LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 12

2411102441052

Angga Maulana Saputra

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

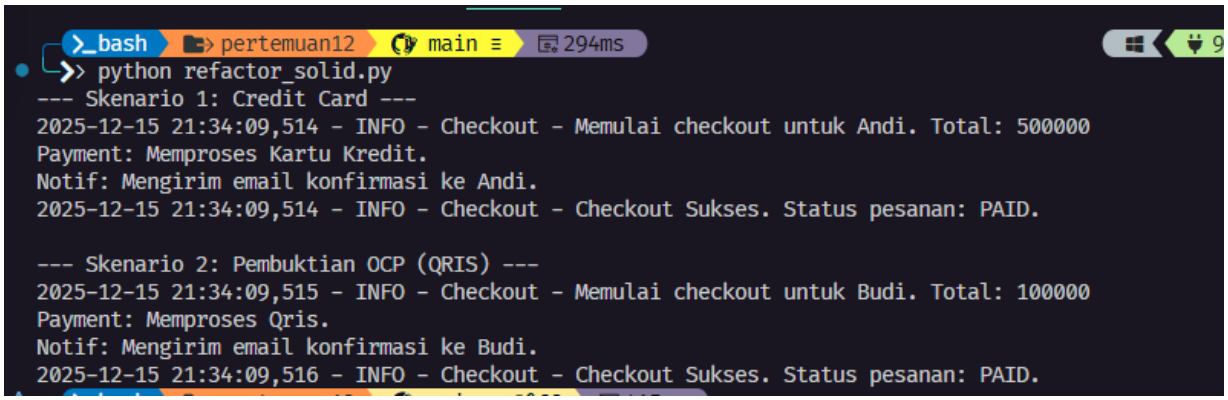
EMAIL : 2411102441052@umkt.ac.id

Link Github: https://github.com/nikamushi/praktikum-pbo/tree/main/pertemuan12

a. Screenshoot kode latihan

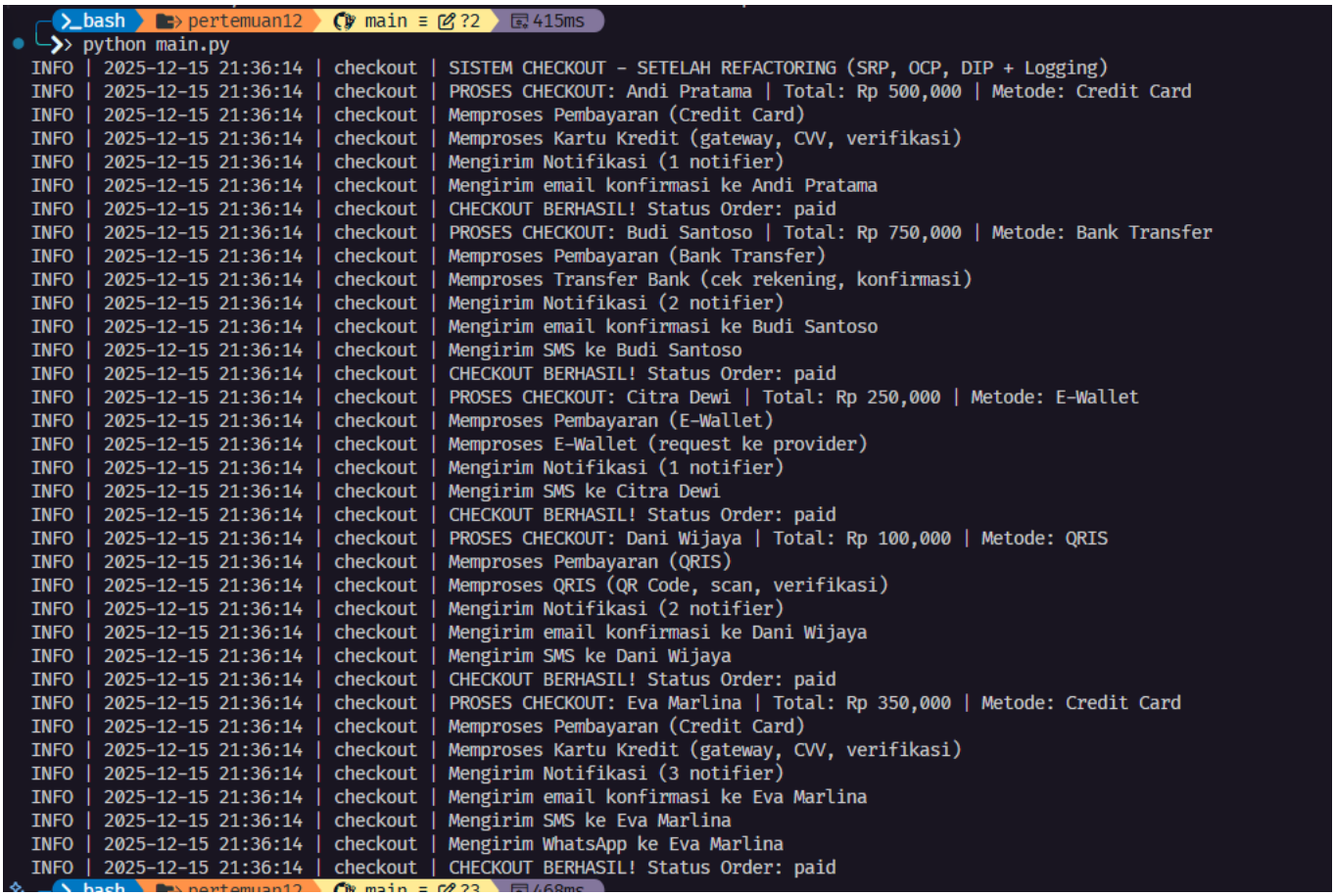
```python
from abc import ABC, abstractmethod
from dataclasses import dataclass
import logging

# Konfigurasi dasar: Semua log level INFO ke atas akan ditampilkan
# Format: Waktu - Level - Nama Kelas/Fungsi - Pesan
logging.basicConfig(
    level=logging.INFO,
    format='%(asctime)s - %(levelname)s - %(name)s - %(message)s'
)
# Tambahkan logger untuk kelas yang akan kita gunakan
LOGGER = logging.getLogger('Checkout')

@dataclass
class Order:
    customer_name: str
    total_price: float
    status: str = "open"

class IPaymentProcessor(ABC):
    """Kontrak: Semua prosesor pembayaran HRUS punya method 'process'."""
    @abstractmethod
    def process(self, order: Order) -> bool:
        pass

class INotificationService(ABC):
    """Kontrak: Semua layanan notifikasi harus punya method 'send'."""
    @abstractmethod
    def send(self, order: Order):
        pass

class CreditCardProcessor(IPaymentProcessor):
    def process(self, order: Order) -> bool:
        print("Payment: Memproses Kartu Kredit.")
        return True

class EmailNotifier(INotificationService):
    def send(self, order: Order):
        print(f"Notif: Mengirim email konfirmasi ke {order.customer_name}.")

class CheckoutService: # Tanggung jawab tunggal: Mengkoordinasikan Checkout
    """
    Kelas high-level untuk mengkoordinasi proses transaksi pembayaran.

    Kelas ini memisahkan logika pembayaran dan notifikasi (memenuhi SRP).
    """
    def __init__(self, payment_processor: IPaymentProcessor, notifier: INotificationService):
        """
