

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

Lembar Kerja Peserta Didik (LKPD)

Petunjuk Pengerjaan

Isilah LKPD ini sesuai dengan tahapan pada sintaks Scientific Computational Learning (SCL). Jawaban Anda akan menjadi bagian dari dokumentasi proyek yang dinilai.

1. Identifikasi Masalah Utama

Lakukan identifikasi masalah utama dari studi kasus yang diberikan! Fokuskan apa yang menjadi input dan output dari program yang akan dibuat! Kemudian tuliskan di bawah ini

Input :

Data produk roti (nama, kode, bahan baku, biaya produksi, harga jual)

Jenis produk roti yang dipilih (roti manis, croissant, butter cookies, muffin)

Jumlah pcs roti yang akan diproduksi

Output :

Informasi semua produk

Estimasi profit dari produksi roti

Simulasi proses produksi roti (pengadonan, pengembangan, pemanggangan, topping)

Dalam hal ini, output yang kamu tuliskan di atas adalah masalah utama dari proyek

2. Identifikasi Submasalah

Berdasarkan identifikasi output yang diperoleh sebelumnya, identifikasi apa yang menjadi submasalah. Submasalah bisa teridentifikasi melalui fitur-fitur yang ada dari kebutuhan program

Submasalah 1 : *Menyimpan dan menampilkan data produk roti.*

Submasalah 2 : *Menghitung estimasi profit dan jumlah dari produksi tertentu.*

Submasalah 3 : *Menjalankan simulasi proses produksi tiap jenis roti sesuai urutan proses produksi masing-masing.*

Submasalah 4 : *Mengelompokkan produk berdasarkan jenis dengan atribut dan method yang relevan.*

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

3. Identifikasi Pola dalam Masalah

Berdasarkan masalah utama atau submasalah yang sudah teridentifikasi, identifikasikan pula obyek apa saja yang ada di dalamnya.

Tuliskan obyek utama apa saja yang ada di dalam masalah!

ProdukRoti

Berdasarkan obyek utama tersebut, adakah sub-obyeknya? Jika ada tuliskan!

RotiManis, Croissant, ButterCookies, Muffin

Setelah kalian menentukan sub-obyeknya, amati adakah kesamaan atribut atau method untuk setiap sub-obyeknya? Jika ada, tuliskan!

Atribut: *nama_produk, kode_produk, daftar_bahan, biaya_produksi, harga_jual*

Method: *tampil_info(), hitung_profit(), pengadonan(), pemanggangan(), topping()*

4. Identifikasi Abstraksi Submasalah

Berdasarkan adanya kesamaan atribut atau method untuk setiap sub-obyeknya, jadikan obyek utama tersebut sebagai abstract class sekaligus superclass dan sub-obyeknya sebagai subclassnya. Tuliskan apa yang menjadi abstract superclass dan subclassnya

Abstract Superclass : *ProdukRoti*

Subclasses : *RotiManis, Croissant, ButterCookies, Muffin*

Selanjutnya tentukan atribut dan method masing-masing abstract superclass dan subclasses

Atribut Abstract Superclass:

Nama_produk

Kode_produk

Daftar_bahan

Biaya_produksi

Harga_jual

Method Abstract Superclass:

tampil_info()

hitung_profit()

pengadonan()

pemanggangan()

topping()

Method Khusus Subclass:

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

pengembangan() (khusus RotiManis, Croissant, Muffin)

5. Merumuskan Alternatif Solusi Struktur Class

Berdasarkan hasil langkah no 4 sebelumnya, lakukan analisis terhadap superclass dan subclassesnya, apakah memungkinkan dibuat sebuah interface untuk memfasilitasi subclass yang memiliki method khusus yang tidak dimiliki subclass lainnya.

Jika kalian menemukan potensi untuk membuat interface, maka tuliskan beserta method abstraknya!

Nama Interface: *ProsesProduksi*

Method Abstrak:

pengadonan()

pengembangan()

pemanggangan()

topping()

6. Memutuskan Rancangan Solusi

Berdasarkan hasil dari langkah 4 dan 5, selanjutnya lakukan analisis apakah sudah memenuhi prinsip SOLID! Jika belum, maka lakukan restrukturisasi terhadap rancangan solusi.

