

LAPORAN

# Polymorphism



**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

Oleh :

**Naufal Firmansyah\_5210411166**

## TEORI

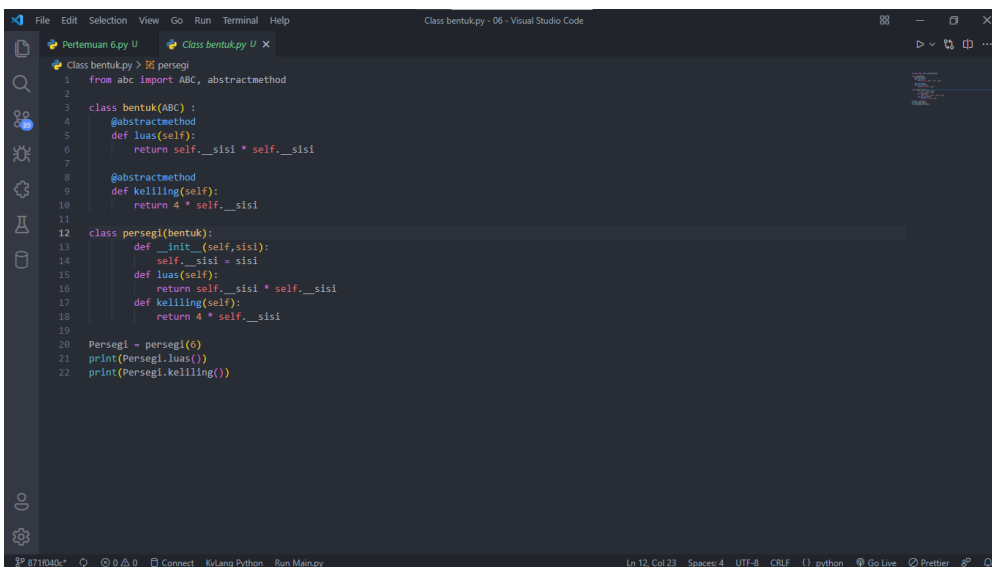
Polymorphism adalah suatu objek yang dapat memiliki berbagai bentuk objek. Overriding merupakan fitur yang memungkinkan kita untuk mengimplementasikan ulang sebuah method atau fungsi pada sebuah class child. Overloading adalah sebuah Teknik yang digunakan untuk mengatur perilaku dari suatu fungsi. Dan abstraction adalah suatu method yang digunakan untuk menyembunyikan detail yang tidak terlalu penting dari user.

## TUJUAN

Polymorphism, overriding, overloading, dan abstraction digunakan agar program terlihat lebih rapi, mencegah adanya penulisan berulang dari sebuah source code, dan digunakan agar developer lebih bisa efisien dalam mengerjakan suatu proyek.

## Program

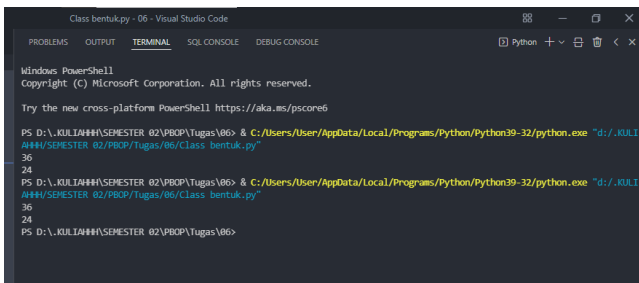
### 1.



```
1 from abc import ABC, abstractmethod
2
3 class bentuk(ABC):
4     @abstractmethod
5     def luas(self):
6         return self.__sisi * self.__sisi
7
8     @abstractmethod
9     def keliling(self):
10        return 4 * self.__sisi
11
12 class persegi(bentuk):
13     def __init__(self, sisi):
14         self.__sisi = sisi
15     def luas(self):
16         return self.__sisi * self.__sisi
17     def keliling(self):
18         return 4 * self.__sisi
19
20 Persegi = persegi(6)
21 print(Persegi.luas())
22 print(Persegi.keliling())
```