        Menginisialisasi CheckoutService dengan dependensi yang diperlukan.

        Args:
            payment_processor (IPaymentProcessor): Implementasi interface pembayaran.
            notifier (INotificationService): Implementasi interface notifikasi.
        """
        self.payment_processor = payment_processor
        self.notifier = notifier

    def run_checkout(self, order: Order):
        """
        Menjalankan proses checkout dan memvalidasi pembayaran.

        Args:
            order (Order): Objek pesanan yang akan diproses.

        Returns:
            bool: True jika checkout sukses, False jika gagal.
        """
        # Logging akan ganti print()
        LOGGER.info(f"Memulai checkout untuk {order.customer_name}. Total: {order.total_price}")
        payment_success = self.payment_processor.process(order) # Delegasi 1

        if payment_success:
            order.status = "paid"
            self.notifier.send(order) # Delegasi 2
            LOGGER.info("Checkout Sukses. Status pesanan: PAID.")
            return True
        else:
            # Gunakan level ERROR/WARNING untuk masalah
            LOGGER.error("Pembayaran gagal. Transaksi dibatalkan.")
            return False

        pass

andi_order = Order("Andi", 500000)
email_service = EmailNotifier()

cc_processor = CreditCardProcessor()
checkout_cc = CheckoutService(payment_processor_cc_processor, notifier_email_service)
print("--- Skenario 1: Credit Card ---")
checkout_cc.run_checkout(andi_order)

class QrisProcessor(IPaymentProcessor):
    def process(self, order: Order) -> bool:
        print("Payment: Memproses Qris.")
        return True

budi_order = Order("Budi", 100000)
qris_processor = QrisProcessor()

checkout_qris = CheckoutService(payment_processor_qris_processor, notifier_email_service)
print("\n    Skenario 2: Pembuktian OCP (QRIS)    ")
checkout_qris.run_checkout(budi_order)
```

