

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

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 : Jenis, nama, kode, bahan, biaya, harga, Produk, Jumlah pcs

Output : Konfirmasi tambah, Daftar Produk, Estimasi profit, Proses produksi

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 : Manajemen pendataan produk

Submasalah 2 : Penghitungan profit

Submasalah 3 : Pembuatan menu

Submasalah 4 : Produk keluar

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!

Obyek Utama : Hanari Bakery

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

Sub-obyek : Roti Manis, Croissant, Butter Cookies, Muffin

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

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

Kesamaan Atribut : nama, kode, bahan, biaya produksi, harga jual

Kesamaan Method : pengadonan(), pemanggangan(), estimasi_profit()

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 : Hanari Bakery

Subclasses : Roti Manis, Croissant, Butter Cookies, Muffin

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

Atribut Abstract superclass : nama_produk, kode_produk, bahan_baku, biaya_produksi, harga_jual

Atribut Abstract subclasses : jenis_topping, waktu_pengembangan

Method Abstract superclass : tampilkan_info(), estimasi_profit(jumlah_pcs), proses_pengadonan(), proses_pemanggangan()

Method Abstract subclasses : proses_pengembangan(), proses_topping(), proses_pengadonan(), proses_pemanggangan()

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!

Interface : IBaking, IPackaging, ILabeling

Method Abstract : Baking(), Packaging(), Labeling()

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

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

Belum memenuhi prinsip SOLID karena **Dependency Inversion Principle (DIP)** belum diterapkan.

- Program masih langsung bergantung pada input/output di menu CLI. Belum ada pemisahan antara logic program dan UI. Bisa diperbaiki dengan memisahkan lapisan logika dan antarmuka input.

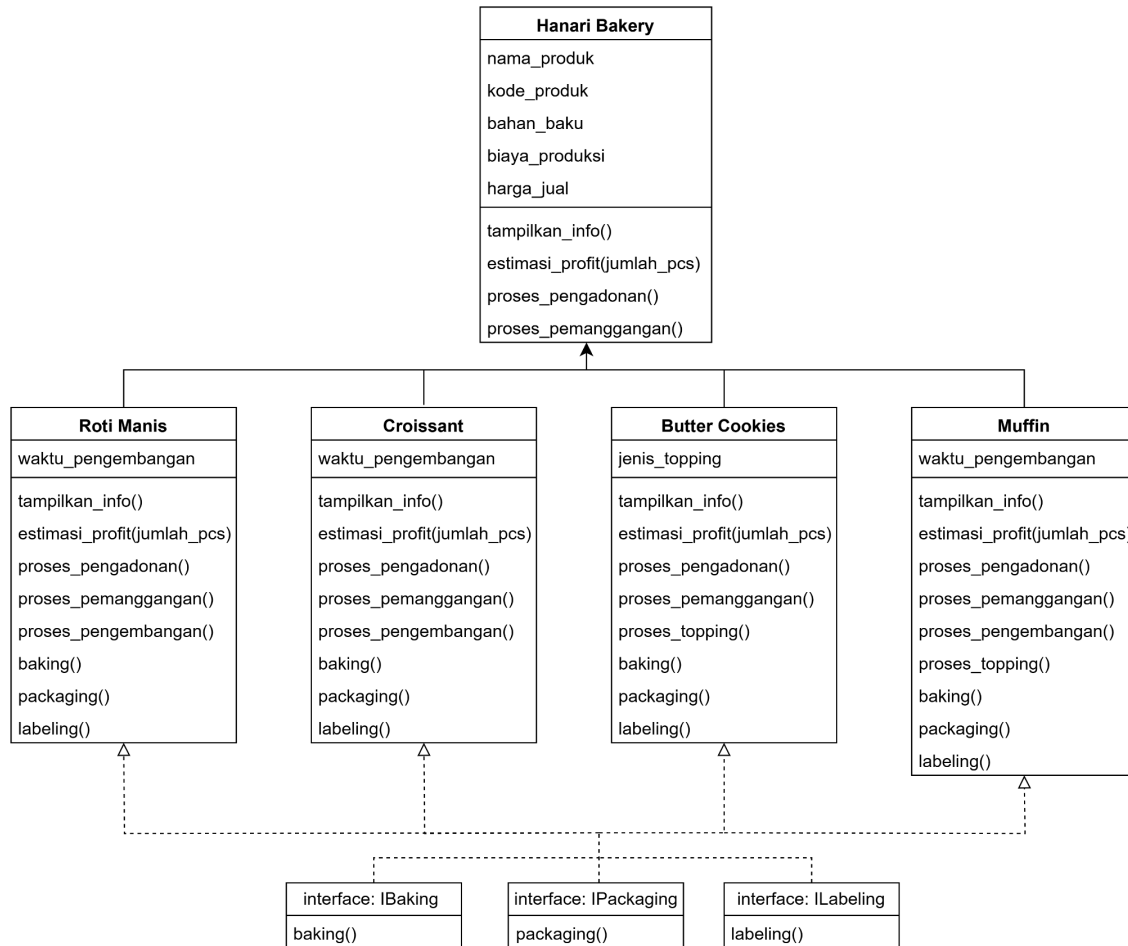
Rancangan Solusi Akhir (Setelah Analisis):

