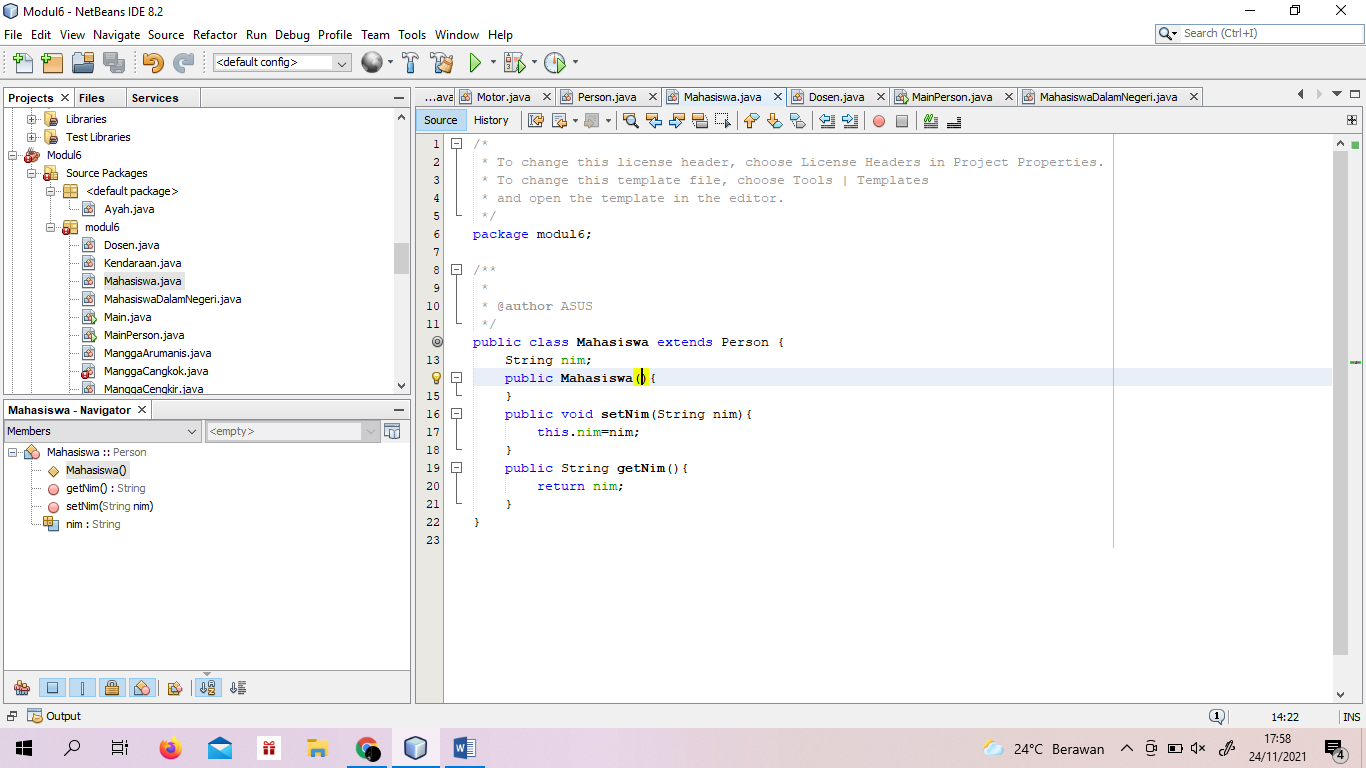
Konsep pewarisan yang menggabungkan beberapa konsep pewarisan lainnya.