b. Output

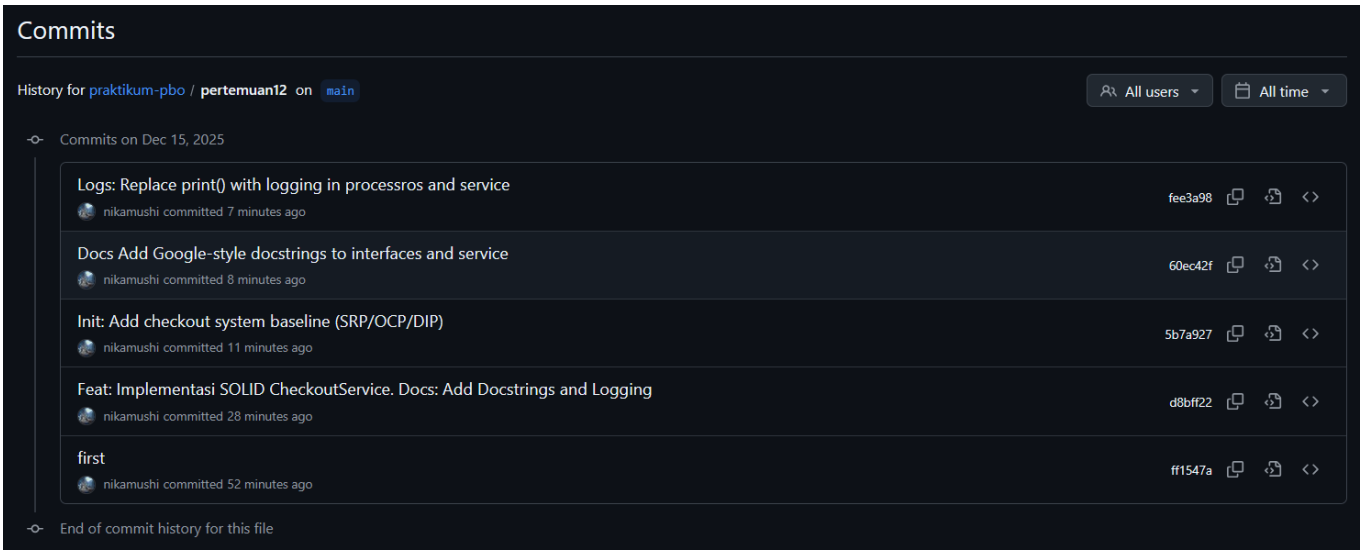

>_bash
pertemuan12
main ≡
294ms
>> python refactor_solid.py
--- Skenario 1: Credit Card ---
2025-12-15 21:34:09,514 - INFO - Checkout - Memulai checkout untuk Andi. Total: 500000
Payment: Memproses Kartu Kredit.
Notif: Mengirim email konfirmasi ke Andi.
2025-12-15 21:34:09,514 - INFO - Checkout - Checkout Sukses. Status pesanan: PAID.
--- Skenario 2: Pembuktian OCP (QRIS) ---
2025-12-15 21:34:09,515 - INFO - Checkout - Memulai checkout untuk Budi. Total: 100000
Payment: Memproses Qris.
Notif: Mengirim email konfirmasi ke Budi.
2025-12-15 21:34:09,516 - INFO - Checkout - Checkout Sukses. Status pesanan: PAID.