1. Gunakan struktur OOP yang sudah ada (abstract class + subclass + interface)
2. Pisahkan logika menu ke dalam modul atau fungsi terpisah:
 - Contoh: input_handler.py untuk semua input
 - produk_handler.py untuk manajemen produk
 - simulasi.py untuk logika proses produksi
3. Tambahkan validasi input untuk mencegah data kosong atau kode duplikat
4. (Pengembangan Lanjut) Tambahkan penyimpanan ke file (JSON atau CSV) agar data tidak hilang saat program ditutup.
5. (Opsional) Buat class Karyawan jika nanti ingin menambah sistem gaji atau operator produksi.

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 in

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

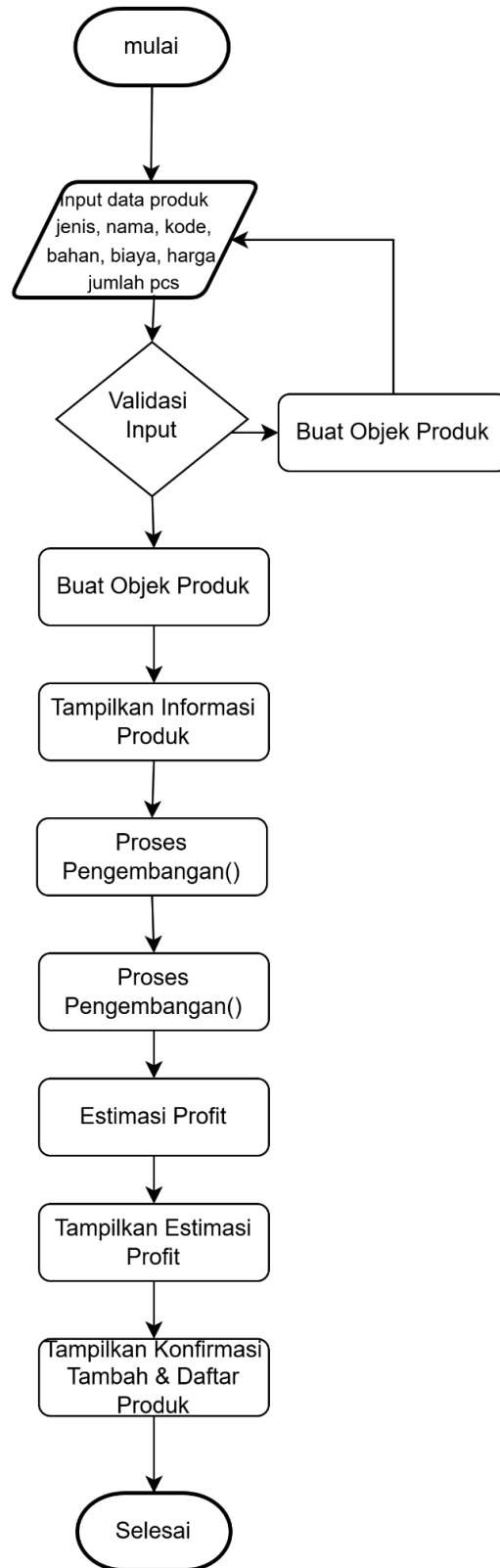


8. Merancang Algoritma

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.

