## A. IDENTITAS

| | |
|---|---|
| NIM | : 2403099 |
| Nama Lengkap | : AHMAD MAUALAN KUDUS |
| Kelas | : D3 TI2B |
| Program Studi | : D3 TEKNIK INFORMATIKA |
| Jurusan | : TEKNIK INFORMATIKA |

## B. DESKRIPSI MATA KULIAH

| | |
|---|---|
| Nama Matakuliah | : Pemograman Berorientasi Objek |
| Dosen Pengampu | : Fachrul Pralienka Bani Muhamad, M.Kom. |
| Pertemuan ke | : ~~1,2,3,4,5,6,7,8,9,10,11,12,13,14,~~15,16 |
| Materi | : Dasar Pemrograman Java |
| Minggu | : Sembilan |
| Ruangan | : Lab. Pemograman & R.501 |

## C. HASIL PRAKTIKUM

A. Layered Architecture Pattern
Pola arsitektur yang mengatur komponen-komponen aplikasi menjadi beberapa lapisan (layer) terpisah secara logis dengan menerapkan prinsip pemisahan tanggung jawab (separation of concerns)

- Presentation Layer
  Lapisan yang berinteraksi dengan pengguna.
  - Presentation Pattern
    1. Menampilkan data
    2. Menerima input
    3. Validasi input
- Business Logic / Application Layer
  Lapisan (inti) logika bisnis yang mendefinisikan apa yang dilakukan aplikasi.
  - Domain Logic Pattern
    1. Menerapkan aturan bisnis spesifik
    2. Manajemen transaksi
    3. Memastikan konsistensi data
- Persistence / Data Access Layer

Lapisan yang menjembatani abstraksi bisnis dari detail penyimpanan data.

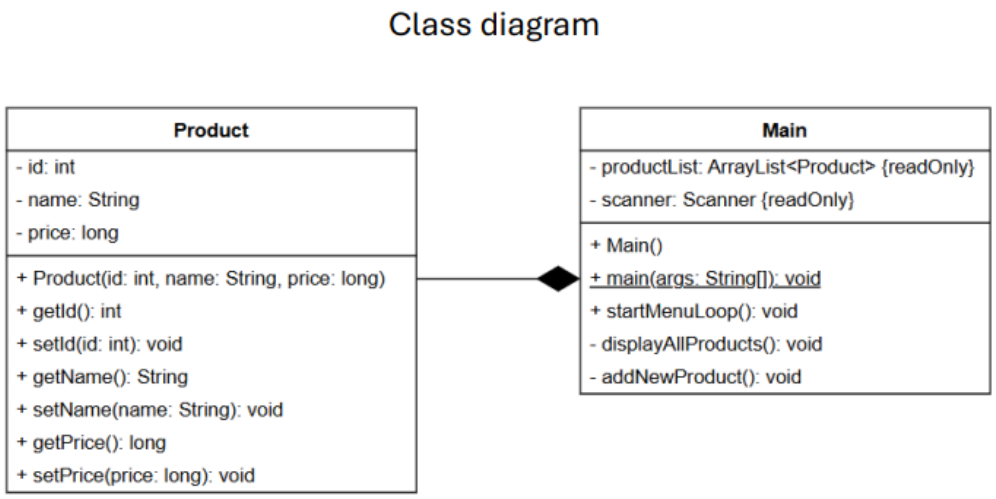
- Data Persistence Pattern
  1. Mengelola data (CRUD)
  2. Akses database (querying)
  3. Mengonversi data dari format database

- Database / Data Source Layer
  Lapisan fisik penyimpanan data (database server)

  - Pattern
    1. Penyimpanan data
    2. Mengelola izin akses database
    3. Server DB fisik (MySQL, PostgreSQL, atau lainnya)