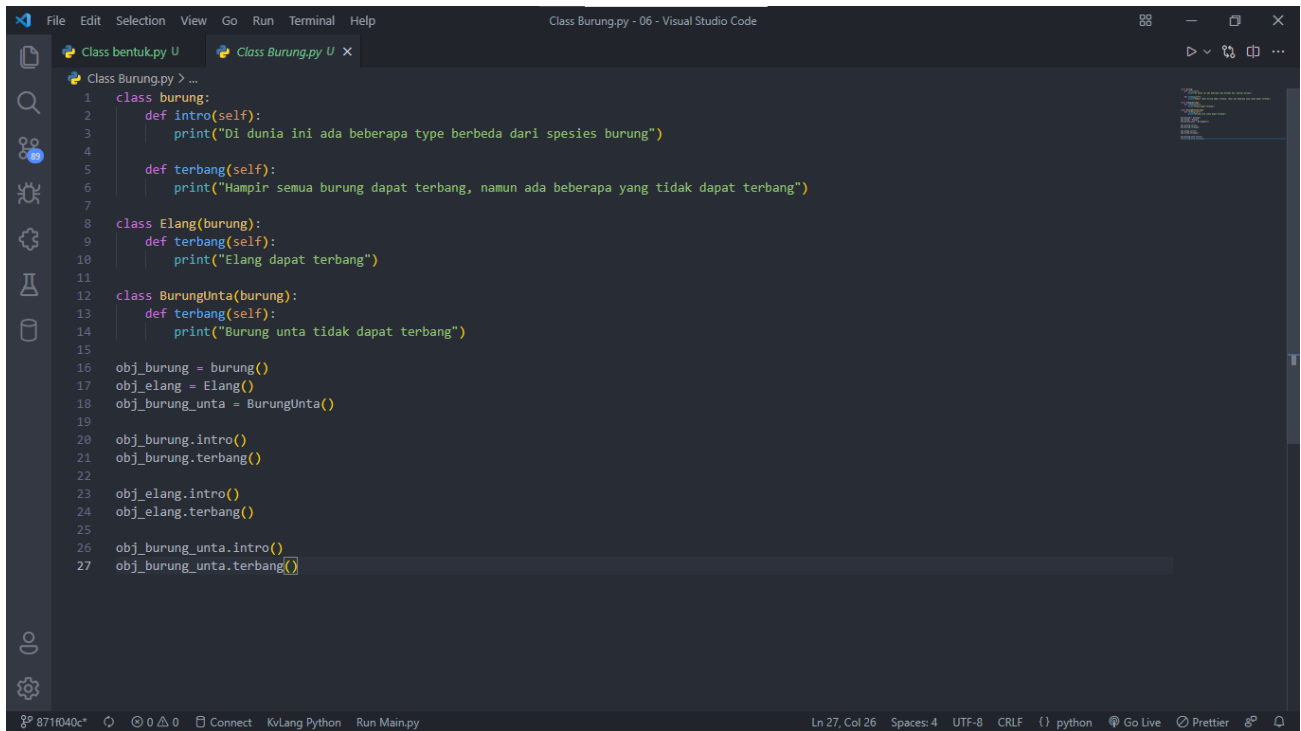
## Penjelasan :

Pertama-tama kita mengimport storage yang dimiliki oleh python. Kemudian class bentuk mengambil data dari storage ABC dan kemudian menampung method abstract. Begitupula dengan class persegi yang mengambil data dari class bentuk yang kemudian semuanya di hitung dan dicetak lalu ditampilkan pada layar user. Output :



```
PS D:\KULIAH\SEMESTER 02\PROPA\Tugas\06> & C:/Users/User/AppData/Local/Programs/Python/Python39-32/python.exe "d:/KULIAH/SEMESTER 02/PROPA/Tugas/06/Class_bentuk.py"
36
24
PS D:\KULIAH\SEMESTER 02\PROPA\Tugas\06> & C:/Users/User/AppData/Local/Programs/Python/Python39-32/python.exe "d:/KULIAH/SEMESTER 02/PROPA/Tugas/06/Class_bentuk.py"
36
24
PS D:\KULIAH\SEMESTER 02\PROPA\Tugas\06>
```