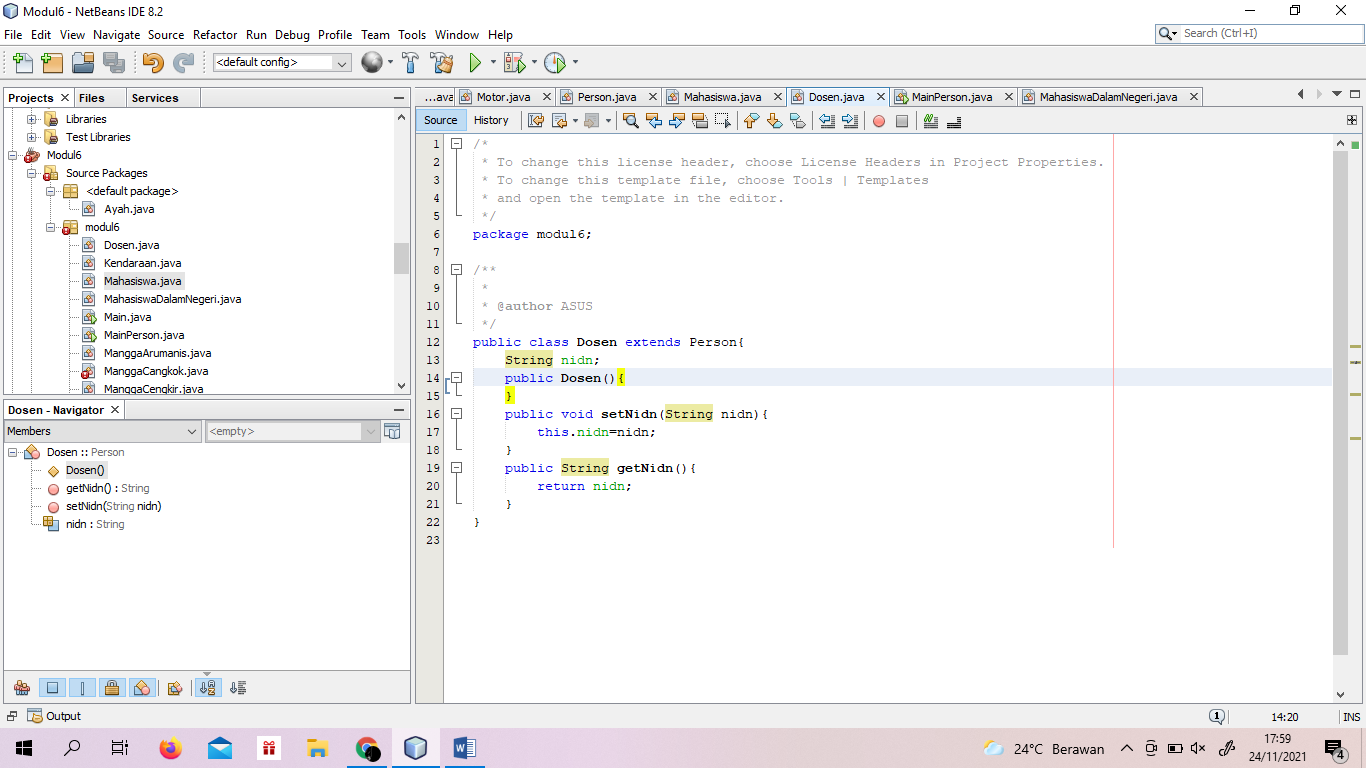
Person.java



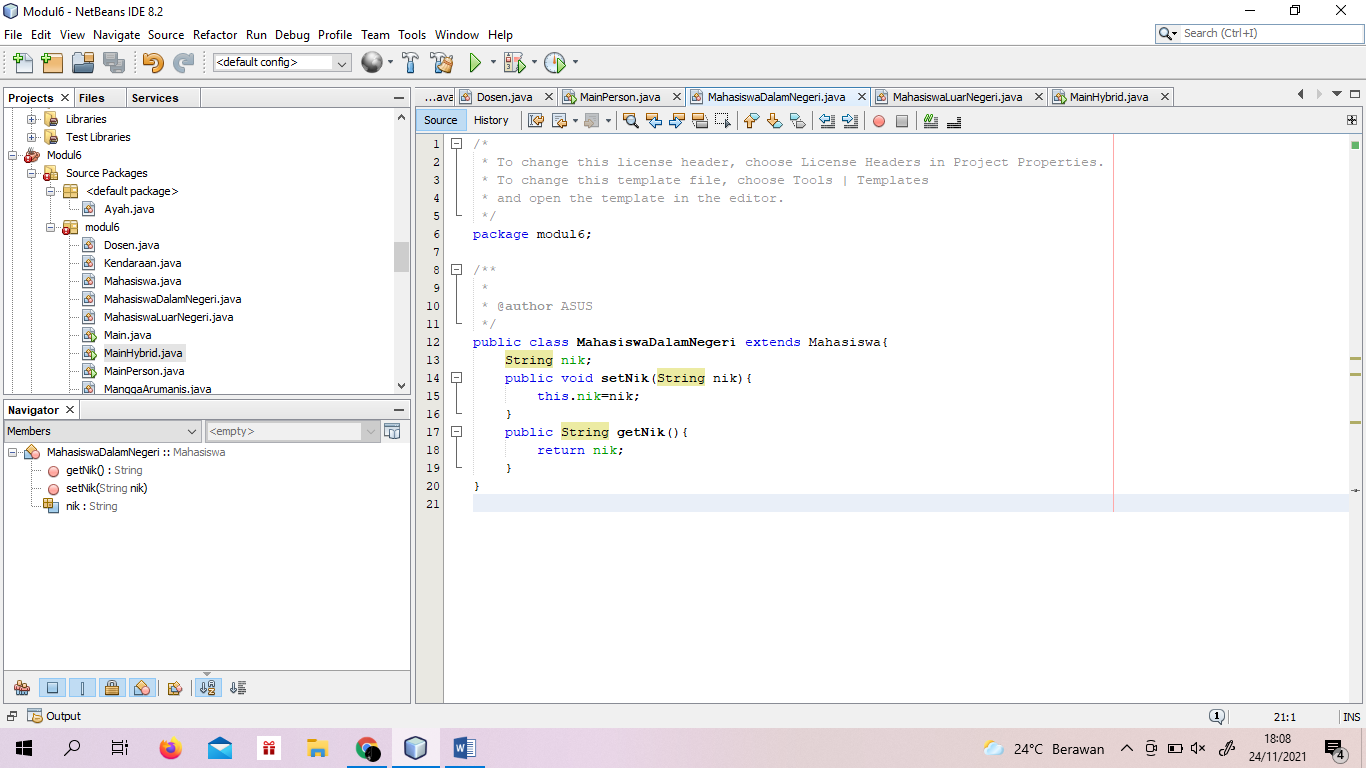
Mahasiswa.java



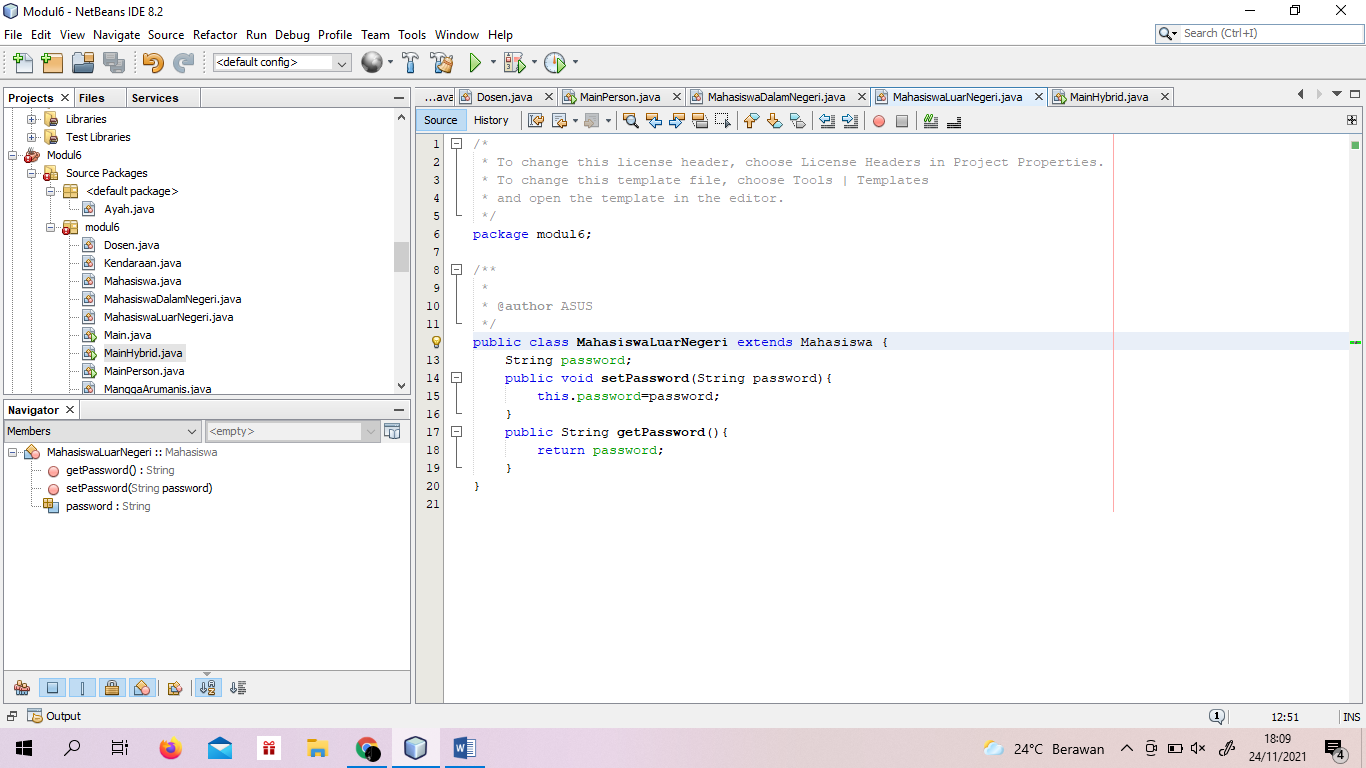
Dosen.java



MahasiswaDalamNegeri.java



MahasiswaLuarNegeri.java



MainHybrid.java



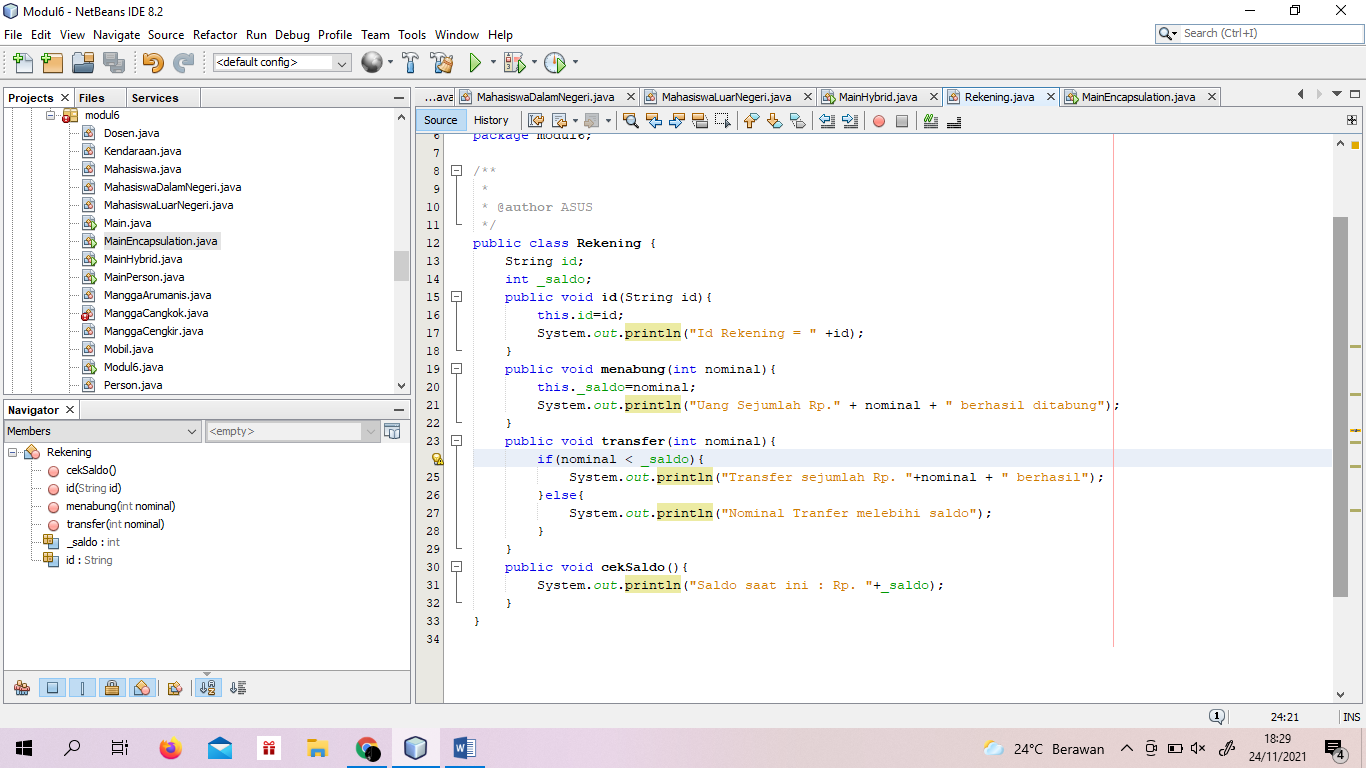
Hasil :