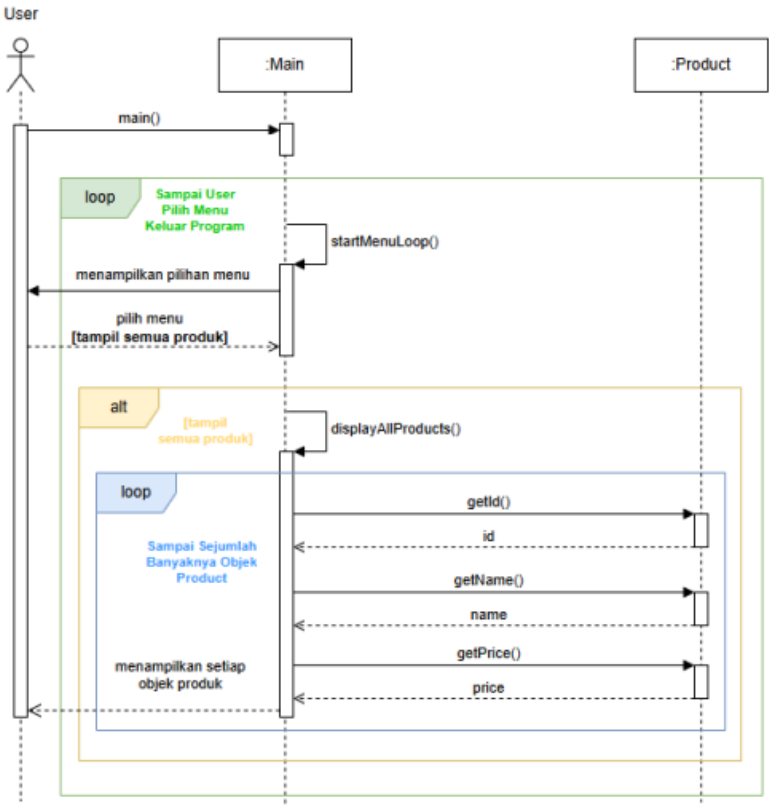
B. kode program TANPA PATTERN

1. Class Diagram

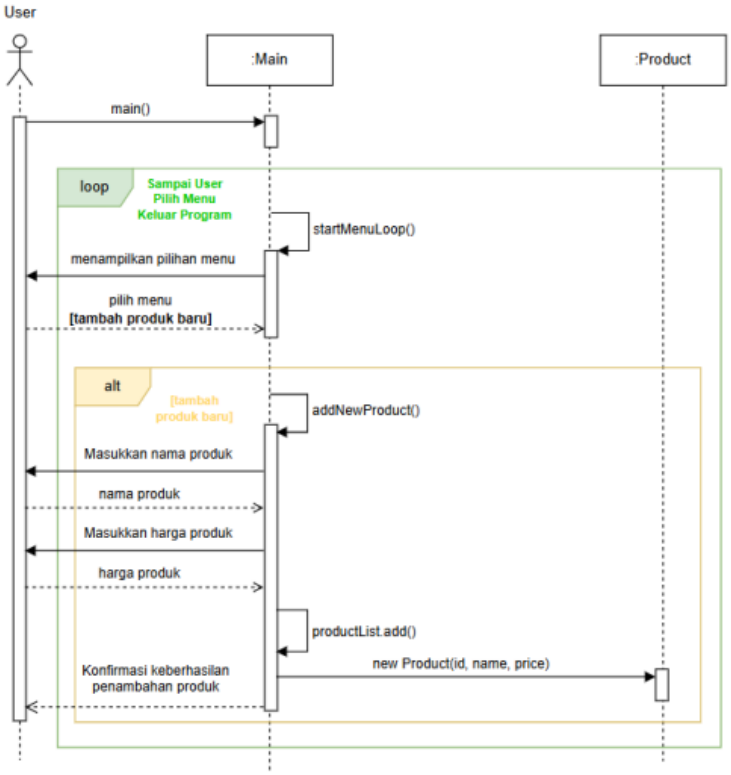



2. Sequence Diagram

- Menampilkan semua daftar produk

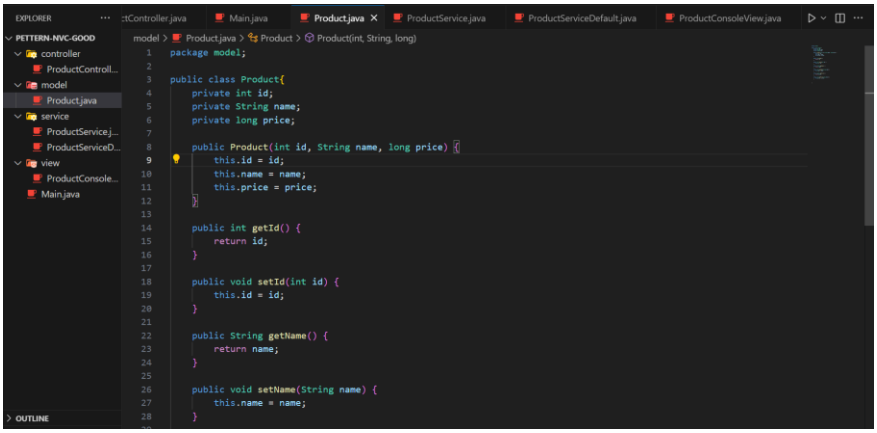



- Menambah produk baru




3. Kode program

   a. Product.java
   Buat folder Bernama “pettern-nvc-good”, di dalam folder tersebut buat folder “model” di dalam foldernya buat file “Product.java” isi kode programnya dibawah ini :



```java
package model;

public class Product{
    private int id;
    private String name;
    private long price;

    public Product(int id, String name, long price) {
        this.id = id;
        this.name = name;
        this.price = price;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
```

```
29
30      public double getPrice() {
31          return price;
32      }
33
34      public void setPrice(long price) {
35          this.price = price;
36      }
37
38
39
40  }
```

b. Main.java

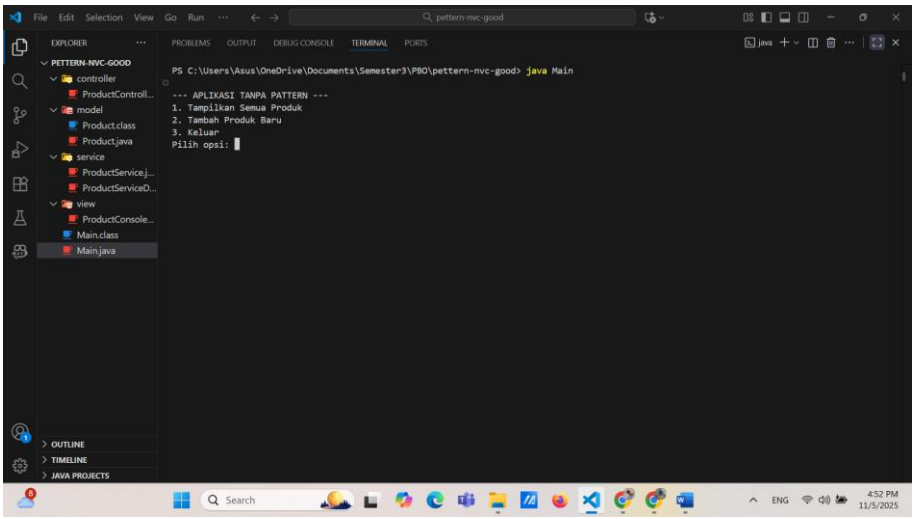
```
import java.util.ArrayList;
import java.util.Scanner;
import model.Product;

public class Main {

    private final ArrayList<Product> productList = new ArrayList<>();
    private final Scanner scanner = new Scanner(System.in);

    public Main() {
        // Data awal
        productList.add(new Product(id: 1, name: "Laptop ASUS", price: 9500000));
        productList.add(new Product(id: 2, name: "Monitor Dell", price: 2500000));
    }

    public void startMenuLoop() {
        boolean running = true;
        while (running) {
            System.out.println(x: "\n--- APLIKASI TANPA PATTERN ---");
            System.out.println(x: "1. Tampilkan Semua Produk");
            System.out.println(x: "2. Tambah Produk Baru");
            System.out.println(x: "3. Keluar");
            System.out.print(s: "Pilih opsi: ");
            try {
                int choice = Integer.parseInt(scanner.nextLine());
                switch (choice) {
                    case 1:
                        displayAllProducts();
                        break;
```

```
                case 2:
                    addNewProduct();
                    break;
                case 3:
                    running = false;
                    System.out.println(x: "Terima kasih telah menggunakan aplikasi!");
                    break;
                default:
                    System.out.println(x: "Opsi tidak valid.");
            }
        } catch (NumberFormatException e) {
            System.out.println(x: "Input tidak valid. Masukkan angka.");
        }
    }
    scanner.close();
}

private void displayAllProducts() {
    System.out.println(x: "\n--- Daftar Produk ---");
    if (productList.isEmpty()) {
        System.out.println(x: "Tidak ada produk tersedia.");
    } else {
        for (Product product : productList) {
            System.out.println(product.getId() + " - " + product.getName() + " Rp. " + product.getPrice());
        }
    }
}
```

```
private void addNewProduct() {
    System.out.print(s: "Masukkan Nama Produk: ");
    String name = scanner.nextLine();

    if (name.trim().isEmpty()) {
        System.out.println(x: "Error: Nama produk tidak boleh kosong.");
        return;
    }

    System.out.print(s: "Masukkan Harga Produk: ");
    String priceString = scanner.nextLine();
    try {
        long price = Long.parseLong(priceString);
        if (price <= 0) {
            throw new IllegalArgumentException(s: "Harga harus angka positif di atas nol!");
        }
        int newId = productList.size() + 1;
        productList.add(new Product(newId, name, price));
        System.out.println(x: "Produk berhasil ditambahkan!");
    } catch (NumberFormatException e) {
        System.out.println(x: "Error: Harga tidak valid. Masukkan angka.");
    } catch (IllegalArgumentException e) {
        System.out.println("Error: " + e.getMessage());
    }
}

Run | Debug
public static void main(String[] args) {
    Main app = new Main();
```

