

Lengkapi Code dibawah ini

Tugas Mg-09 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-1 Mg-09 PBO-RC — Lengkapi Script dibawah ini

```
#Tugas-1 Mg-09
from abc import _____, _____

# --- Abstract Base Class ---
class LibraryItem(_____):
    def __init__(self, title: str, item_id: str):
        self._____ = title # Private attribute
        self._____ = item_id # Private attribute
        self.checked_out = False

    @property
    def title(self):
        return self._____

    @property
    def item_id(self):
        return self._____

    @abstractmethod
    def _____(self): # Abstract method for checkout
        pass

    @abstractmethod
    def return_item(self):
        pass

    def _____(self) -> str: # String representation
        return f"{self.__class__.__name__}: {self.title} (ID: {self.
            item_id})"

# --- Book Subclass ---
class Book(LibraryItem):
    def __init__(self, title: str, item_id: str, author: str):
        _____._____ (self, title, item_id)
        self._____ = author # Private attribute

    @property
    def author(self):
        return self._____

    def check_out(self):
        if not self.checked_out:
            self.checked_out = _____
```

```

        print(f"Book '{self.title}' checked out.")
    else:
        print("Book already checked out.")

    def return_item(self):
        if self.checked_out:
            self.checked_out = False
            print(f"Book '{self.title}' returned.")

# --- DVD Subclass ---
class DVD(Book):
    def __init__(self, title: str, item_id: str, duration: int):
        super().__init__(title, item_id)
        self.__duration = duration # Private attribute

    def check_out(self):
        if not self.checked_out:
            print(f"DVD '{self.title}' checked out for {self.__duration}
                minutes.")
            self.checked_out = True
        else:
            print("DVD already checked out.")

    def __str__(self):
        return f"DVD: {self.title} (Duration: {self.__duration} mins)&
            quot;

# --- Library Class (Composition) ---
class Library:
    def __init__(self):
        self.__items = [] # List of LibraryItem objects

    def add_item(self, item: LibraryItem):
        self.items.append(item)
        print(f"Added {item.title} to the library.")

    def search_by_title(self, title: str) -> list:
        return [item for item in self.items if item.title.lower() in title.lower()]

    def __len__(self) -> int: # Total items in library
        return len(self.__items)

    def __add__(self, other: 'Library') -> 'Library':
        new_library = Library()
        new_library.items = self.__items + other.__items
        return new_library

# --- Test Code ---
if __name__ == '__main__':
    book1 = Book("The Python Guide", "BK001", "Alice Smith")
    dvd1 = DVD("Learn OOP in 30 Days", "DVD001", 120)

```

```
library = Library()
library.add_item(book1)
library._____(dvd1)

book1.check_out()
dvd1.check_out()
dvd1._____( )

print(f"Total items: {library.total_items()}")
```

Jawaban Isian Tugas-1 Mg-09 PBO-RC

1. ABC, abstractmethod
2. ABC
3. __title
4. __item_id
5. __title
6. __item_id
7. check_out
8. __str__
9. super().__init__
10. __author
11. __author
12. True
13. False
14. LibraryItem
15. __duration
16. __str__
17. items
18. add_item
19. title
20. self.items
21. self.items, other.items
22. add_item
23. __str__