Kelompok 7

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)



1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

9. Menuliskan Kode Program (terlampir)

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

```
from abc import ABC, abstractmethod

# Interfaces
class IBaking(ABC):
    @abstractmethod
    def bake(self): pass

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

class ILabeling(ABC):
    @abstractmethod
    def label(self): pass

# Abstract Class
class HanariBakery(ABC):
    def __init__(self, nama_produk, kode_produk, bahan_baku,
biaya_produksi, harga_jual, jenis_topping, waktu_pengembangan):
        self.nama_produk = nama_produk
        self.kode_produk = kode_produk
        self.bahan_baku = bahan_baku # dictionary
        self.biaya_produksi = biaya_produksi
        self.harga_jual = harga_jual
        self.jenis_topping = jenis_topping
        self.waktu_pengembangan = waktu_pengembangan

    def tampilkan_info(self):
        print(f"\n=== {self.nama_produk.upper()}
({self.kode_produk}) ===")
        print("Bahan Baku:")
        for b, j in self.bahan_baku.items():
            print(f"- {b}: {j}")
        print(f"Biaya Produksi: Rp{self.biaya_produksi}")
        print(f"Harga Jual: Rp{self.harga_jual}")
```

Kelompok 7

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

```
        print(f"Topping: {self.jenis_topping}")
        print(f"Waktu Pengembangan: {self.waktu_pengembangan}
menit")

    def estimasi_profit(self, jumlah_pcs):
        return (self.harga_jual - self.biaya_produksi) *
jumlah_pcs

    @abstractmethod
    def proses_pengadonan(self): pass

    @abstractmethod
    def proses_pemanggangan(self): pass

    @abstractmethod
    def proses_topping(self): pass

    def proses_pengembangan(self):
        print("Tidak memerlukan proses pengembangan.")

# Subclass
class RotiManis(HanariBakery, IBaking, IPackaging, ILabeling):
    def proses_pengadonan(self): print("Mengaduk adonan roti
manis...")
    def proses_pengembangan(self): print("Proofing roti manis...")
    def proses_pemanggangan(self): print("Memanggang roti
manis...")
    def proses_topping(self): print("Menambahkan topping
manis...")
    def bake(self): print("Baking roti manis...")
    def pack(self): print("Mengemas roti manis...")
    def label(self): print("Label: Roti Manis")

class Croissant(HanariBakery, IBaking, IPackaging, ILabeling):
    def proses_pengadonan(self): print("Mengaduk adonan
croissant...")
    def proses_pengembangan(self): print("Melipat dan proofing
```

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

```
croissant...")
    def proses_pemanggangan(self): print("Memanggang
croissant...")
    def proses_topping(self): print("Menambahkan butter glaze...")
    def bake(self): print("Baking croissant...")
    def pack(self): print("Mengemas croissant...")
    def label(self): print("Label: Croissant")

class ButterCookies(HanariBakery, IBaking, IPackaging, ILabeling):
    def proses_pengadonan(self): print("Mengaduk adonan butter
cookies...")
    def proses_pemanggangan(self): print("Memanggang butter
cookies...")
    def proses_topping(self): print("Menambahkan gula tabur...")
    def bake(self): print("Baking butter cookies...")
    def pack(self): print("Mengemas butter cookies...")
    def label(self): print("Label: Butter Cookies")

class Muffin(HanariBakery, IBaking, IPackaging, ILabeling):
    def proses_pengadonan(self): print("Mengaduk adonan
muffin...")
    def proses_pengembangan(self): print("Proofing muffin...")
    def proses_pemanggangan(self): print("Memanggang muffin...")
    def proses_topping(self): print("Menambahkan topping buah...")
    def bake(self): print("Baking muffin...")
    def pack(self): print("Mengemas muffin...")
    def label(self): print("Label: Muffin")

# CLI & Data
produk_list = []

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


1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

```
print("4. Simulasi Produksi")
print("0. Keluar")
pilihan = input("Pilih menu: ")

if pilihan == "1": tambah_produk()
elif pilihan == "2": tampilkan_produk()
elif pilihan == "3": kalkulator_profit()
elif pilihan == "4": simulasi_produksi()
elif pilihan == "0": break
else: print("Pilihan tidak valid.")

