

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)

PRAKTIKUM 13



2411102441052

Angga Maulana Saputra

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

Link Github:

a. Screenshot Output CLI

```

○ >_bash ➜ latihan_mandiri ➜ main ≈ ?4 ~1 ➜ 327ms
>> python main_app.py
REPOSITORY - INFO - ProductRepository initialized with 3 products.
MAIN_APP - INFO - POS Application Initialized.

Menu:
1. Tampilkan Produk
2. Tambahkan Keranjang
3. Checkout
4. Keluar
Pilih opsi (1-4): ■

○ >_bash ➜ latihan_mandiri ➜ main ≈ ?4 ~1 ➜ 327ms
>> python main_app.py
Pilih opsi (1-4): 1
MAIN_APP - INFO -
--- DAFTAR PRODUK ---
MAIN_APP - INFO - [P001] Laptop Gaming - Rp150,000,000
MAIN_APP - INFO - [P002] Mouse Wireless - Rp25,000,000
MAIN_APP - INFO - [P003] Keyboard Mech - Rp8,000,000

Menu:
1. Tampilkan Produk
2. Tambahkan Keranjang
3. Checkout
4. Keluar
Pilih opsi (1-4): ■

○ >_bash ➜ latihan_mandiri ➜ main ≈ ?4 ~1 ➜ 327ms
>> python main_app.py
1. Tampilkan Produk
2. Tambahkan Keranjang
3. Checkout
4. Keluar
Pilih opsi (1-4): 2
Masukkan ID Produk: p003
Jumlah (default 1): 1
SERVICES - INFO - Added 1x Keyboard Mech to cart.

Menu:
3. Checkout
4. Keluar
Pilih opsi (1-4): 3
MAIN_APP - INFO -
Total Belanja: Rp8,000,000
SERVICES - INFO - Memproses pembayaran DEBIT CARD sejumlah: Rp8,000,000
SERVICES - INFO - Verifikasi chip/PIN, otorisasi bank...
MAIN_APP - INFO - TRANSAKSI BERHASIL.
MAIN_APP - INFO -
--- STRUK PEMBELIAN ---
MAIN_APP - INFO - Keyboard Mech x1 = Rp8,000,000
MAIN_APP - INFO - -----
MAIN_APP - INFO - TOTAL AKHIR Rp8,000,000
MAIN_APP - INFO - -----

```

```
Menu:  
1. Tampilkan Produk  
2. Tambahkan Keranjang  
3. Checkout  
4. Keluar  
Pilih opsi (1-4): 4  
MAIN_APP - INFO - Aplikasi dihentikan.
```

- b. Refleksi: Dependency Injection (DI) membuat PosApp modular dan fleksibel karena PosApp bergantung pada abstraksi (IPaymentProcessor/INotificationService), bukan implementasi tertentu. Akibatnya, metode pembayaran/notifikasi baru (mis. Debit/QRIS/WhatsApp) bisa ditambahkan cukup dengan membuat class baru dan meng-inject-nya di main tanpa mengubah PosApp, sehingga patuh OCP/DIP dan meminimalkan risiko regresi. DI juga meningkatkan testability (bisa inject mock/fake), memudahkan maintenance, dan menyiapkan pondasi pengembangan lanjut (diskon, pajak, logging terstruktur, DB, atau migrasi ke REST/GUI) hanya dengan mengganti komposisi di composition root.