Foto Tulisan Tangan di Kertas

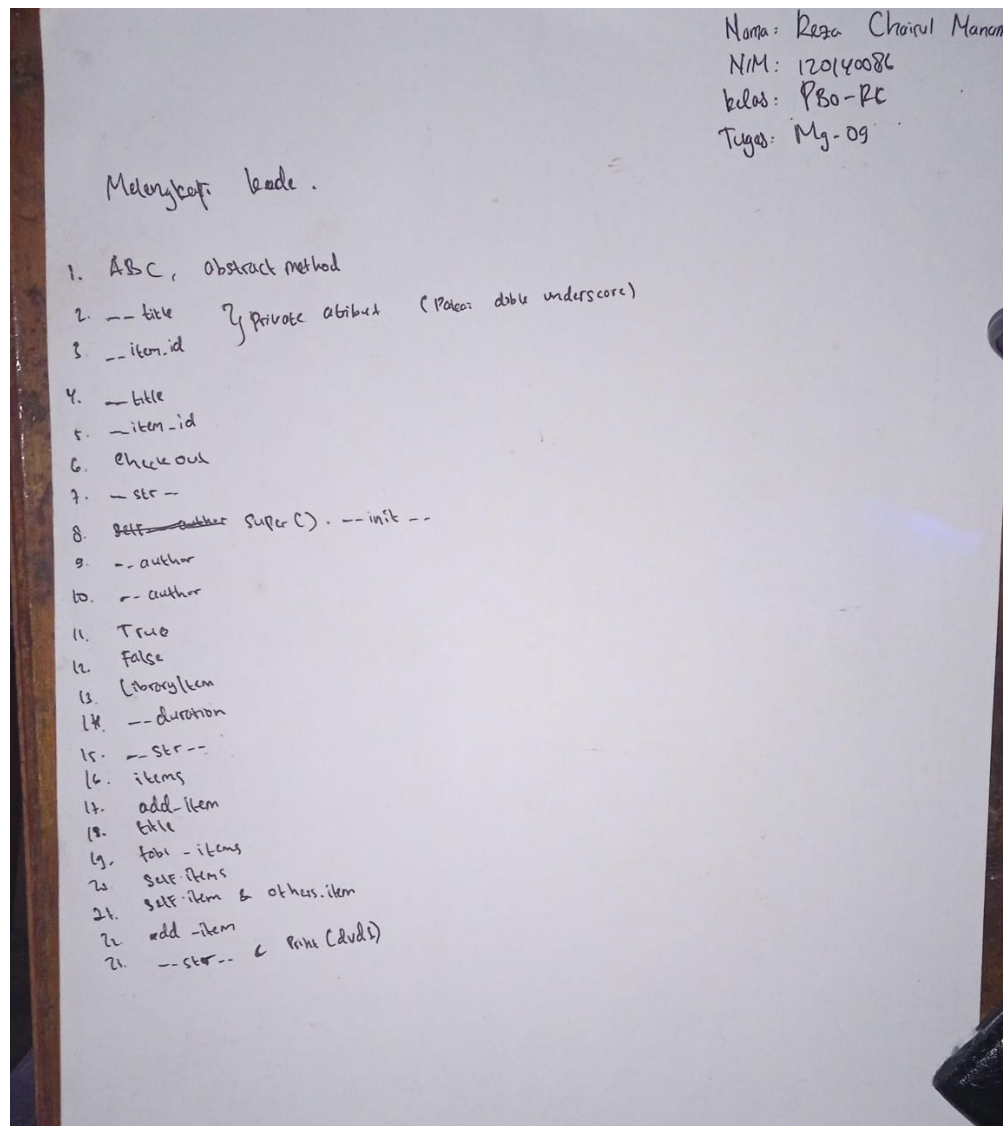


Figure 1: Jawaban ditulis tangan di kertas

Soal Example 1-4 Mg-09 PBO-RC

Exercise 1: Encapsulation & Getters/Setters

Create a Student class with private attributes and controlled access:

```
1 class Student:
2     def __init__(self, name: str, student_id: str):
3         self.__name = name # Private attribute for name
4         self.__student_id = student_id # Private attribute for ID
5         self.__grades = [] # Private list of grades
6
7     # Add a read-only property for student_id
8     @property
9     def student_id(self):
10         return self.__student_id
11
12     # Add a setter for name (allow only non-empty strings)
13     @property
14     def name(self, new_name: str):
15         if isinstance(new_name, str) and new_name.strip() != "":
16             self.__name = new_name.strip()
17
18     def add_grade(self, grade: float):
19         if 0 <= grade <= 100:
20             self.__grades.append(grade)
21
22     def average_grade(self) -> float:
23         if len(self.__grades) == 0:
24             return 0.0
25         return sum(self.__grades) / len(self.__grades)
```

