

POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
JOBSHEET-11 POLIMORFISME
2C/20/SHINE DEVI OKTAVIANA R S P

Pertanyaan Percobaan 1

1. Class apa sajakah yang merupakan turunan dari class **Employee**?
 - InternshipEmployee dan PermanentEmployee
2. Class apa sajakah yang implements ke interface **Payable**?
 - PermanentEmployee dan ElectricityBill
3. Perhatikan class **Tester1**, baris ke-10 dan 11. Mengapa **e**, bisa diisi dengan objek **pEmp** (merupakan objek dari class **PermanentEmployee**) dan objek **iEmp** (merupakan objek dari class **InternshipEmployee**) ?
 - Karena e adalah objek dari class Employee yang merupakan parent dari PermanentEmployee dan InternshipEmployee
4. Perhatikan class **Tester1**, baris ke-12 dan 13. Mengapa **p**, bisa diisi dengan objek **pEmp** (merupakan objek dari class **PermanentEmployee**) dan objek **eBill** (merupakan objek dari class **ElectricityBill**) ?
 - Karena p merupakan objek dari class Payable yang merupakan interface. Class Payable ini diimplementasikan pada class PermanentEmployee dan ElectricityBill
5. Coba tambahkan sintaks:
p = iEmp;
e = eBill;
pada baris 14 dan 15 (baris terakhir dalam method **main**) ! Apa yang menyebabkan error?
 - iEmp tidak mengimplementasikan interface Payable dan eBill bukan merupakan child dari Employee

POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
JOBSHEET-11 POLIMORFISME
2C/20/SHINE DEVI OKTAVIANA R S P

6. Ambil kesimpulan tentang konsep/bentuk dasar polimorfisme!
 - Objek yang memiliki berbagai kondisi yang dapat diakses di interface yang sama

Pertanyaan Percobaan 2

1. Perhatikan class **Tester2** di atas, mengapa pemanggilan **e.getEmployeeInfo()** pada baris 8 dan **pEmp.getEmployeeInfo()** pada baris 10 menghasilkan hasil sama?
 - pEmp merupakan child dari Employee. Karena hanya pEmp yang diberikan input nilai, maka input nilai tersebut juga digunakan oleh e
2. Mengapa pemanggilan method **e.getEmployeeInfo()** disebut sebagai (virtual method invocation), sedangkan **pEmp.getEmployeeInfo()** tidak?
 - Karena, pEmp.getEmployeeInfo() memanggil overridden method pada parent class (Employee).
3. Jadi apakah yang dimaksud dari virtual method invocation? Mengapa disebut virtual?
 - Pemanggilan overriding method pada subclass, dimana seharusnya memanggil overridden method pada parent class.
 - Disebut virtual karena method overridden dan overriding memiliki kemiripan, yaitu memiliki nama method yang sama, namun operasinya berbeda

Pertanyaan Percobaan 3

1. Perhatikan array **e** pada baris ke-8, mengapa ia bisa diisi dengan objek-objek dengan tipe yang berbeda, yaitu objek **pEmp** (objek dari **PermanentEmployee**) dan objek **iEmp** (objek dari **InternshipEmployee**)
 - Karena pEmp dan iEmp merupakan objek dari subclass yang merupakan turunan dari objek e
2. Perhatikan juga baris ke-9, mengapa array **p** juga diisi dengan objek-objek dengan tipe yang berbeda, yaitu objek **pEmp** (objek dari **PermanentEmployee**) dan objek **eBill** (objek dari **ElectricityBilling**) ?
 - Karena objek pEmp dan eBill mengimplementasikan interface dari objek p (interface class Payable)
3. Perhatikan baris ke-10, mengapa terjadi error?
 - Karena eBill bukan merupakan child dari e

POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
JOBSHEET-11 POLIMORFISME
2C/20/SHINE DEVI OKTAVIANA R S P

Pertanyaan Percobaan 4

1. Perhatikan class **Tester4** baris ke-7 dan baris ke-11, mengapa pemanggilan **ow.pay(eBill)** dan **ow.pay(pEmp)** bisa dilakukan, padahal jika diperhatikan method **pay()** yang ada di dalam class **Owner** memiliki argument/parameter bertipe **Payable**?

- Karena eBill dan pEmp mengimplementasikan interface Payable

Jika diperhatikan lebih detil, eBill merupakan objek dari ElectricityBill dan pEmp merupakan objek dari PermanentEmployee.

2. Jadi apakah tujuan membuat argument bertipe **Payable** pada method **pay()** yang ada di dalam class **Owner**?
- untuk memproses method dari class yang mengimplementasikan interface Payable

3. Coba pada baris terakhir method **main()** yang ada di dalam class **Tester4** ditambahkan perintah **ow.pay(iEmp);**
Mengapa terjadi error?

- Karena iEmp tidak mengimplementasikan interface Payable

4. Perhatikan class **Owner**, diperlukan untuk apakah sintaks **p instanceof ElectricityBill** pada baris ke-6 ?

- untuk dapat membuat objek ElectricityBill yang mengimplementasikan interface Payable

5. Perhatikan kembali class Owner baris ke-7, untuk apakah casting objek disana (**ElectricityBill eb = (ElectricityBill) p**) diperlukan ?
Mengapa objek **p** yang bertipe **Payable** harus di-casting ke dalam objek **eb** yang bertipe **ElectricityBill** ?

- karena eb merupakan p(sudah mengimplementasikan interface Payable)

- di-casting ke objek eb karena agar dapat mengakses method yang ada di ElectricityBill

POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
JOBSHEET-11 POLIMORFISME
2C/20/SHINE DEVI OKTAVIANA R S P

TUGAS

```
Output - Zombie (run) × Zombie.java × WalkingZombie.java × IDestroyable.java ×
run:
Walking Zombie Data
Health: 100
Level: 1

Jumping Zombie Data:
Health: 100
Level: 2

Strength Barrier: 100

-----
Walking Zombie Data
Health: 42
Level: 1
Jumping Zombie Data:
Health: 67
Level: 2

Strength Barrier: 65
BUILD SUCCESSFUL (total time: 0 seconds)
```

POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
JOBSHEET-11 POLIMORFISME
2C/20/SHINE DEVI OKTAVIANA R S P

```
Output - Zombie (run) × Zombie.java × WalkingZombie.java × IDestroyable.java ×
run:
Walking Zombie Data
Health: 100
Level: 1

Jumping Zombie Data:
Health: 100
Level: 2

Strength Barrier: 100

-----
Walking Zombie Data
Health: 42
Level: 1
Jumping Zombie Data:
Health: 67
Level: 2

Strength Barrier: 65
BUILD SUCCESSFUL (total time: 0 seconds)
```