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) -&gt; str:  # String representation
        return f&quot;{self.__class__.__name__}: {self.title} (ID: {self.
            item_id})&quot;

# --- 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&quot;Book '{self.title}' checked out.&quot;)
        else:
            print(&quot;Book already checked out.&quot;)

    def return_item(self):
        if self.checked_out:
            self.checked_out = ______
            print(f&quot;Book '{self.title}' returned.&quot;)

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

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

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

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

    def ______(self, item: LibraryItem):
        self.items.append(item)
        print(f&quot;Added {item.title} to the library.&quot;)

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

    def ______(self) -&gt; int:  # Total items in library
        return len(______)

    def __add__(self, other: 'Library') -&gt; 'Library':
        new_library = Library()
        new_library.items = ______ + ______
        return new_library

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

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

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

    print(f&amp;quot;Total items: {library.total_items()}&amp;quot;)
```

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

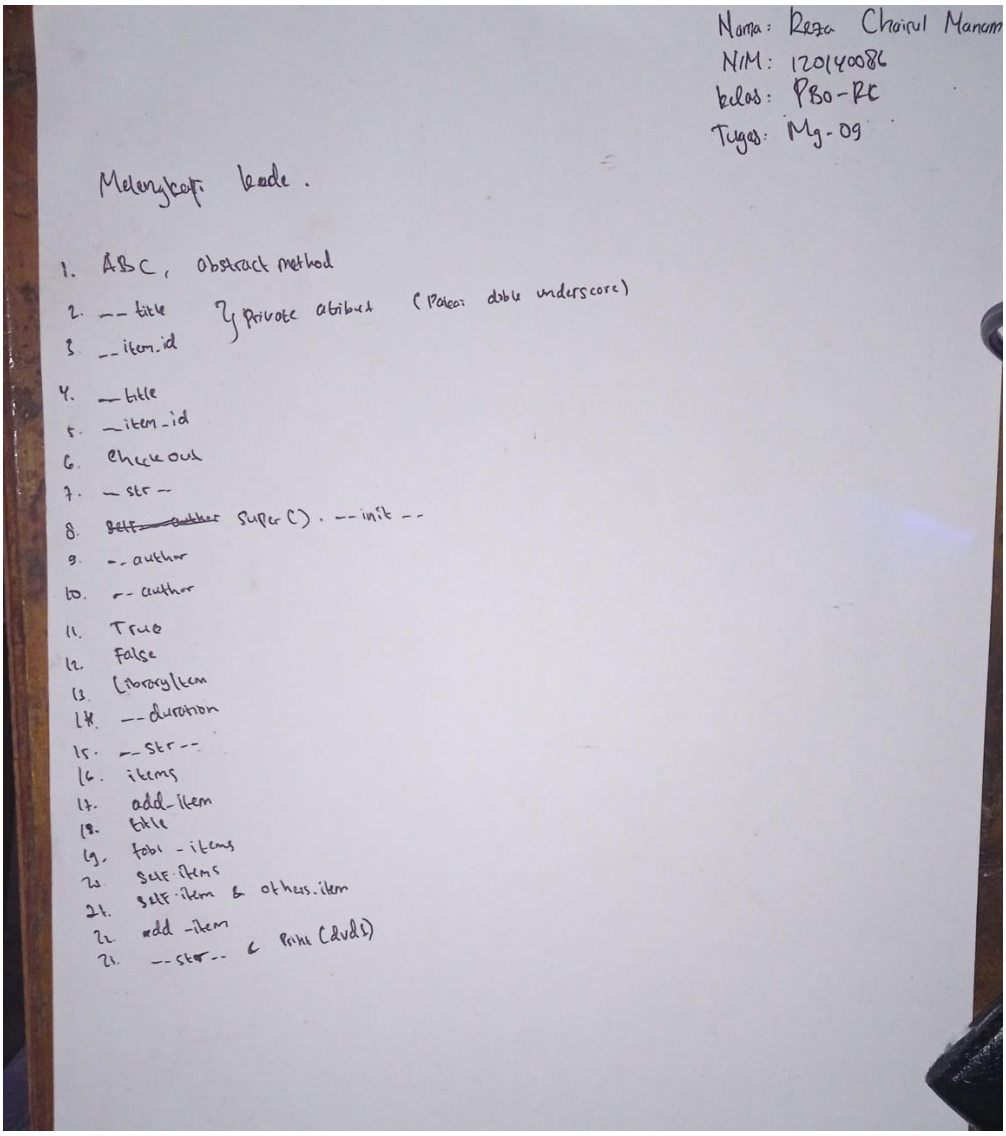

Figure 1: Jawaban ditulis tangan di kertas

Soal Example 1-4 Mg-09 PBO-RC

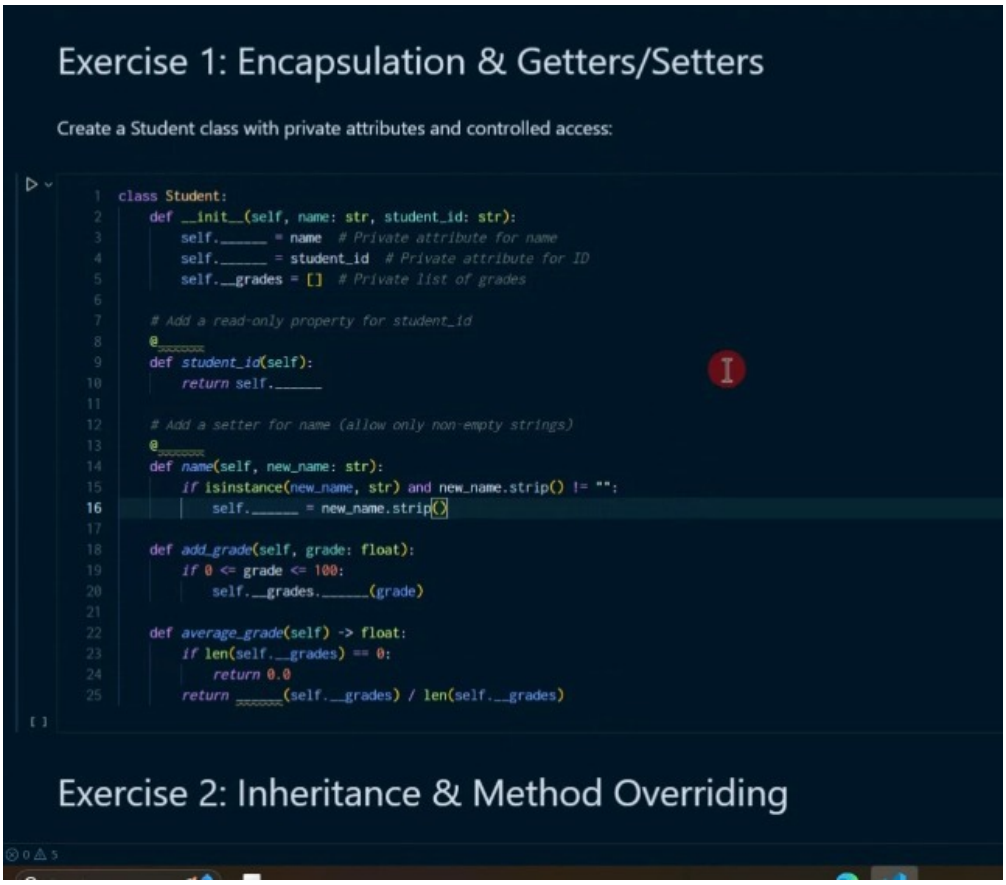

Figure 2: Soal Example 1 Mg-09 PBO-RC

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

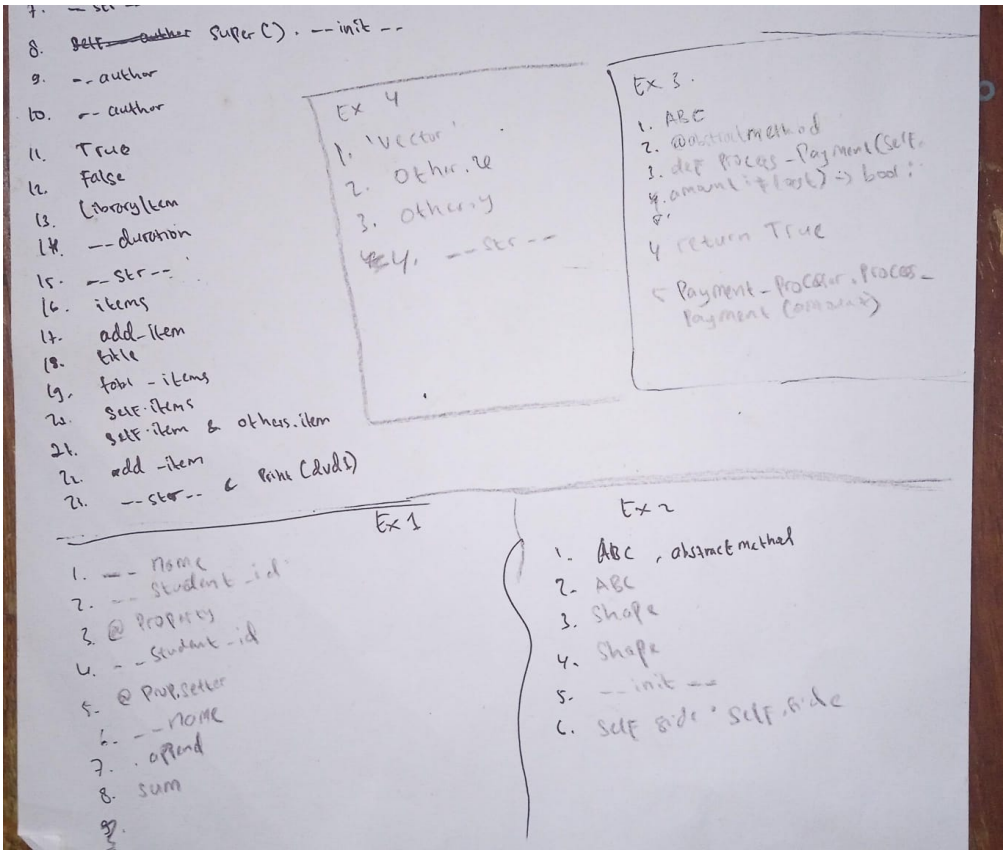

Figure 5: Jawaban Example 1-4 ditulis tangan di kertas

Link Source Code

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