Tugas Pemahaman Konsep OOP

Tugas-Mg6 PBO

Student : Reza Chairul Manam, reza.120140086@student.itera.ac.id
Student ID Number : 120140086
Class : PBO-RC
Lecturer : I Wayan Wiprayoga Wisesa S.Kom., M.Kom, wayan.wisesa@if.itera.ac.id

Listing 1: Tugas-Mg6 PBO-RC

```python
#Tugas Mg6
# File: 06_Tugas.py
# Author: Reza Chairul Manam
# NIM: 120140086
# Class: PBO-RC
# ==========================================================

from abc import ABC, abstractmethod # Import modul ABC dan abstractmethod

# Abstraksi
class AlatMusik(ABC): # Kelas AlatMusik
    def __init__(self, nama, jenis):
        self.nama = nama            # Enkapsulasi (protected)
        self.jenis = jenis          # Enkapsulasi (protected)
        self.__volume = 50          # Enkapsulasi (private) (default volume)

    @abstractmethod # Dekorator abstractmethod
    def mainkan(self): # Method abstract mainkan
        pass

    def atur_volume(self, level): # Setter untuk atribut private
        if 0 <= level <= 100:
            self.__volume = level
        else:
            print("Volume harus antara 0 - 100")

    def get_volume(self): # Getter untuk atribut private
        return self.__volume  # Getter untuk atribut private

# Pewarisan
class Gitar(AlatMusik): # Kelas Gitar
    def __init__(self, nama, jumlah_senar):
        super().__init__(nama, "Petik")
        self.jumlah_senar = jumlah_senar

    # Polimorfisme
    def mainkan(self): # Method mainkan
        return f"Memainkan {self.nama} dengan {self.jumlah_senar} senar.
   Volume: {self.get_volume()}"

class Piano(AlatMusik): # Kelas Piano
    def __init__(self, nama, jumlah_tuts):
        super().__init__(nama, "Tuts")
```

```
        self.jumlah_tuts = jumlah_tuts

    # Polimorfisme
    def mainkan(self): # Method mainkan
        return f"Memainkan {self.nama} dengan {self.jumlah_tuts} tuts. Volume:
    {self.get_volume()}"

# Tambah alat musik, buat kelas baru yang mewarisi AlatMusik.
# class Bass(AlatMusik): # Kelas Bass
#     def __init__(self, nama, jumlah_senar):
#         super().__init__(nama, "Petik")
#         self.jumlah_senar = jumlah_senar

#     # Polimorfisme
#     def mainkan(self): # Method mainkan
#         return f"Memainkan {self.nama} dengan {self.jumlah_senar} senar.
   Volume: {self.get_volume()}"

# Implementasi
gitar1 = Gitar("Gitar Akustik", 6) # Objek Gitar
piano1 = Piano("Grand Piano", 88) # Objek Piano
# bass1 = Bass("Bass Elektrik", 4) # Objek Bass

gitar1.atur_volume(75)  # Mengatur volume gitar
piano1.atur_volume(60)  # Mengatur volume piano
# bass1.atur_volume(80)   # Mengatur volume bass

print(gitar1.mainkan())
print(piano1.mainkan())
# print(bass1.mainkan())
```

Listing 2: Hasil Eksekusi Program

```
#06_Tugas.py
#output
# Memainkan Gitar Akustik dengan 6 senar. Volume: 75
# Memainkan Grand Piano dengan 88 tuts. Volume: 60
```

1 Penjelasan

Program ini menerapkan konsep *Object-Oriented Programming* (OOP) dengan menggunakan abstraksi, enkapsulasi, pewarisan, dan polimorfisme dalam simulasi alat musik. Kelas abstrak `AlatMusik` berperan sebagai induk dengan atribut `nama` dan `jenis` yang dienkapsulasi secara *protected*, serta atribut `volume` yang dienkapsulasi secara *private*. Kelas ini memiliki metode abstrak `mainkan()` yang harus diimplementasikan oleh kelas turunannya.

Pewarisan diterapkan pada kelas `Gitar` dan `Piano`, yang merupakan turunan dari `AlatMusik`. Kelas-kelas ini mengimplementasikan metode `mainkan()` sesuai dengan karakteristik masing-masing alat musik (*polymorphism*). Selain itu, terdapat metode `atur_volume()` sebagai *setter* dan `get_volume()` sebagai *getter* untuk mengatur volume alat musik. Program ini memungkinkan pengguna untuk mengatur volume alat musik sebelum memainkannya, dan menampilkan informasi mengenai alat musik yang sedang dimainkan beserta volumenya.

Pada implementasi, dibuat objek `gitar1` dan `piano1` yang masing-masing memiliki jumlah senar dan jumlah tuts tertentu. Volume alat musik dapat disesuaikan sebelum dimainkan, dan hasil keluaran menunjukkan alat musik yang dimainkan beserta volumenya. Dengan pendekatan ini, program dapat dengan mudah diperluas untuk mendukung alat musik lainnya dengan mewarisi kelas `AlatMusik`.

Link Source Code

Source code lengkap di tautan berikut: github.com/rezachairul/Latihan-PBO-RC-2025