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

    cls = {"roti manis": RotiManis, "croissant": Croissant,
"butter cookies": ButterCookies, "muffin": Muffin}
    if jenis in cls:
        produk = cls[jenis](nama, kode, bahan, biaya, harga,
topping, waktu)
        produk_list.append(produk)
        print("Produk berhasil ditambahkan!")
    else:
        print("Jenis tidak dikenali.")
```

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

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

def kalkulator_profit():
    kode = input("Masukkan kode produk: ")
    jumlah = int(input("Jumlah pcs: "))
    for p in produk_list:
        if p.kode_produk == kode:
            profit = p.estimasi_profit(jumlah)
            print(f"Estimasi profit: Rp{profit}")
            return
    print("Produk tidak ditemukan.")

def simulasi_produksi():
    kode = input("Masukkan kode produk: ")
    for p in produk_list:
        if p.kode_produk == kode:
            print("\n-- Simulasi Produksi --")
            p.proses_pengadonan()
            p.proses_pengembangan()
            p.proses_pemanggangan()
            p.proses_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

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

```
PROBLEMS OUTPUT DEBUG CONSOLE
```

```
PS C:\Users\HP\OneDrive\Desktop\PBO> & 'c:\Users\HP\AppData\Local\Programs\Python\Python39-64\Scripts\python.exe' -c "import sys; sys.path.append('C:\\Users\\HP\\AppData\\Local\\Programs\\Python\\Python39-64\\Scripts'); import debugpy; debugpy.listen(5846); debugpy.run(['PBO\\PBO2\\UAS.py'])"
```

```
===== HANARI BAKERY SYSTEM =====
```

```
1. Tambah Produk
```

```
2. Tampilkan Semua Produk
```

```
3. Kalkulator Estimasi Profit
```

```
4. Simulasi Produksi
```

```
0. Keluar
```

```
Pilih menu: █
```

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)

```
===== HANARI BAKERY SYSTEM =====
1. Tambah Produk
2. Tampilkan Semua Produk
3. Kalkulator Estimasi Profit
4. Simulasi Produksi
0. Keluar
Pilih menu: 1

-- Tambah Produk --
Nama produk: Cookies
Kode produk: 02
Jenis produk (roti manis / croissant / butter cookies / muffin): roti manis
Nama bahan (ketik 'done' jika selesai): done
Biaya produksi per-n pcs: 400000
Harga jual per-n pcs: 4000
Produk berhasil ditambahkan!
```

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

Kode sudah baik, alasannya:

- A. Menggunakan inheritance dengan benar untuk membedakan produk
- B. Menggunakan abstract class dan interface sesuai kebutuhan (baking, packing, labeling)

1. Athaya Alifia Zahra (K3524046)
2. Afifah Zahrah Muthmainnah (K3524082)
3. Nadhira Zulsyifa S. (K3524082)

C. Struktur sudah cukup rapi dan mudah di kembangkan

Kekurangan dan potensi pengembangan:

- A. Belum fleksibel untuk karyawan: Jika ingin menambahkan fitur seperti operator atau karyawan dengan tunjangan, saat ini sistem belum mendukung itu
- B. Belum menerapkan prinsip SOLID sepenuhnya:
 - a. *Single Responsibility*: Fungsi `tambah_produk` dan `simulasi_produksi` memegang beberapa tanggung jawab sekaligus (input, validasi, pembuatan objek)

13. Menyusun Rencana Potensi Pengembangan

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

- Pisahkan tanggung jawab fungsi: Refaktor seperti `tambah_produk()` agar hanya menangani satu tugas (input, validasi, pembuatan objek dipisah).
- Modularisasi program: Pecah kode ke beberapa file seperti [produk.py](#), [interface.py](#), dan [main.py](#) agar lebih terstruktur.
- Simpan data ke file: Tambahkan fitur simpan/muat produk ke file JSON atau CSV agar data tidak hilang saat program ditutup.
- Tambahkan class Karyawan: Untuk mendukung pengembangan sistem gaji/operator ke depannya.

14. Link Repository GitHub

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

https://github.com/athayaalifiaa/ProjectUAS_Kelompok_7_PBO