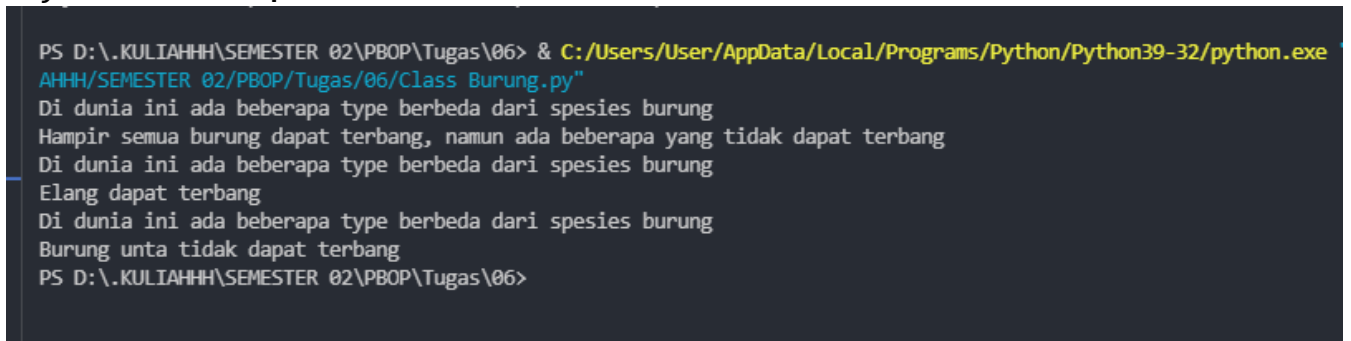
2.



```
1 class burung:
2     def intro(self):
3         print("Di dunia ini ada beberapa type berbeda dari spesies burung")
4
5     def terbang(self):
6         print("Hampir semua burung dapat terbang, namun ada beberapa yang tidak dapat terbang")
7
8 class Elang(burung):
9     def terbang(self):
10        print("Elang dapat terbang")
11
12 class BurungUnta(burung):
13     def terbang(self):
14        print("Burung unta tidak dapat terbang")
15
16 obj_burung = burung()
17 obj_elang = Elang()
18 obj_burung_unta = BurungUnta()
19
20 obj_burung.intro()
21 obj_burung.terbang()
22
23 obj_elang.intro()
24 obj_elang.terbang()
25
26 obj_burung_unta.intro()
27 obj_burung_unta.terbang()
```

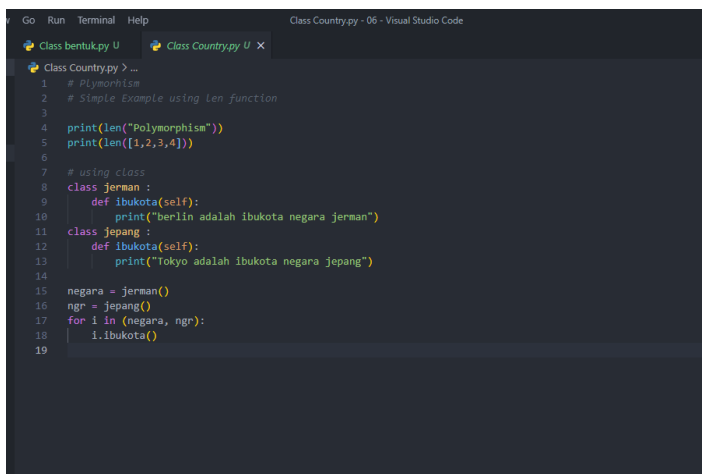
### Penjelasan :

Class burung digunakan untuk menampung objek yang menjelaskan tentang burung. Kemudian class Elang mengambil data dari class burung yang kemudian digunakan untuk menampung objek yang menjelaskan tentang elang. Kemudian ada class BurungUnta yang digunakan untuk menampung objek yang menjelaskan tentang BurungUnta. Setelah itu setiap class nanti akan dicetak dan ditampilkan pada layar user. **Output :**



```
PS D:\KULIAHH\SEMESTER 02\PBOP\Tugas\06> & C:/Users/User/AppData/Local/Programs/Python/Python39-32/python.exe AH\SEMESTER 02\PBOP\Tugas\06\Class Burung.py"
Di dunia ini ada beberapa type berbeda dari spesies burung
Hampir semua burung dapat terbang, namun ada beberapa yang tidak dapat terbang
Di dunia ini ada beberapa type berbeda dari spesies burung
Elang dapat terbang
Di dunia ini ada beberapa type berbeda dari spesies burung
Burung unta tidak dapat terbang
PS D:\KULIAHH\SEMESTER 02\PBOP\Tugas\06>
```