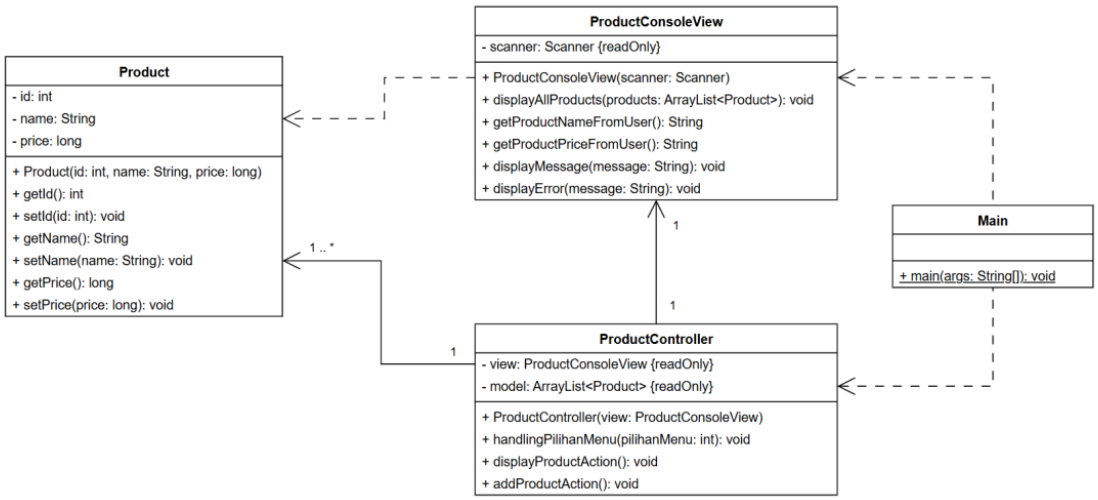
```
    Run | Debug
    public static void main(String[] args) {
        Main app = new Main();
        app.startMenuLoop();

    }
}
```

