Nama : Bravely Dirgayuska
NIM : 21120122140142
RPLBK B

“IMPLEMENTASI CODIGAN INTERFACE SEGREGATION PRINCIPLE”

Interface Segregation Principle (ISP) adalah salah satu prinsip SOLID dalam pemrograman yang menyatakan bahwa antarmuka (interface) sebaiknya tidak memaksa klien untuk mengimplementasikan metode yang tidak mereka gunakan. Dalam kata lain, sebaiknya kita memisahkan antarmuka besar menjadi beberapa antarmuka kecil yang lebih spesifik sesuai kebutuhan klien.

```
from abc import ABC, abstractmethod

# Antarmuka dasar untuk semua printer
class Printer(ABC):
    @abstractmethod
    def print_document(self, document: str) -> None:
        pass

# Antarmuka spesifik untuk printer yang memiliki fitur scan
class Scanner(ABC):
    @abstractmethod
    def scan_document(self) -> str:
        pass

# Antarmuka spesifik untuk printer yang memiliki fitur fax
class FaxMachine(ABC):
    @abstractmethod
    def send_fax(self, document: str) -> None:
        pass

# Implementasi kelas untuk printer yang hanya bisa print
class SimplePrinter(Printer):
    def print_document(self, document: str) -> None:
        print(f"Printing document: {document}")

# Implementasi kelas untuk printer multifungsi yang bisa print, scan, dan fax
class MultiFunctionPrinter(Printer, Scanner, FaxMachine):
    def print_document(self, document: str) -> None:
        print(f"Printing document: {document}")
```

```
    def scan_document(self) -> str:
        scanned_data = "Scanned document content"
        print(f"Scanning document...")
        return scanned_data

    def send_fax(self, document: str) -> None:
        print(f"Sending fax: {document}")

# Penggunaan kelas SimplePrinter
simple_printer = SimplePrinter()
simple_printer.print_document("Simple document")

# Penggunaan kelas MultiFunctionPrinter
multi_printer = MultiFunctionPrinter()
multi_printer.print_document("Multifunction document")
scanned_content = multi_printer.scan_document()
multi_printer.send_fax(scanned_content)
```

Kode di atas mendemonstrasikan penerapan Interface Segregation Principle (ISP) dengan memecah antarmuka besar menjadi antarmuka yang lebih kecil dan lebih spesifik sesuai dengan kebutuhan klien. Dalam contoh ini, antarmuka `Printer` hanya menangani fungsi pencetakan, sementara `Scanner` dan `FaxMachine` masing-masing menangani fungsi pemindaian dan pengiriman faks. Hal ini memungkinkan klien seperti `SimplePrinter`, yang hanya membutuhkan kemampuan mencetak, untuk mengimplementasikan hanya antarmuka `Printer` tanpa harus dipaksa untuk mengimplementasikan metode lain yang tidak relevan, seperti pemindaian atau pengiriman faks. Di sisi lain, `MultiFunctionPrinter`, yang membutuhkan semua kemampuan tersebut, dapat mengimplementasikan semua antarmuka yang diperlukan. Dengan demikian, setiap kelas hanya bergantung pada metode yang benar-benar dibutuhkan, sehingga kode menjadi lebih fleksibel, modular, dan mudah dipelihara, yang merupakan inti dari Interface Segregation Principle.