3.



```
1 # Polymorphism
2 # Simple Example using len function
3
4 print(len("Polymorphism"))
5 print(len([1,2,3,4]))
6
7 # using class
8 class Jerman:
9     def ibukota(self):
10        print("berlin adalah ibukota negara jerman")
11
12 class Jepang:
13     def ibukota(self):
14        print("Tokyo adalah ibukota negara jepang")
15
16 negara = [Jerman(), Jepang()]
17 for i in (negara, ngr):
18     i.ibukota()
19
```

### Penjelasan :

Fungsi print digunakan untuk menampilkan suatu objek kepada user kemudian di dalam fungsi print ada sebuah "len" yang digunakan untuk mengetahui Panjang dari sebuah variabel atau objek yang berada di dalamnya. Lalu ada berbagai class yang digunakan untuk mendefinisikan masing-masing objek pada tiap class yang nantinya akan ditampilkan pada layar user. **Output :**

```
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06> & C:/Users/User/AppData/Local/Programs/Python/Python39-32/python.exe "d:/KULIAHHH/SEMESTER 02/PBOP/Tugas/06/Class Country.py"
12
4
berlin adalah ibukota negara jerman
Tokyo adalah ibukota negara jepang
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06>
```

4.

```
Class kucing.py > Dog > bersuara
1 class Kucing:
2     def __init__(self, nama, umur):
3         self.nama = nama
4         self.umur = umur
5
6     def bersuara(self):
7         print('Meow')
8
9 class Dog:
10    def __init__(self, nama, umur):
11        self.nama = nama
12        self.umur = umur
13
14    def bersuara(self):
15        print('Guk...guk...')
16
17 kucing1 = Kucing("Tom", 3)
18 anjing1 = Dog("Spike", 4)
19
20 for hewan in (kucing1, anjing1):
21     hewan.bersuara()
22
```

### Penjelasan :

Class Kucing digunakan untuk menampung objek kucing berupa nama dan umur. Kemudian di dalam class kucing ada fungsi bersuara yang digunakan untuk menampung objek suara kucing yaitu Meow. Begitupula pada class dog yang memiliki fungsi bersuara yang menampung suara anjing yaitu Guk...guk... yang kemudian nantinya akan ditampilkan pada layar user. **Output :**

```
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06> & C:/Users/User/AppData/Local/Programs/Python/Python39-32/python.exe "d:/KULIAHHH/SEMESTER 02/PBOP/Tugas/06/Class kucing.py"
Meow
Guk...guk...
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06>
```

5.

```
class Segiempat.py : m
1  class Segiempat():
2      def __init__(self, panjang, lebar):
3          self.panjang = panjang
4          self.lebar    = lebar
5
6      def hitungLuas(self):
7          print("Luas Segiempat =", self.panjang * self.lebar, "m^2")
8
9  class Bujursangkar(Segiempat):
10     def __init__(self, sisi):
11         self.side = sisi
12         Segiempat.__init__(self, sisi, sisi)
13
14     def hitungLuas(self):
15         print("Luas bujur sangkar =", self.side*self.side, "m^2")
16
17  b = Bujursangkar(4)
18  s = Segiempat(2,4)
19  b.hitungLuas()
20  s.hitungLuas()
```

#### Penjelasan :

Class “Segiempat” digunakan untuk menampung fungsi yang nantinya akan digunakan untuk menghitung luas dari bangun ruang segiempat. Kemudian class “Bujursangkar” mengambil data dari class “Segiempat” yang nantinya akan digunakan untuk menghitung luas sebuah bangun ruang bujur sangkar. Kemudian setelah dihitung akan dicetak dan ditampilkan pada layar user.

