Nama : Siti Hajar Kasman

Npm : 0735221159

Kelas : 5IF3

Mata Kuliah : Pemrograman Mobile

1. Impelementasi Source Code

class Product {

String id;

String name;

double price;

int stock;

Product({required this.id, required this.name, required this.price, required this.stock});

void updateStock(int quantity) {

if (quantity > stock) {

throw Exception("Stok tidak cukup");

}

stock -= quantity;

}

String getDetails() {

return 'Product ID: $id\nName: $name\nPrice: \$${price.toStringAsFixed(2)}\nStock: $stock';

}

}

class User {

String id;

String name;

List<String> purchaseHistory;

User({required this.id, required this.name}) : purchaseHistory = [];

Future<void> purchaseProduct(Product product, int quantity) async {

try {

await Future.delayed(Duration(seconds: 2)); // Simulasi pembelian asinkron

product.updateStock(quantity);

purchaseHistory.add('Purchased ${product.name} x$quantity');

print('Pembelian berhasil!');

} catch (e) {

print('Error: $e');

}

}

void showPurchaseHistory() {

print('Riwayat Pembelian ${name}:');

for (var item in purchaseHistory) {

print('- $item');

}

}

}

class EcommerceSystem {

List<Product> products = [];

List<User> users = [];

void addProduct(Product product) {

products.add(product);

}

void addUser(User user) {

users.add(user);

}

void listProducts() {

if (products.isEmpty) {

print('Tidak ada produk tersedia.');

return;

}

for (var product in products) {

print(product.getDetails());

}

}

List<Product> searchProductByName(String name) {

return products.where((product) => product.name.toLowerCase().contains(name.toLowerCase())).toList();

}

List<Product> searchProductByPrice(double minPrice, double maxPrice) {

return products.where((product) => product.price >= minPrice && product.price <= maxPrice).toList();

}

}

void main() async {

var ecommerce = EcommerceSystem();

var product1 = Product(id: 'P001', name: 'Laptop', price: 800.00, stock: 10);

var product2 = Product(id: 'P002', name: 'Smartphone', price: 500.00, stock: 20);

ecommerce.addProduct(product1);

ecommerce.addProduct(product2);

var user1 = User(id: 'U001', name: 'John Doe');

ecommerce.addUser(user1);

print('Daftar Produk:');

ecommerce.listProducts();

print('');

print('John mencoba membeli Laptop (2 unit):');

await user1.purchaseProduct(product1, 2);

print('');

user1.showPurchaseHistory();

print('');

print('Mencari produk dengan nama "Smartphone":');

var searchResults = ecommerce.searchProductByName("Smartphone");

searchResults.forEach((product) {

print(product.getDetails());

});

print('');

print('Mencari produk dengan harga antara \$400 dan \$800:');

var priceSearchResults = ecommerce.searchProductByPrice(400.00, 800.00);

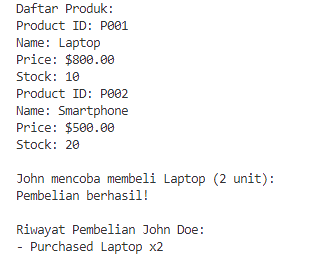
priceSearchResults.forEach((product) {

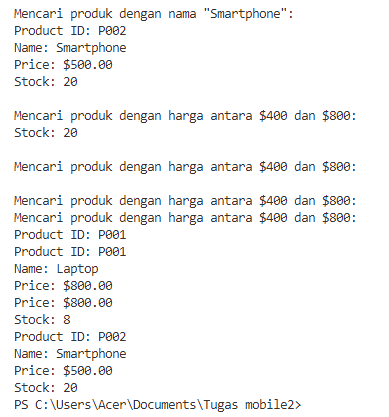
print(product.getDetails());

});

}

1. Hasil Output dari program diatas:





1. Referensi yang dipakai oleh program diatas.

* <https://dart.dev/guides/libraries/futures-error-handling>
* <https://dart.dev/guides/language/language-tour#exceptions>
* <https://dart.dev/guides/libraries/library-tour#collections>