c. Screenshoot kode latihan mandiri

d. Output


bash
pertemuan12
main ≡ ?2
415ms
python main.py
INFO | 2025-12-15 21:36:14 | checkout | SISTEM CHECKOUT – SETELAH REFACTORING (SRP, OCP, DIP + Logging)
INFO | 2025-12-15 21:36:14 | checkout | PROSES CHECKOUT: Andi Pratama | Total: Rp 500,000 | Metode: Credit Card
INFO | 2025-12-15 21:36:14 | checkout | Memproses Pembayaran (Credit Card)
INFO | 2025-12-15 21:36:14 | checkout | Memproses Kartu Kredit (gateway, CVV, verifikasi)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim Notifikasi (1 notifier)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim email konfirmasi ke Andi Pratama
INFO | 2025-12-15 21:36:14 | checkout | CHECKOUT BERHASIL! Status Order: paid
INFO | 2025-12-15 21:36:14 | checkout | PROSES CHECKOUT: Budi Santoso | Total: Rp 750,000 | Metode: Bank Transfer
INFO | 2025-12-15 21:36:14 | checkout | Memproses Pembayaran (Bank Transfer)
INFO | 2025-12-15 21:36:14 | checkout | Memproses Transfer Bank (cek rekening, konfirmasi)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim Notifikasi (2 notifier)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim email konfirmasi ke Budi Santoso
INFO | 2025-12-15 21:36:14 | checkout | Mengirim SMS ke Budi Santoso
INFO | 2025-12-15 21:36:14 | checkout | CHECKOUT BERHASIL! Status Order: paid
INFO | 2025-12-15 21:36:14 | checkout | PROSES CHECKOUT: Citra Dewi | Total: Rp 250,000 | Metode: E-Wallet
INFO | 2025-12-15 21:36:14 | checkout | Memproses Pembayaran (E-Wallet)
INFO | 2025-12-15 21:36:14 | checkout | Memproses E-Wallet (request ke provider)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim Notifikasi (1 notifier)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim SMS ke Citra Dewi
INFO | 2025-12-15 21:36:14 | checkout | CHECKOUT BERHASIL! Status Order: paid
INFO | 2025-12-15 21:36:14 | checkout | PROSES CHECKOUT: Dani Wijaya | Total: Rp 100,000 | Metode: QRIS
INFO | 2025-12-15 21:36:14 | checkout | Memproses Pembayaran (QRIS)
INFO | 2025-12-15 21:36:14 | checkout | Memproses QRIS (QR Code, scan, verifikasi)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim Notifikasi (2 notifier)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim email konfirmasi ke Dani Wijaya
INFO | 2025-12-15 21:36:14 | checkout | Mengirim SMS ke Dani Wijaya
INFO | 2025-12-15 21:36:14 | checkout | CHECKOUT BERHASIL! Status Order: paid
INFO | 2025-12-15 21:36:14 | checkout | PROSES CHECKOUT: Eva Marlina | Total: Rp 350,000 | Metode: Credit Card
INFO | 2025-12-15 21:36:14 | checkout | Memproses Pembayaran (Credit Card)
INFO | 2025-12-15 21:36:14 | checkout | Memproses Kartu Kredit (gateway, CVV, verifikasi)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim Notifikasi (3 notifier)
INFO | 2025-12-15 21:36:14 | checkout | Mengirim email konfirmasi ke Eva Marlina
INFO | 2025-12-15 21:36:14 | checkout | Mengirim SMS ke Eva Marlina
INFO | 2025-12-15 21:36:14 | checkout | Mengirim WhatsApp ke Eva Marlina
INFO | 2025-12-15 21:36:14 | checkout | CHECKOUT BERHASIL! Status Order: paid

e. Screenshoot history commit



f. Refleksi singkat:

- Docstring memperjelas kontrak dan maksud setiap class/method sehingga memudahkan kolaborasi dan onboarding.
- Logging menggantikan print untuk observabilitas; level INFO/WARNING membantu memisahkan informasi normal dan anomali saat debugging.
- Kombinasi Docstring + Logging meningkatkan traceability serta mempercepat identifikasi masalah di pipeline dan saat integrasi tim.