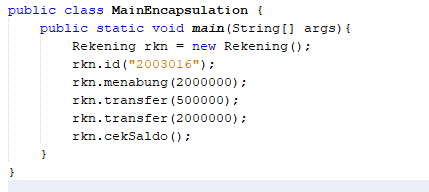
**6.2. Encapsulation**

Konsep dari salah satu pilar Pemrograman Berorientasi Objek (PBO) untuk menyembunyikan informasi (information hiding). Tujuannya adalah membatasi akses suatu nilai melalui suatu proses atau method (baik prosedur maupun function). Adapun beberapa access modifier telah dibahas modul sebelumnya (public +, private -, protected #, dan default ~).

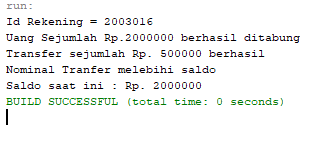
Rekening.java



MainEncapsulation.java



Hasil :

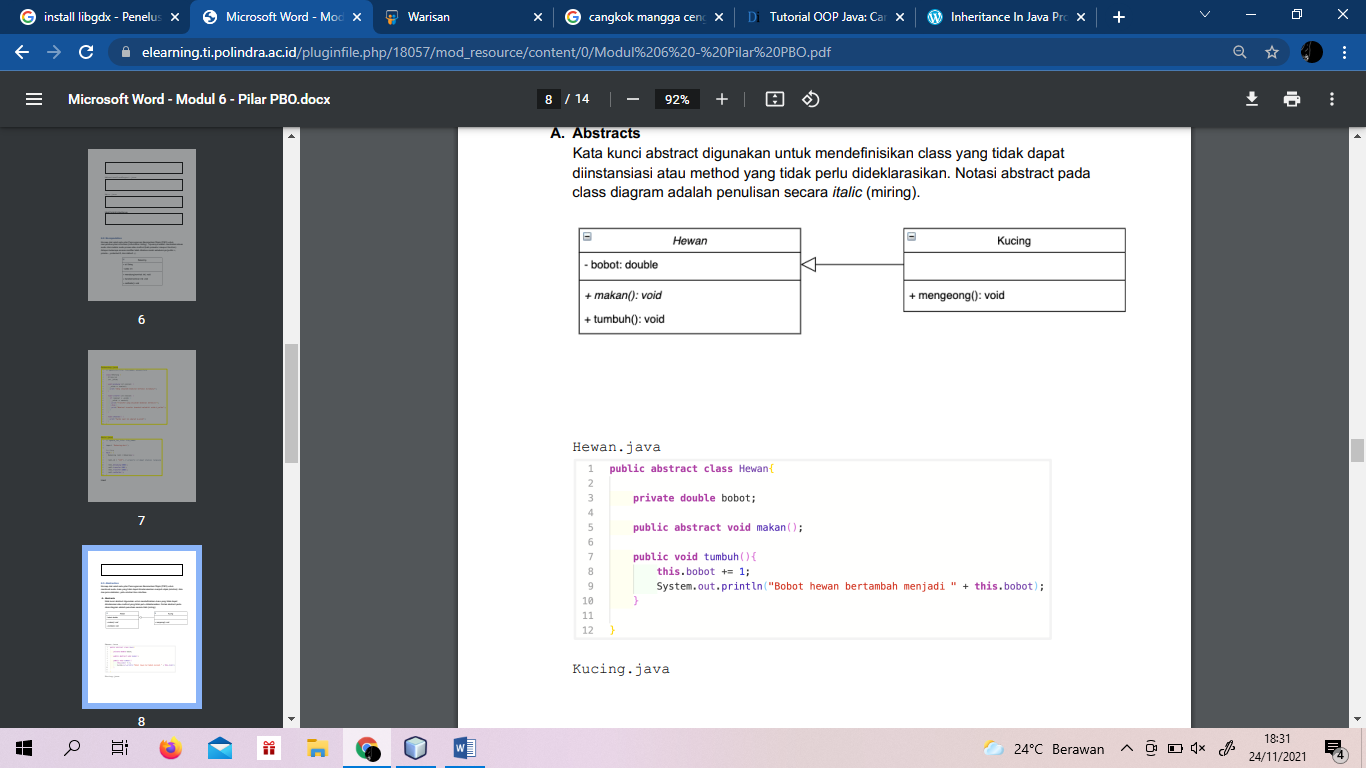


**6.3. Abstraction**

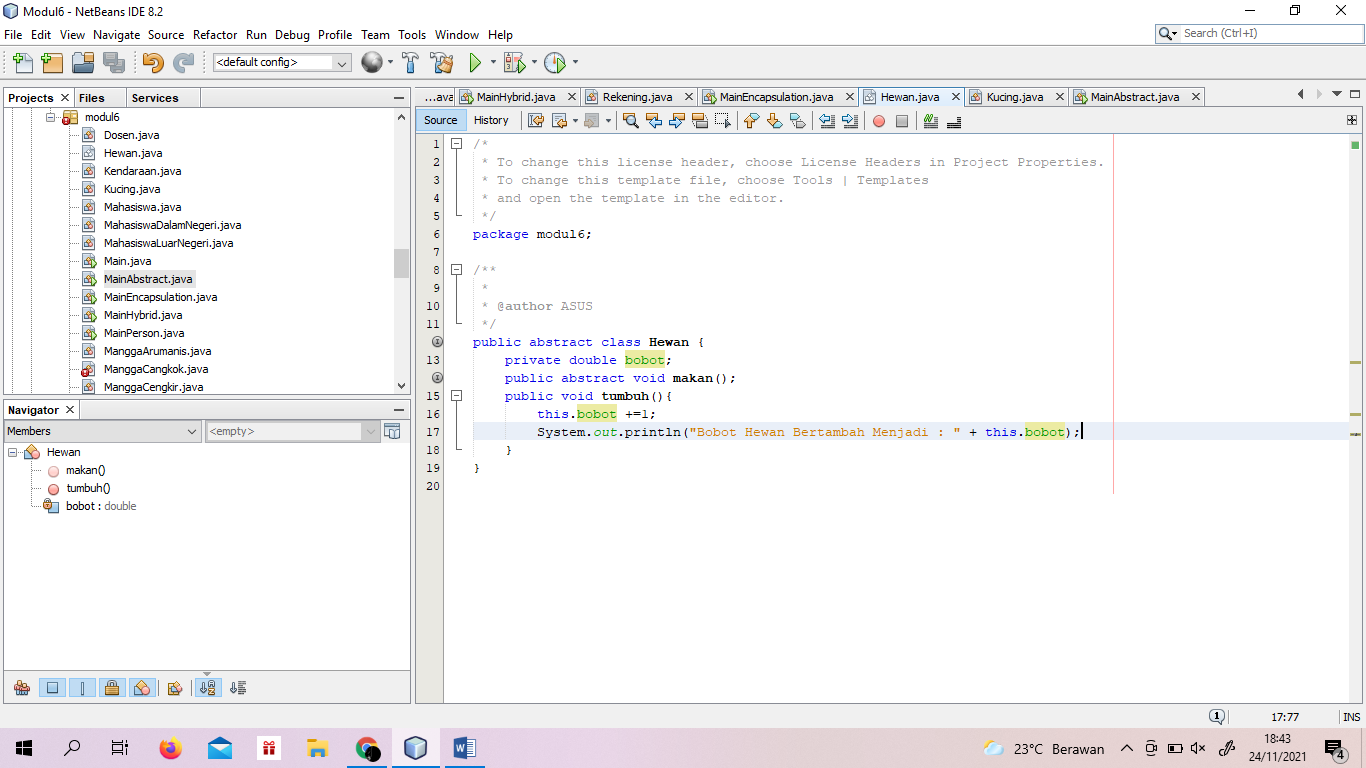
Konsep dari salah satu pilar Pemrograman Berorientasi Objek (PBO) untuk membuat suatu class yang tidak dapat diinstansiasikan menjadi objek (abstrak). Ada dua jenis abstraksi, yaitu abstract dan interface.