#### Output :

```
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06> & C:/Users/User/AppData/Local/Programs/Python/Py
AHHH/SEMESTER 02/PBOP/Tugas/06/Class Segiempat .py"
Luas bujur sangkar = 16 m^2
Luas Segiempat = 8 m^2
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06>
```

6.

```
Mahasiswa_Overload.py > ...
1 # Mahasiswa Overloading
2 class Mahasiswa:
3     def __init__(self, nama, nim):
4         self.nama = nama
5         self.nim = nim
6
7     def tampilMhs(self):
8         print("Nama:", self.nama, ", nim:", self.nim)
9
10    # Method Overloading
11    def hitungSKS(self, jmlsks=None, sks=None):
12        if jmlsks !=None and sks!=None:
13            totalsks = jmlsks + sks
14            print("Total sks =", totalsks)
15        else:
16            totalsks = jmlsks
17            print("Total sks =", totalsks)
18
19        if totalsks >= 100:
20            print("Diperbolehkan mengambil skripsi")
21        else:
22            print("Tidak diperbolehkan mengambil skripsi")
23
24
25 mhs1 = Mahasiswa("Eren", 123180015)
26 mhs2 = Mahasiswa("Luffy", 123190007)
27 mhs1.tampilMhs()
28 mhs2.tampilMhs()
29 mhs1.hitungSKS(80, 34)
30 mhs2.hitungSKS(83)]
```

#### Penjelasan :

Class Mahasiswa digunakan untuk menampung sebuah fungsi yang berisi tentang objek yang nantinya akan digunakan untuk menampung nama dan nim mahasiswa. Kemudian fungsi HitungSKS digunakan untuk menghitung total sks yang didapatkan oleh mahasiswa, tidak hanya itu fungsi ini juga digunakan untuk memberi info apakah mahasiswa tersebut bisa mengambil skripsi atau tidak.

#### Output :

```
PS D:.\KULIAHHH\SEMESTER 02\PBOP\Tugas\06> & C:/User
AHHH/SEMESTER 02/PBOP/Tugas/06/Mahasiswa_Overload.py
Nama: Eren , nim: 123180015
Nama: Luffy , nim: 123190007
Total sks = 114
Diperbolehkan mengambil skripsi
Total sks = 83
Tidak diperbolehkan mengambil skripsi
PS D:.\KULIAHHH\SEMESTER 02\PBOP\Tugas\06>
```

7.

```
1 class Mahasiswa:
2     def __init__(self, nama, nim):
3         self.nama = nama
4         self.nim = nim
5
6     def detSiswa(self):
7         print(self.nama, 'alamat: wall rose')
8
9 class Siswa(Mahasiswa):
10     def __init__(self, nama, nim):
11         self.nama = nama
12         self.nim = nim
13
14     def detSiswa(self):
15         print(self.nama, 'Jurusan: Informatika')
16
17 mhs1 = Siswa('Mikasa', 135105)
18 mhs2 = Mahasiswa('Nezuko', 135119)
19
20 print(mhs1.nim, mhs1.nama)
21 mhs1.detSiswa()
22 print(mhs2.nim, mhs2.nama)
23 mhs2.detSiswa()
```

#### Penjelasan :

Class Mahasiswa merupakan class yang nantinya akan digunakan untuk menampung data berupa nama dan nim begitu juga dengan class Siswa yang mengambil data dari Class Mahasiswa yang merupakan class parent. Yang kemudian nantinya akan dicetak dan ditampilkan pada layar user.

#### Output :

```
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06> & C:/Users/User/AppData/Local
AHHH/SEMESTER 02/PBOP/Tugas/06/Mahasiswa.py"
135105 Mikasa
Mikasa Jurusan: Informatika
135119 Nezuko
Nezuko alamat: wall rose
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06>
```

**8.**

```
class Pegawai:
    jumlah = 0

    def __init__(self, nama, gaji) -> None:
        self.nama = nama
        self.gaji = gaji
        Pegawai.jumlah += 1

    def tampiljumlah (self):
        print("Total Pegawai : ", Pegawai.jumlah)

    def tampilpegawai (self):
        print(f'''
Nama      : {self.nama}
Gaji      : {self.gaji}
''')

    def tunjangan(sekkf, istri = None, anak = None):
        if anak != None and istri != None :
            total = anak + istri
            print(f'''
Tunjangan Istri Dan Anak      : {total}''')
        else :
            total = istri
            print(f'''
Tunjangan Istri      : {total}''')

peg1 = Pegawai("EREN", 2000)
peg2 = Pegawai("DEA", 6000)

peg1.tampilpegawai()
peg2.tampilpegawai()
peg1.tunjangan(2500.2500)
peg2.tunjangan(2500)

print("Total Peggawai %d" % Pegawai.jumlah)
rataGaji = (peg1.gaji + peg2.gaji)/ Pegawai.jumlah
print("Rata - rata Gaji Pegawai : ", str(rataGaji))
```

**Penjelasan :**

Class Pegawai menampung berbagai macam fungsi yang kemudian nantinya masing-masing fungsi tersebut akan dicetak dan ditampilkan ke layar user sebagai data yang diinginkan.