Exercise 2: Inheritance & Method Overriding

Figure 2: Soal Example 1 Mg-09 PBO-RC



Figure 3: Soal Example 2 Mg-09 PBO-RC



Figure 4: Soal Example 4 Mg-09 PBO-RC

Jawaban Isian Example-1 Mg-09 PBO-RC

1. `self.__name = name`
2. `self.__student_id = student_id`
3. `self.__grades = []`
4. `return self.__student_id`
5. `@name.setter`
6. `self.__name = new_name.strip()`
7. `if 0 <= grade <= 100:`
8. `self.__grades.append(grade)`
9. `if len(self.__grades) == 0:`
10. `return sum(self.__grades) / len(self.__grades)`

Jawaban Isian Example-2 Mg-09 PBO-RC

1. ABC, abstractmethod
2. ABC
3. Shape
4. Shape
5. `__init__`
6. `self.side * self.side`

Jawaban Isian Example-3 Mg-09 PBO-RC

1. ABC
2. `@abstractmethod`
3. `def process_payment(self, amount: float) -> bool:
 return True`
4. `payment_processor.process_payment(amount)`

Jawaban Isian Example-4 Mg-09 PBO-RC

1. "Vector"
2. other.x
3. other.y
4. __str__

Foto Tulisan Tangan di Kertas - Example 1-4

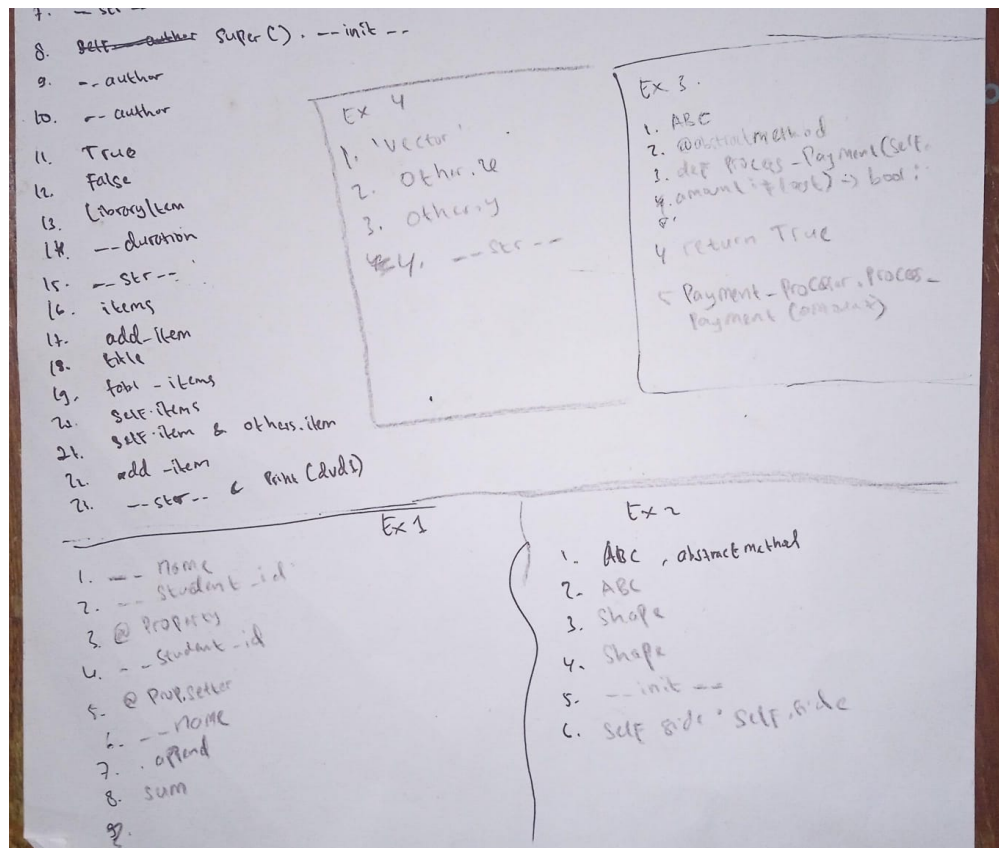


Figure 5: Jawaban Example 1-4 ditulis tangan di kertas

Link Source Code

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