**A. Abstracts**

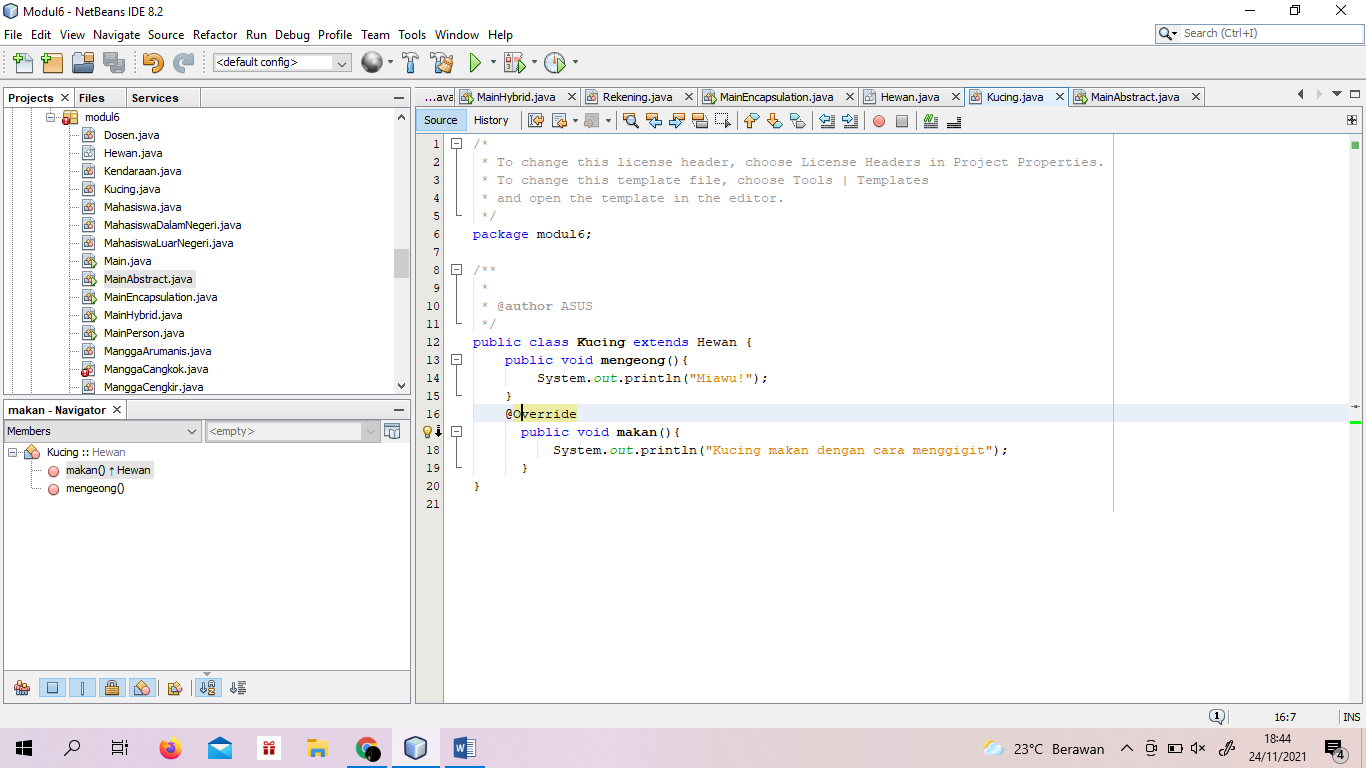
Kata kunci abstract digunakan untuk mendefinisikan class yang tidak dapat diinstansiasi atau method yang tidak perlu dideklarasikan. Notasi abstract pada class diagram adalah penulisan secara italic (miring).



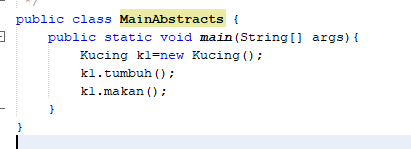
Hewan.java



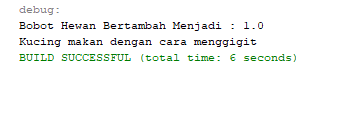
Kucing.java



MainAbstracts.java

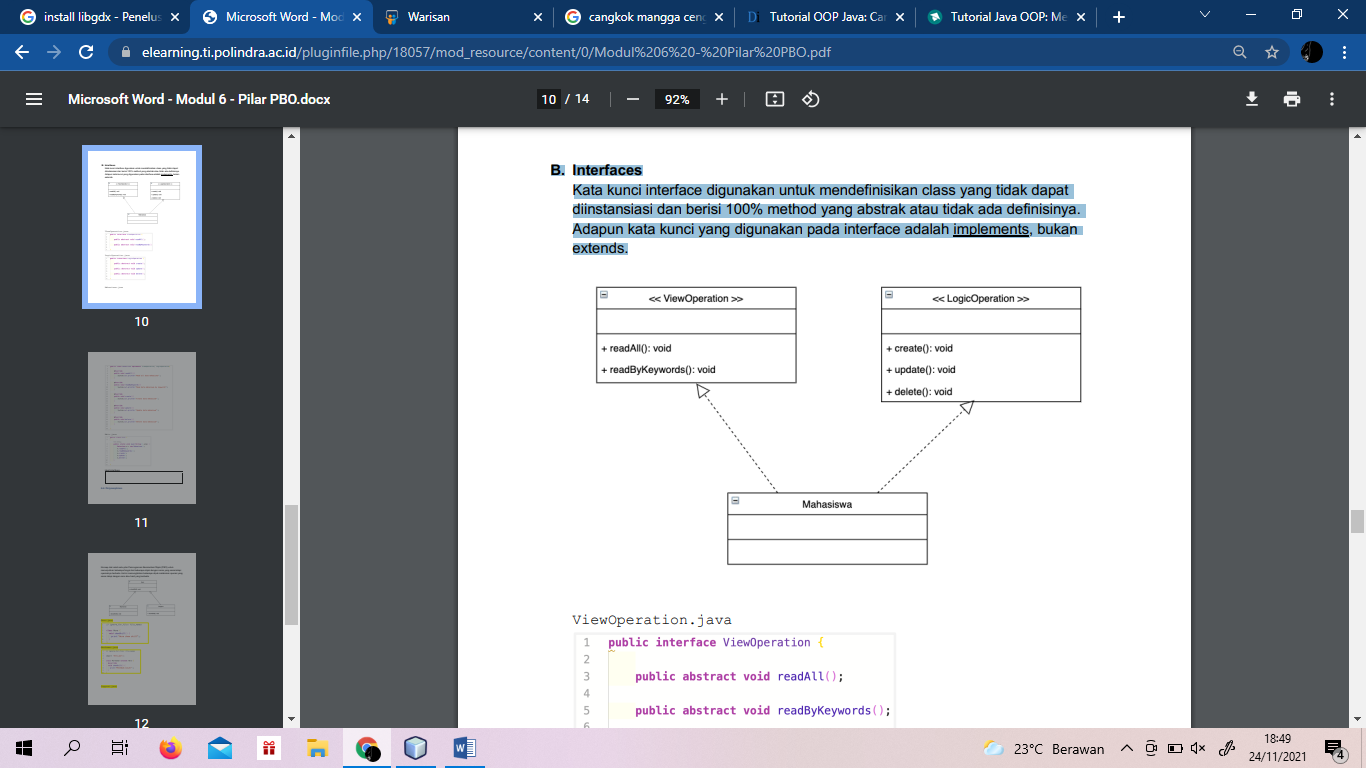


Hasil :