Tuliskan rancangan solusi setelah melalui proses analisis

Rancangan sudah memenuhi prinsip SOLID karena:

Setiap class punya tanggung jawab tunggal

Open-Closed Principle dipenuhi (bisa ditambah jenis roti baru tanpa mengubah class lama)

Interface Segregation: proses produksi standar ada di interface

Tidak ada pelanggaran prinsip OOP

7. Merancang Struktur Class (lampirkan diagram class jika perlu)

Berdasarkan hasil dari langkah 6, buatlah struktur class (superclass, subclass) dan interfaces secara lengkap beserta atribut dan nama methodnya. Tuliskan di bawah ini

<https://drive.google.com/file/d/1kbiJEUHVLY4oN1FCzCTxgtJ3sWUiNjn-/view?usp=sharing>

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

8. Merancang Algoritma

Buatlah rancangan algoritma yang menggambarkan alur proses (flow process) dari program. Kaitkan antara apa yang menjadi input, output, dan fitur yang merupakan hasil dari langkah no 2.

Tampilkan Menu:

Tambah Produk / Tampilkan Semua Produk / Kalkulator Profit / Simulasi Produksi

Jika tambah produk:

Input data → Simpan ke list produk

Jika tampilkan semua:

Loop semua objek produk → tampil_info()

Jika kalkulator profit:

Pilih produk → Masukkan jumlah pcs → hitung_profit() → tampilkan estimasi

Jika simulasi produksi:

Pilih produk → Jalankan proses produksi sesuai urutan method

9. Menuliskan Kode Program (terlampir)

Berdasarkan hasil langkah no 7 dan 8, tuliskan kode programnya menggunakan Python.

Kode Python:

```
from abc import ABC, abstractmethod

class Baking(ABC):
    @abstractmethod
    def bake(self): pass

class Packaging(ABC):
    @abstractmethod
    def pack(self): pass

class Labeling(ABC):
```

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

```
@abstractmethod
def label(self): pass

class ProdukRoti(ABC):
    def __init__(self, nama, kode, bahan, biaya, harga):
        self.nama = nama
        self.kode = kode
        self.bahan = bahan
        self.biaya = biaya
        self.harga = harga

    def tampil_info(self):
        print(f"\n=== {self.nama.upper()} ({self.kode}) ===")
        print("Bahan Baku:")
        for b, j in self.bahan.items():
            print(f"- {b}: {j}")
        print(f"Biaya Produksi: Rp{self.biaya}")
        print(f"Harga Jual: Rp{self.harga}")

    def hitung_profit(self, jumlah):
        return (self.harga - self.biaya) * jumlah

    @abstractmethod
    def pengadonan(self): pass

    def pengembangan(self):
        print("Tidak perlu proses pengembangan.")

    @abstractmethod
    def pemanggangan(self): pass

    @abstractmethod
    def topping(self): pass

class RotiManis(ProdukRoti, Baking, Packaging, Labeling):
    def pengadonan(self): print("Mengaduk adonan roti manis...")
```

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

```
def pengembangan(self): print("Proofing adonan roti manis...")
def pemanggangan(self): print("Memanggang roti manis...")
def topping(self): print("Menambahkan topping manis...")
def bake(self): print("Baking roti manis...")
def pack(self): print("Mengemas roti manis...")
def label(self): print("Memberi label Roti Manis.")

class Croissant(ProdukRoti, Baking, Packaging, Labeling):
    def pengadonan(self): print("Mengaduk adonan croissant...")
    def pengembangan(self): print("Melipat dan fermentasi
croissant...")
    def pemanggangan(self): print("Memanggang croissant...")
    def topping(self): print("Menambahkan butter glaze...")
    def bake(self): print("Baking croissant...")
    def pack(self): print("Mengemas croissant...")
    def label(self): print("Memberi label Croissant.")

class ButterCookies(ProdukRoti, Baking, Packaging, Labeling):
    def pengadonan(self): print("Mengaduk adonan butter
cookies...")
    def pemanggangan(self): print("Memanggang butter cookies...")
    def topping(self): print("Menambahkan taburan gula...")
    def bake(self): print("Baking butter cookies...")
    def pack(self): print("Mengemas butter cookies...")
    def label(self): print("Memberi label Butter Cookies.")

class Muffin(ProdukRoti, Baking, Packaging, Labeling):
    def pengadonan(self): print("Mengaduk adonan muffin...")
    def pengembangan(self): print("Proofing adonan muffin...")
    def pemanggangan(self): print("Memanggang muffin...")
    def topping(self): print("Menambahkan topping buah...")
    def bake(self): print("Baking muffin...")
    def pack(self): print("Mengemas muffin...")
    def label(self): print("Memberi label Muffin.")

produk_list = []
```