4. Compile & Run



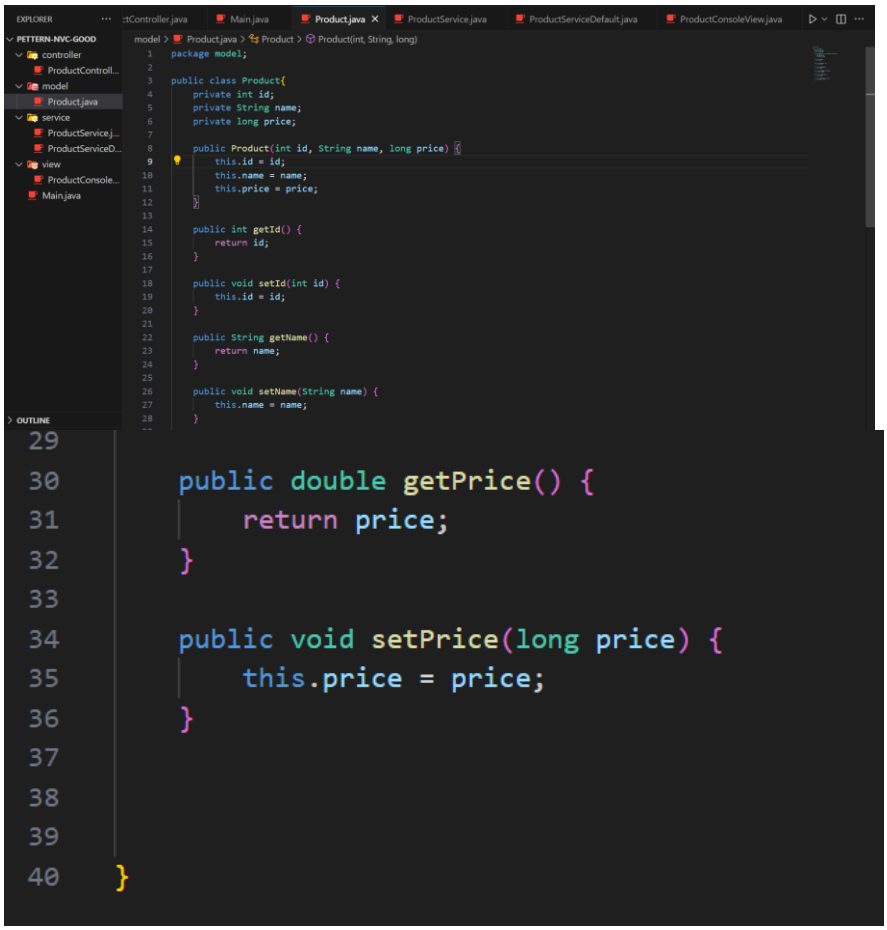

C. kode program MVC Pattern

1. Class Diagram




2. Kode program

a. Product.java



```java
package model;

public class Product{
    private int id;
    private String name;
    private long price;

    public Product(int id, String name, long price) {
        this.id = id;
        this.name = name;
        this.price = price;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(long price) {
        this.price = price;
    }
}
```