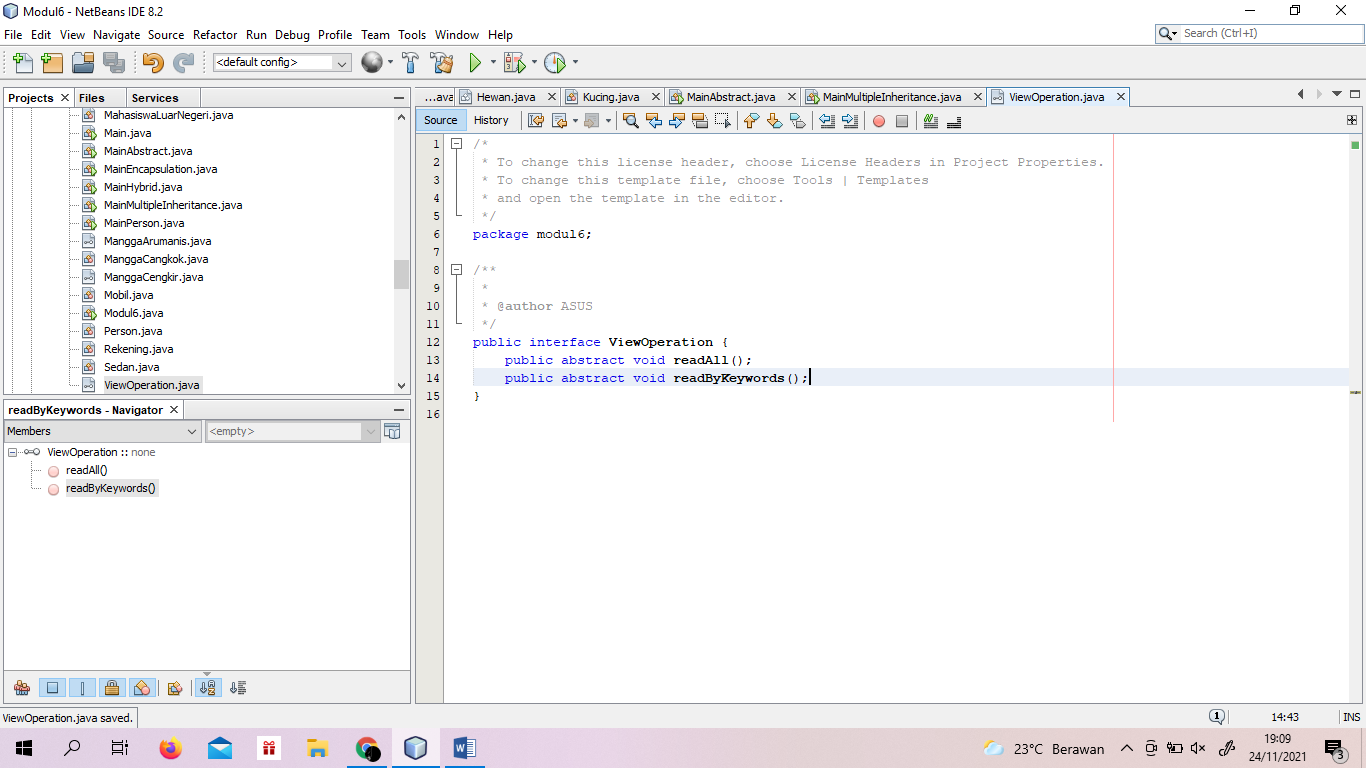


**B. Interfaces**

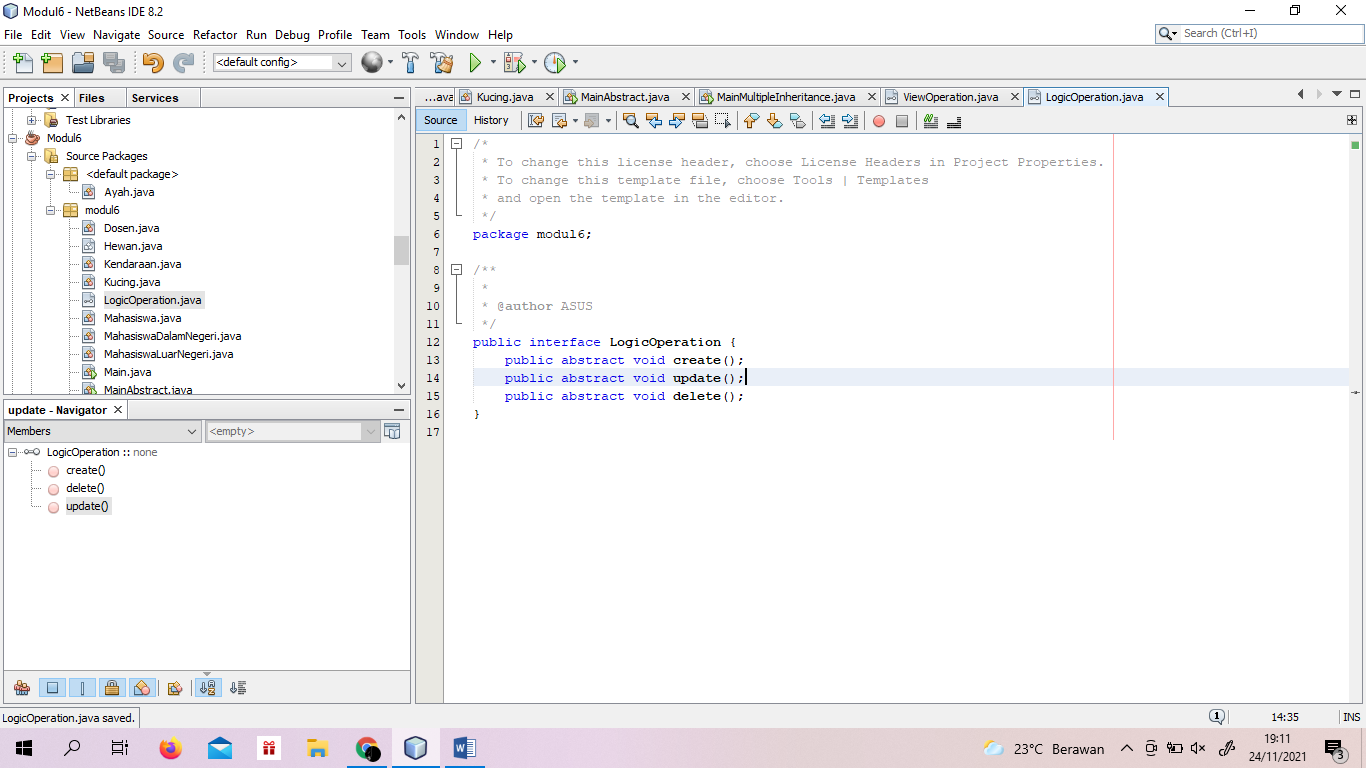
Kata kunci interface digunakan untuk mendefinisikan class yang tidak dapat diinstansiasi dan berisi 100% method yang abstrak atau tidak ada definisinya. Adapun kata kunci yang digunakan pada interface adalah implements, bukan extends.