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

```
def menu():
    while True:
        print("\n===== HANARI BAKERY SYSTEM =====")
        print("1. Tambah Produk")
        print("2. Tampilkan Semua Produk")
        print("3. Kalkulator Estimasi Profit")
        print("4. Simulasi Produksi")
        print("0. Keluar")

        pilih = input("Pilih menu: ")
        if pilih == "1": tambah_produk()
        elif pilih == "2": tampilkan_produk()
        elif pilih == "3": kalkulator_profit()
        elif pilih == "4": simulasi_produksi()
        elif pilih == "0": print("Terima kasih!"); break
        else: print("Pilihan tidak valid.")

def tambah_produk():
    print("\n-- Tambah Produk --")
    nama = input("Nama produk: ")
    kode = input("Kode produk: ")
    jenis = input("Jenis produk (roti manis / croissant / butter cookies / muffin): ").lower()
    bahan = {}
    while True:
        bhn = input("Nama bahan (ketik 'done' jika selesai): ")
        if bhn == "done": break
        jumlah = input(f"Jumlah {bhn}: ")
        bahan[bhn] = jumlah
    biaya = int(input("Biaya produksi per-n pcs: "))
    harga = int(input("Harga jual per-n pcs: "))

    produk = None
    if jenis == "roti manis":
        produk = RotiManis(nama, kode, bahan, biaya, harga)
    elif jenis == "croissant":
```

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

```
        produk = Croissant(nama, kode, bahan, biaya, harga)
    elif jenis == "butter cookies":
        produk = ButterCookies(nama, kode, bahan, biaya, harga)
    elif jenis == "muffin":
        produk = Muffin(nama, kode, bahan, biaya, harga)
    else:
        print("Jenis tidak dikenali.")
        return

    produk_list.append(produk)
    print("Produk berhasil ditambahkan!")

def tampilkan_produk():
    if not produk_list:
        print("Belum ada produk.")
    for p in produk_list:
        p.tampil_info()

def kalkulator_profit():
    tampilkan_produk()
    kode = input("\nMasukkan kode produk untuk estimasi: ")
    jumlah = int(input("Jumlah pcs yang akan diproduksi: "))
    for p in produk_list:
        if p.kode == kode:
            profit = p.hitung_profit(jumlah)
            print(f"Estimasi profit: Rp{profit}")
            return
    print("Produk tidak ditemukan.")

def simulasi_produksi():
    tampilkan_produk()
    kode = input("\nMasukkan kode produk untuk simulasi: ")
    for p in produk_list:
        if p.kode == kode:
            print("\n-- Simulasi Produksi --")
            p.pengadonan()
```


Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

```
        p.pengembangan()  
        p.pemangngangan()  
        p.topping()  
        p.bake()  
        p.pack()  
        p.label()  
        return  
    print("Produk tidak ditemukan.")  
  
if __name__ == "__main__":  
    menu()
```

10. Melakukan Debugging Kode Program

Lakukan debugging pada kode program apabila terdapat kesalahan sintaks

Sudah dilakukan, tidak ada error sintaks.

11. Menguji Kelayakan Program

Lakukan uji coba kode program dengan menjalankannya, kemudian masukkan beberapa sampel data input yang diberikan, kemudian amati outputnya. Apabila masih terdapat kesalahan output, maka ulangi proses debugging (langkah no 10)

Program berhasil menampilkan informasi produk, menghitung estimasi profit, dan menjalankan simulasi proses produksi dengan baik.

12. Mengevaluasi Keefektifan Solusi

Berdasarkan hasil uji coba dan melihat kepraktisan atau efisiensi dari penulisan kode program yang dibuat, apakah kamu merasakan sudah baik? Atau masih ada kekurangan? Lakukan analisis, terutama pada potensi pengembangan selanjutnya, misalkan suatu saat ada penambahan posisi karyawan baru, misalkan Operator, atau ada penambahan tunjangan baru untuk karyawan, misalkan tunjangan kesehatan. Apakah jika dengan penambahan-penambahan tersebut, kode program kalian sudah memungkinkan hal tersebut dengan berpegang pada prinsip SOLID?

Jika belum, maka tuliskan apa yang menjadi kelemahan dari kode program kalian

Sudah memenuhi prinsip SOLID.

Kode mudah dikembangkan untuk tambahan produk baru.

Kekurangan:

Nama Anggota Kelompok:

1. Herlina Dwi Septiana (K3524012)
2. Ade Amanah Putri (K3524018)
3. Cahaya Fitri Ramadhani (K3524046)

Belum ada GUI (masih CLI).

Belum ada penyimpanan permanen (belum ke database).

13. Menyusun Rencana Potensi Pengembangan

Berdasarkan hasil dari evaluasi no 12, maka tuliskan rencana perbaikan atau pengembangan kode program yang sudah dibuat ke depannya!

Menambahkan GUI menggunakan Tkinter / PyQt.

Menyimpan data produk ke database SQLite/MySQL.

Menambah fitur manajemen karyawan atau manajemen penjualan.

14. Link Repository GitHub

Berdasarkan hasil pekerjaan kelompok masing-masing, tuliskan Link Repository Github kelompok anda

<https://github.com/heycahayaa/Kelompok-4-UAS-PBO>