b. ProductConsoleView

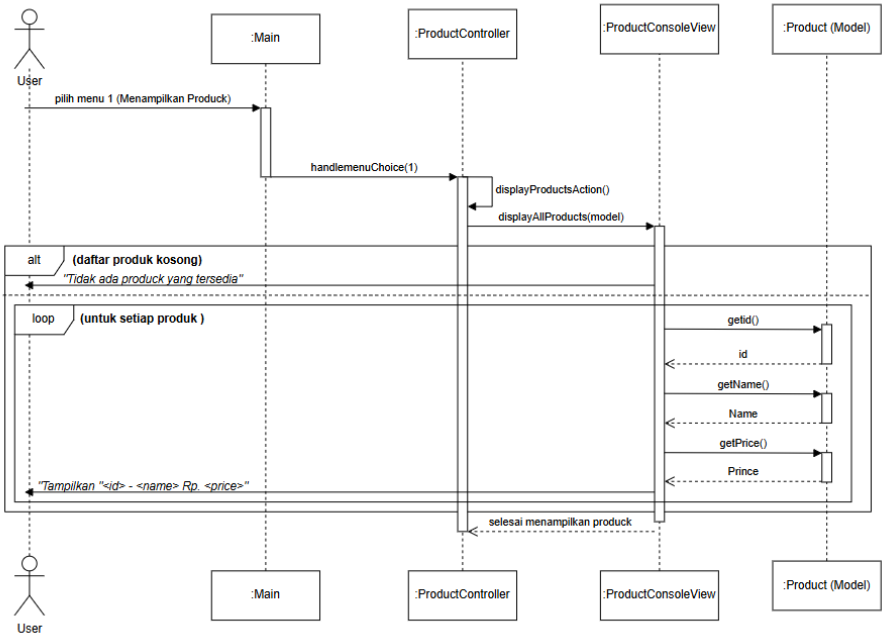
```java
package view;

import java.util.ArrayList;
import java.util.Scanner;
import model.Product;

public class ProductConsoleView {
    private final Scanner scanner;
    public ProductConsoleView(Scanner scanner) {
        this.scanner = scanner;
    }

    public void displayAllProducts(ArrayList<Product> products) {
        System.out.println("\n--- Daftar Produk ---");
        if (products.isEmpty()) {
            System.out.println("Tidak ada produk tersedia.");
        } else {
            for (Product product : products) {
                System.out.println(product.getId() + " - " + product.getName() + " Rp. " + product.getPrice());
            }
        }
    }

    public String getProductNameFromUser (){
        System.out.print("Masukkan Nama Produk: ");
        return scanner.nextLine();
    }
    public String getProductPriceFromUser (){
        System.out.print("Masukkan Harga Product: ");
        return scanner.nextLine();
```

```
    }
    public void displayMessage (String message){
        System.out.println("INFO: " + message);
    }
    public void displayError (String message){
        System.out.println("ERROR: " + message);
    }
}
```

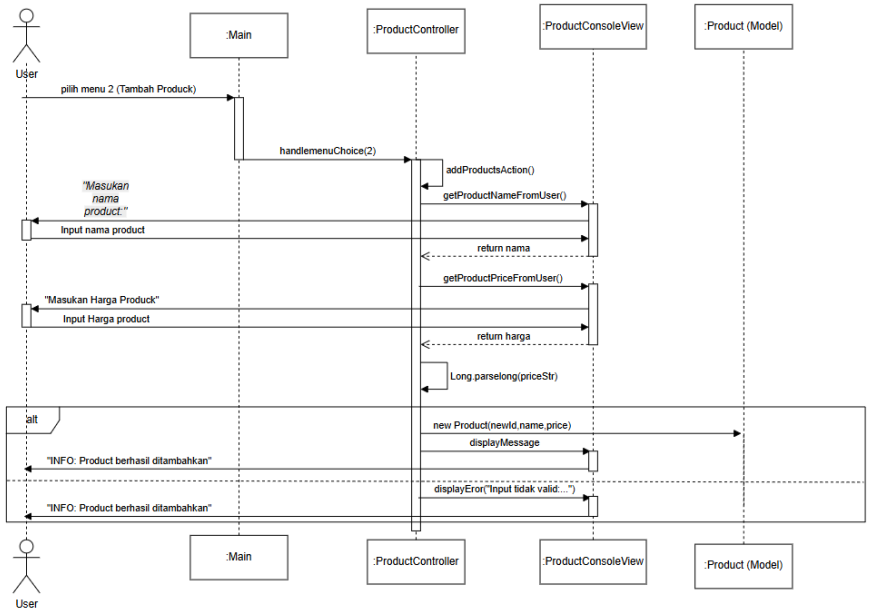
c. ProductController.java

```
controller > ProductController.java > ProductController > ProductController(ProductConsoleView)
1   package controller;
2
3
4   import view.ProductConsoleView;
5   import java.util.ArrayList;
6   import model.Product;
7
8   public class ProductController{
9       private final ProductConsoleView view;
10      private final ArrayList<Product> model = new ArrayList<>();
11
12      public ProductController(ProductConsoleView view){
13          this.view = view;
14          model.add(new Product(id: 1, name: "Laptop ASUS", price: 950));
15      }
16
17      public void handleMenuChoice(int choice){
18          switch (choice) {
19              case 1:
20              displayProductsAction();
21              break;
22              case 2:
23              addProduct();
24              break;
25              case 3:
26              view.displayMessage(message: "keluar dari aplikasi");
27              break;
28              default:
29              view.displayError(message: "pilihan tidak valid");
30          }
31      }
32
33      private void displayProductsAction(){
34          view.displayAllProducts(model);
35      }
36
37      private void addProduct(){
38          String name = view.getProductNameFromUser();
39          String priceStr = view.getProductPriceFromUser();
40          try{
41              double price = Double.parseDouble(priceStr);
42              if (price <= 0 ) {
43                  throw new IllegalArgumentException(s: "harga hatus positif lebih dari 0 ");
44              }
45              int newId = model.size()+1;
46              model.add(new Product(newId, name, (long) price));
47              view.displayMessage(message: "Produk ditambahkan");
48          } catch(IllegalArgumentException e){
49              view.displayError("gagal menambah produk: "+ e.getMessage());
50          }
51      }
52  }
```