**Output :**

```
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06> & C:/Users/User/AppData/Local/
AHHH/SEMESTER 02/PBOP/Tugas/06/Ovrloading_pegawai.py"

Nama      : EREN
Gaji      : 2000

Nama      : DEA
Gaji      : 6000

Tunjangan Istri      : 2500.25

Tunjangan Istri      : 2500
Total Pegawai 2
Rata - rata Gaji Pegawai : 4000.0
PS D:\.KULIAHHH\SEMESTER 02\PBOP\Tugas\06>
```



9.

```
Tugas_ComputePart_Overloading copy.py > ...
1 class ComputerPart:
2     def __init__(self, pabrik, nama):
3         self.pabrik = pabrik
4         self.nama = nama
5
6     def detComputer(self):
7         print('Pabrik: ', self.pabrik, 'nama: ', self.nama)
8
9     def HitungHarga(self, harga=None):
10        if harga != None:
11            h = harga
12            print('Harga: ', h)
13        else:
14            print('Tidak ada pesanan')
15
16 com1 = ComputerPart('LENOVO', 'THINK STRIX')
17 com2 = ComputerPart('ASUS', 'TUF GAMING')
18 com1.detComputer()
19 com2.detComputer()
20 com1.HitungHarga(100000)
21 com2.HitungHarga(200000)
```

#### Penjelasan :

Class “ComputerPart” memiliki berbagai fungsi didalamnya berupa “detComputer” yang digunakan untuk mengetahui data pabrik dan nama dari komputer. Kemudian ada fungsi “HitungHarga” yang digunakan untuk mencari tau harga dari sebuah komputer.

```
AHHH/SEMESTER 02/PBOP/Tugas/06/Tugas_ComputePart_
Pabrik: LENOVO nama: THINK STRIX
Pabrik: ASUS nama: TUF GAMING
Harga: 100000
Harga: 200000
PS D:\KULIAHHH\SEMESTER 02\PBOP\Tugas\06>
```

10.

```
Tugas_ComputePart_Overriding.py > ...
1 class ComputerPart:
2     def __init__(self, pabrikan, nama):
3         self.pabrikan = pabrikan
4         self.nama = nama
5
6     def detComputer(self):
7         print('Nama:',self.nama,'Harga: Rp.1000000')
8
9 class ComputerPart_2(ComputerPart):
10    def __init__(self, pabrikan, nama):
11        self.pabrikan = pabrikan
12        self.nama = nama
13
14    def detComputer(self):
15        print('Nama:',self.nama,'Harga: Rp.200000000')
16
17 com1 = ComputerPart_2('Asus' , 'ROG STRIX PRO')
18 com2 = ComputerPart('Asus' , 'TUF GAMING PRO')
19
20 print(com1.pabrikan, com1.nama)
21 com1.detComputer()
22
23 print(com2.pabrikan, com2.nama)
24 com2.detComputer()
```

#### Penjelasan :

Class “ComputerPart” adalah class induk dan class “ComputerPart2” adalah class child. Class child mengambil data yang ada pada class induk yang kemudian di tampilkan Ketika kita sudah memberikan suatu objek pada setiap class yang didefinisikan.

```
PS D:\.KULIAHH\SEMESTER 02\PBOP\Tugas\06> & C:/Users/
AH\HH/SEMESTER 02/PBOP/Tugas/06/Tugas_ComputePart_Over
Asus ROG STRIX PRO
Nama: ROG STRIX PRO Harga: Rp.200000000
Asus TUF GAMING PRO
Nama: TUF GAMING PRO Harga: Rp.1000000
PS D:\.KULIAHH\SEMESTER 02\PBOP\Tugas\06>
```

#### Kesimpulan :

Dari setiap program yang sudah dibuat diatas dapat disimpulkan dengan penggunaan polymorphism, overloading, overriding, dan abstraction kita dapat mempersingkat source code yang kita tulis untuk membuat suatu program. Namun memerlukan ketelitian yang lebih intens.