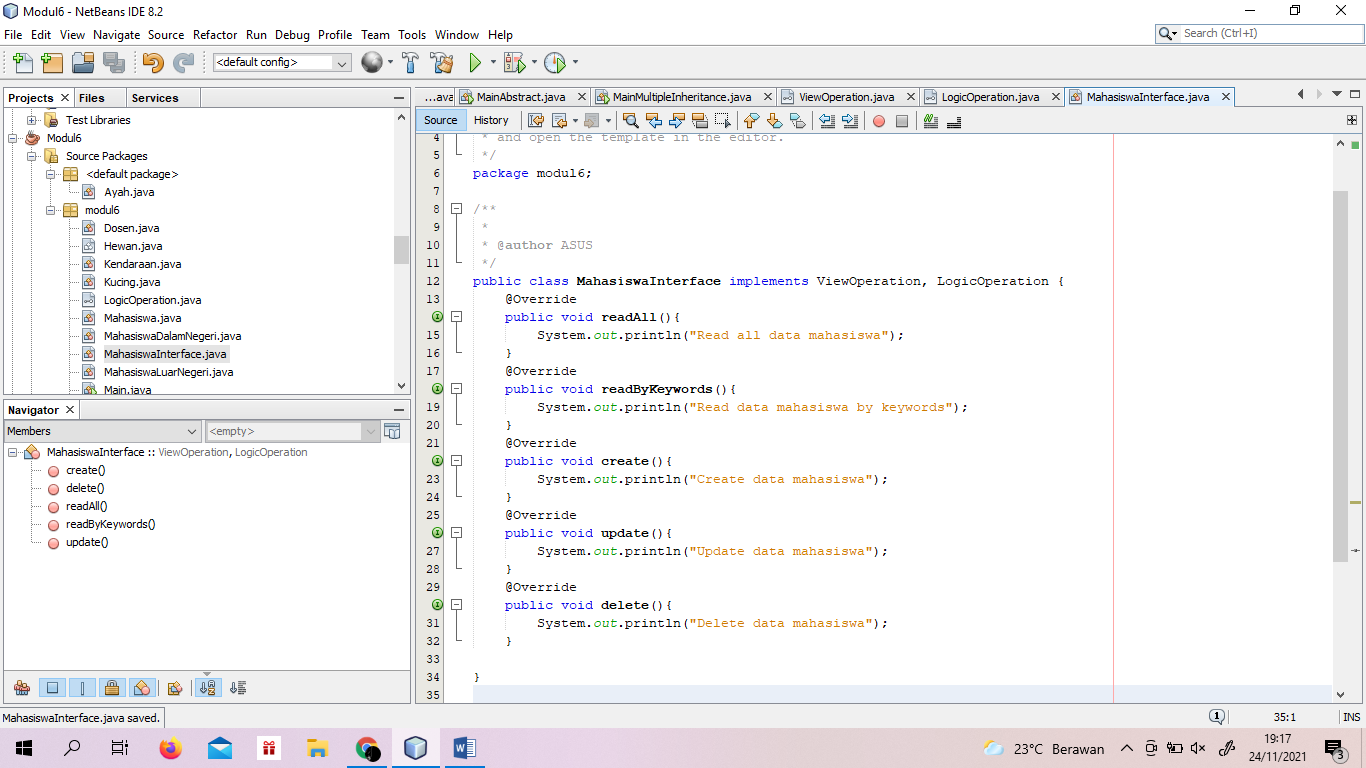
ViewOperation.java



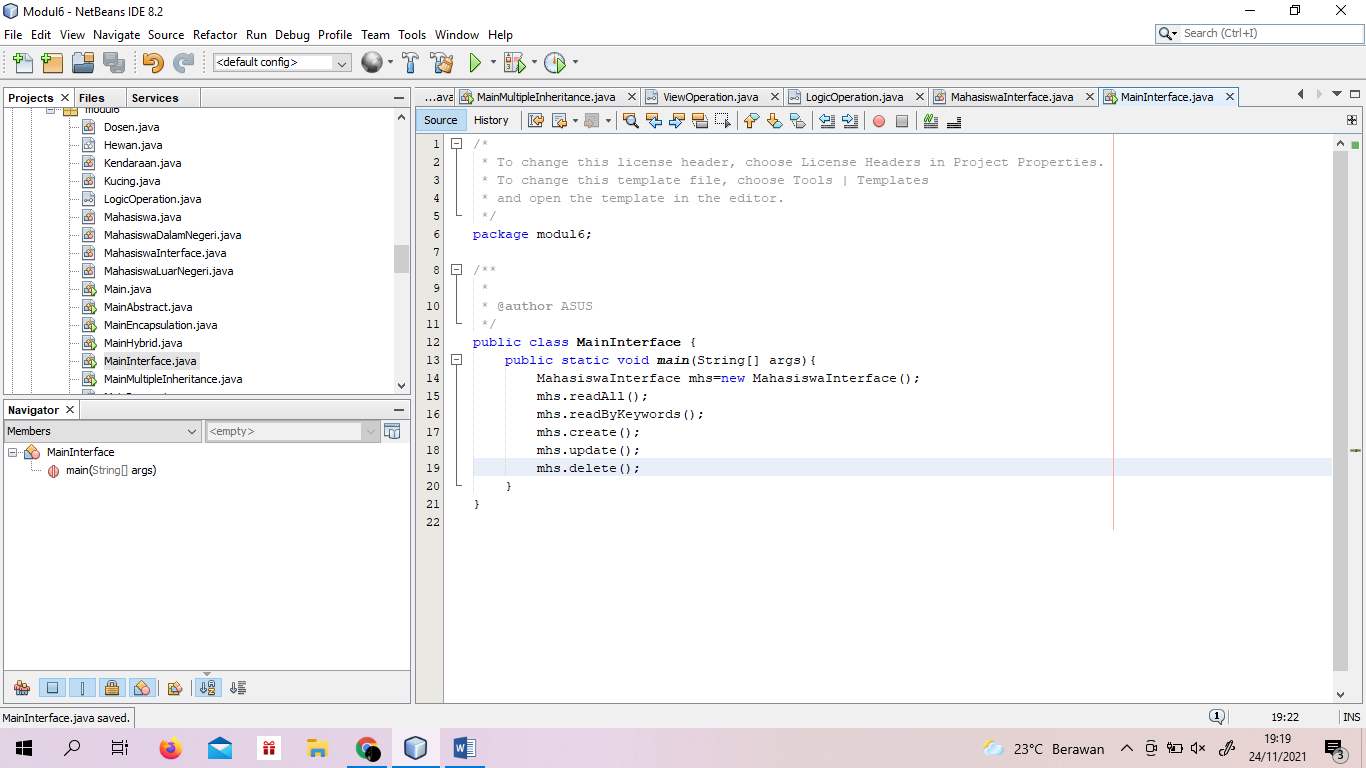
LogicOperation.java



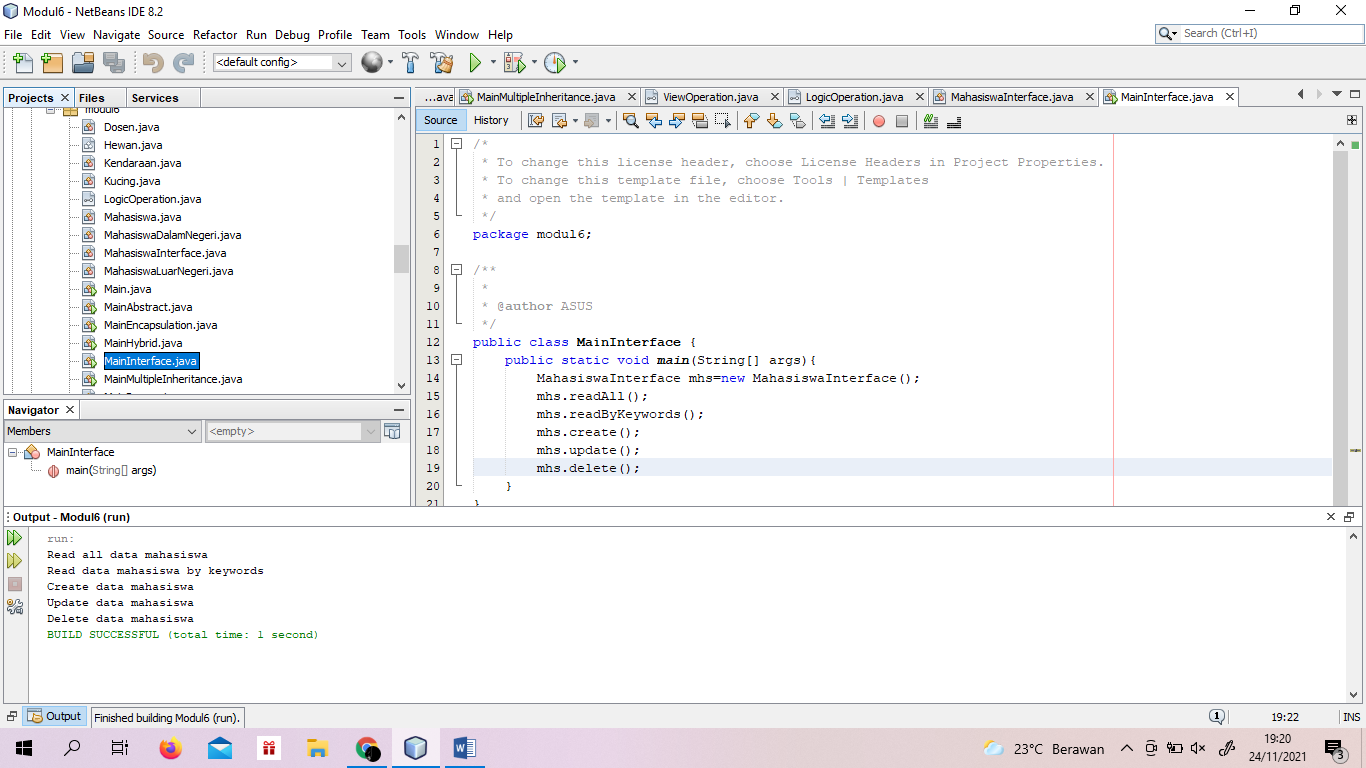
MahasiswaInterface.java



Main.java



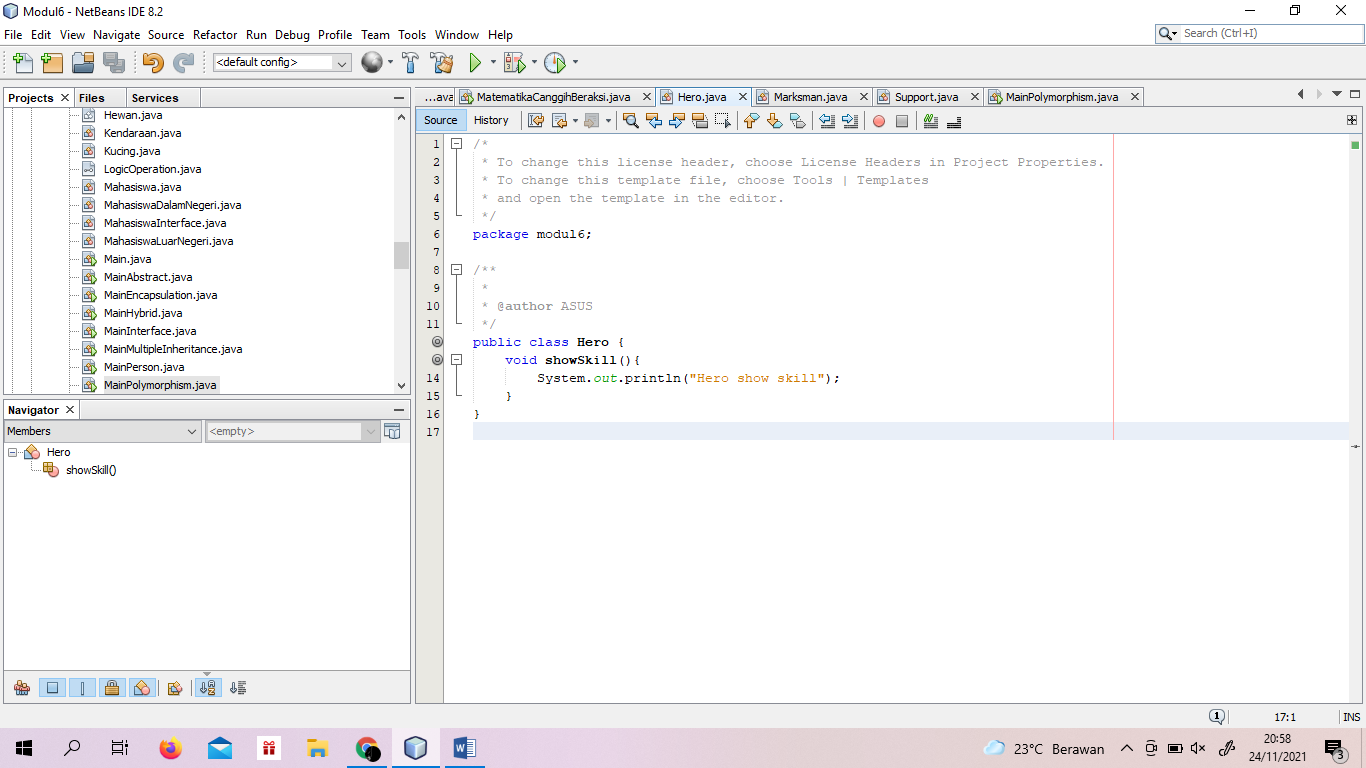
Hasil :



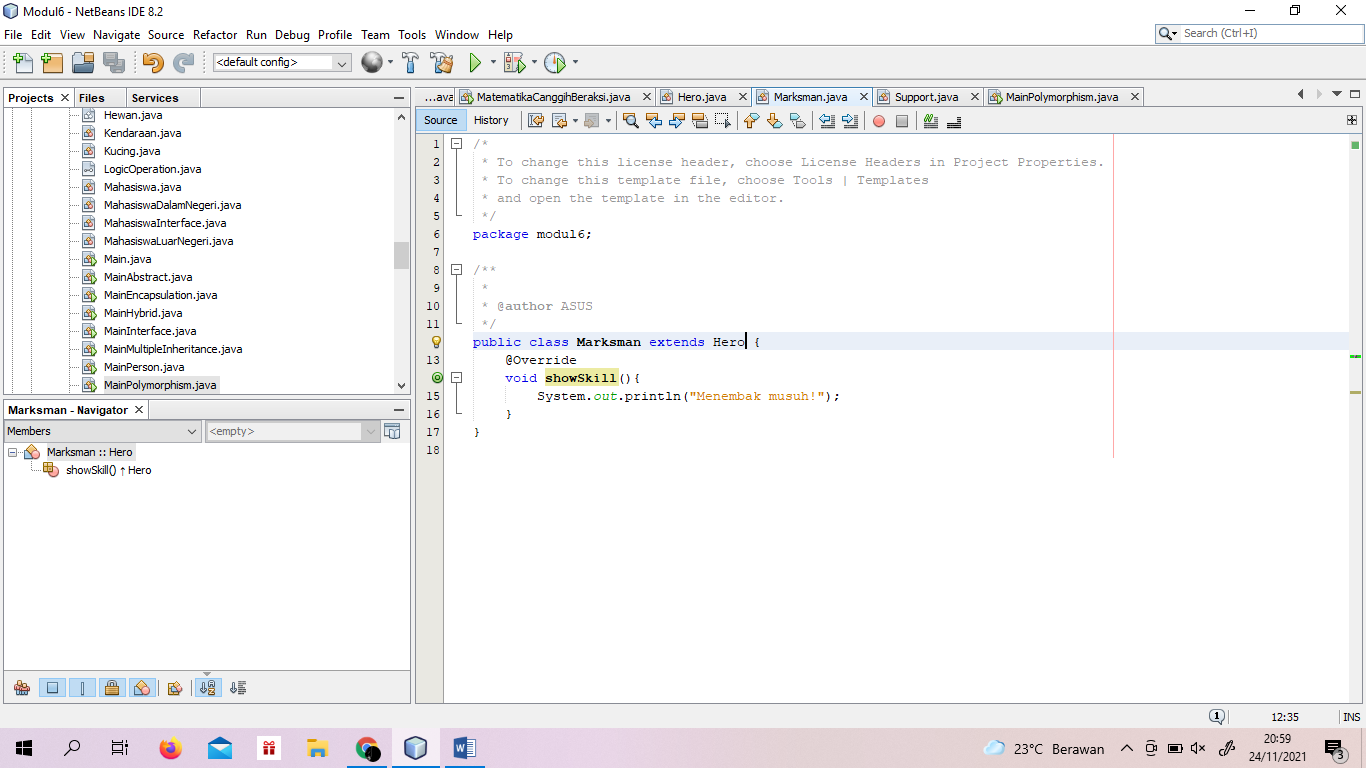
**6.4. Polymorphism**

Konsep dari salah satu pilar Pemrograman Berorientasi Objek (PBO) untuk menunjukkan beberapa fungsi dari beberapa objek dengan nama yang sama tetapi operasinya berbeda. Hal ini memungkinkan beberapa objek melakukan operasi yang sama tetapi dengan cara atau hasil yang berbeda.