3. Sequence Diagram
   a. Menampilkan semua peroduct



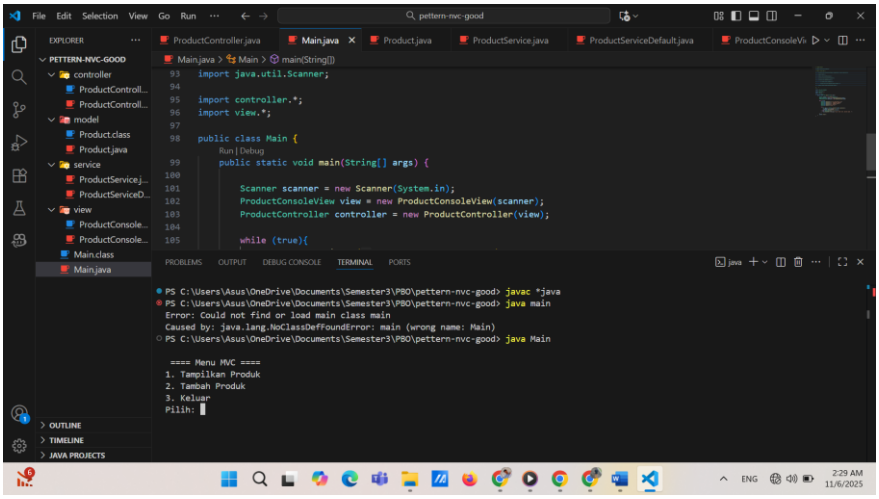

b. Menambahkan Product




d. Main.java

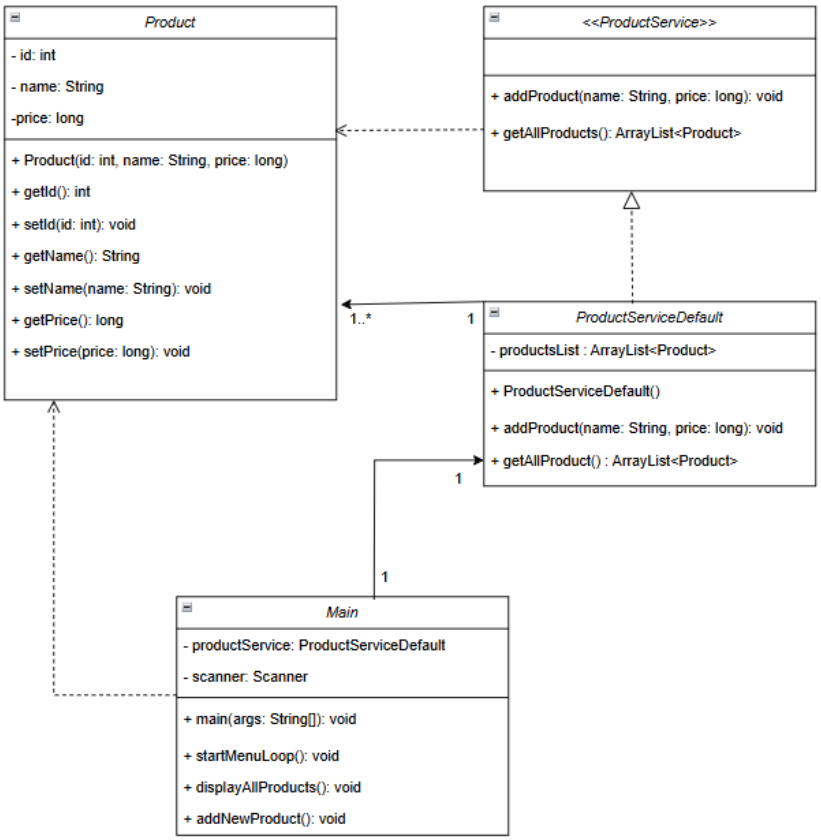
```java
import java.util.Scanner;

import controller.*;
import view.*;

public class Main {
    Run | Debug
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        ProductConsoleView view = new ProductConsoleView(scanner);
        ProductController controller = new ProductController(view);

        while (true){
            System.out.println(x: "\n ==== Menu MVC ====");
            System.out.println(x: "1. Tampilkan Produk");
            System.out.println(x: "2. Tambah Produk");
            System.out.println(x: "3. Keluar");
            System.out.print(s: "Pilih: ");

            try {
                int choice = Integer.parseInt(scanner.nextLine());
                if (choice == 3 ) break;
                controller.handleMenuChoice(choice);
            } catch (Exception e) {
                view.displayError(message: "input tidak valid. masukan angka. ");
            }
        }
        scanner.close();
    }
}
```