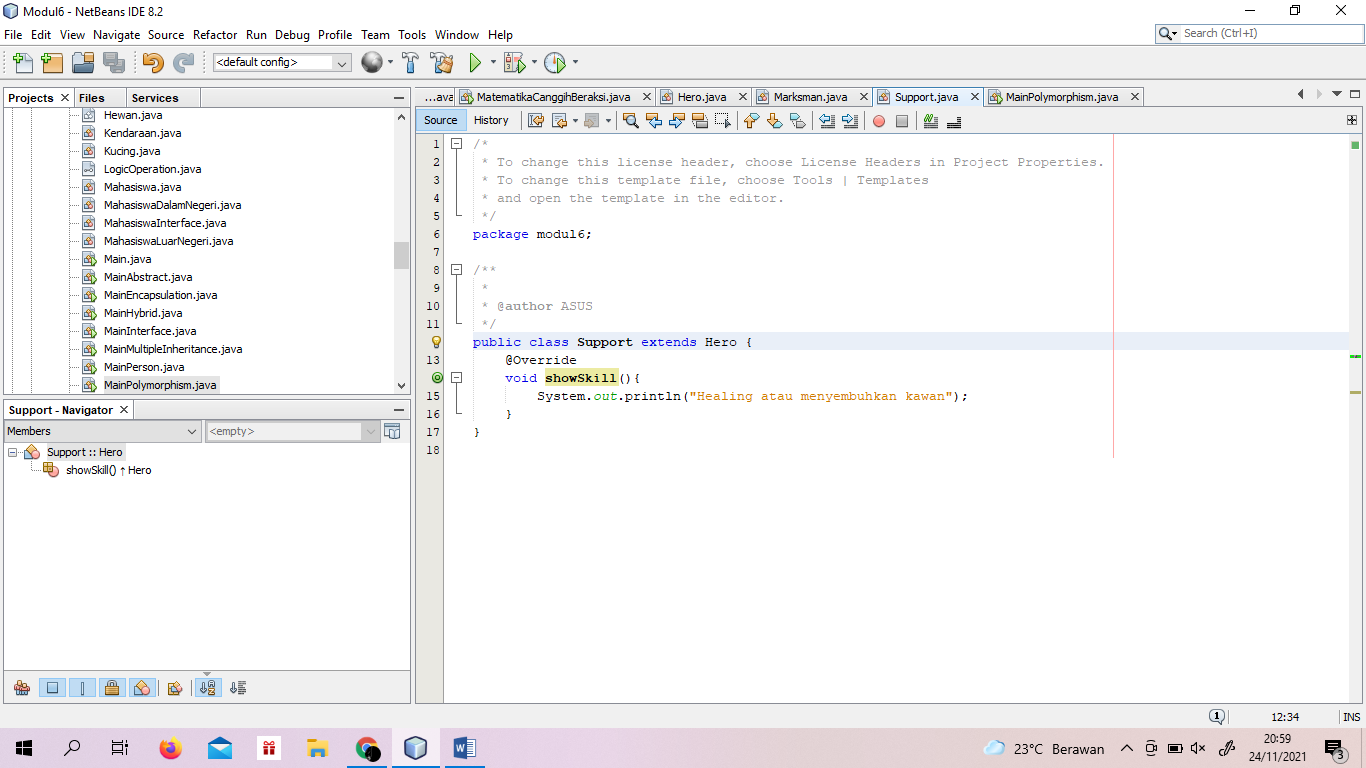
Hero.java



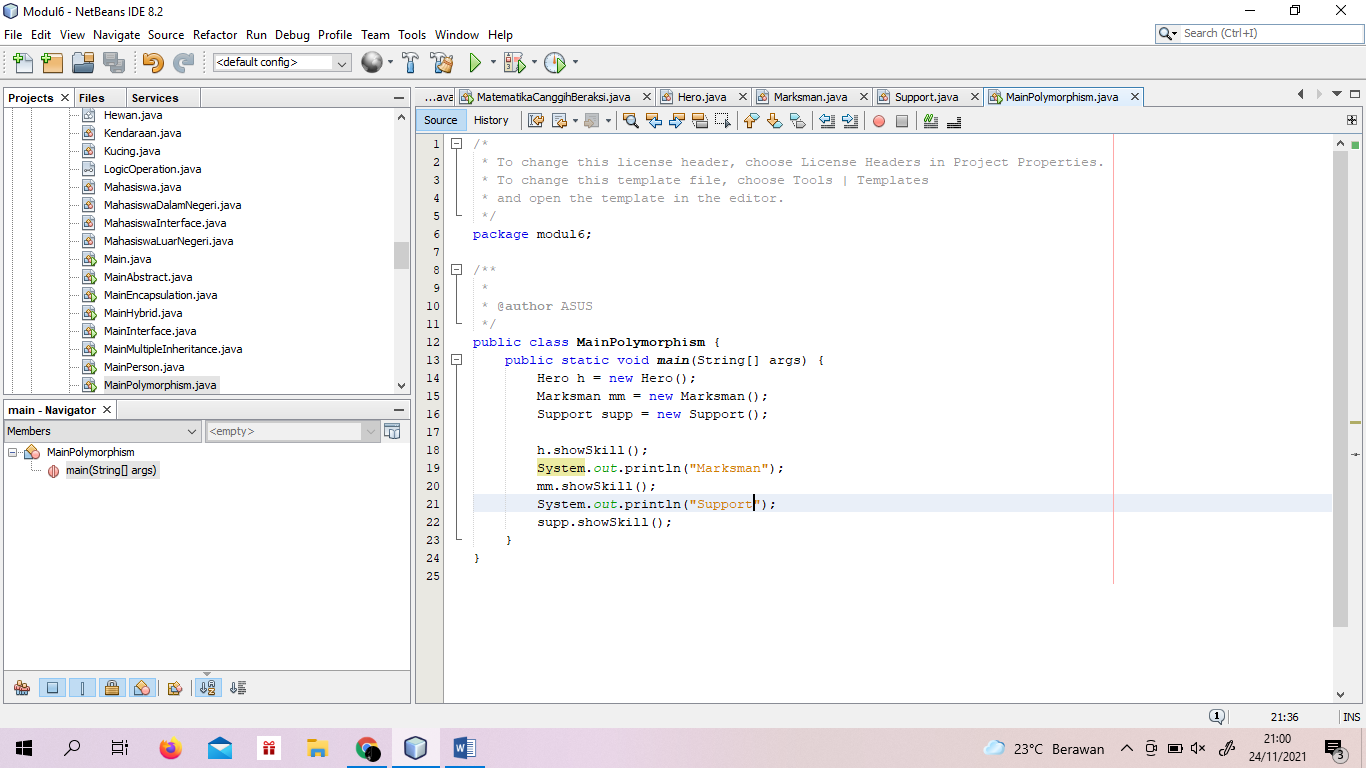
Marksman.java



Support.java



MainPolimorphism.java



Hasil :