4. Compile & Run



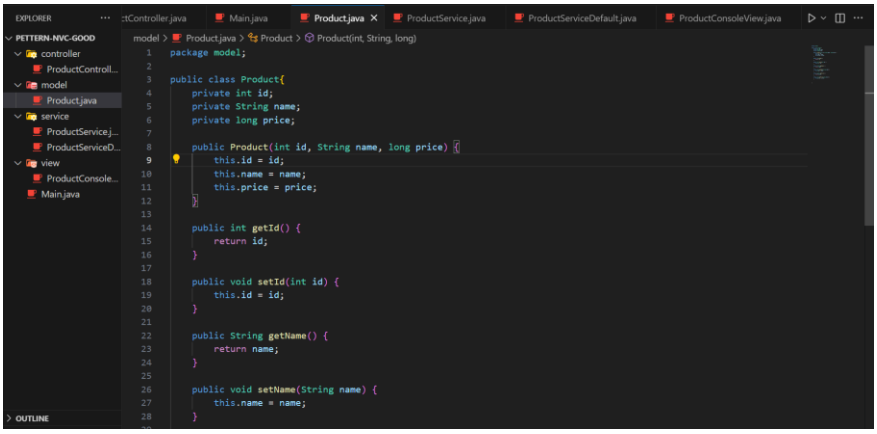

D. kode program Service Layer Pattern

1. Class Diagram




2. Kode program

   a. Product.java



```
29
30      public double getPrice() {
31          return price;
32      }
33
34      public void setPrice(long price) {
35          this.price = price;
36      }
37
38
39
40  }
```

b. ProductService.java

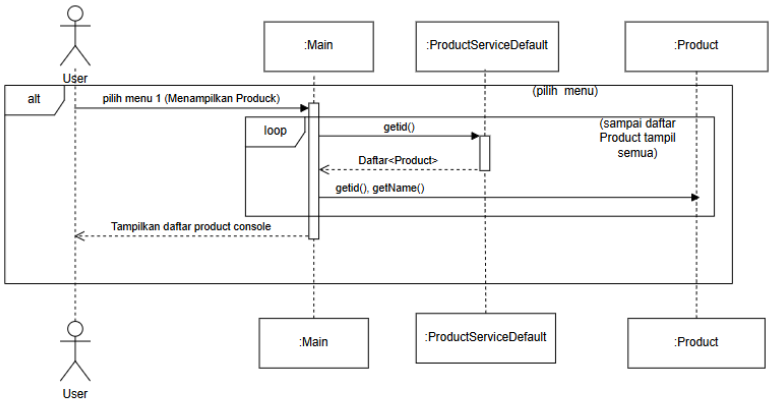
```
service > ProductService.java > ...
1   package service;
2
3   import java.util.ArrayList;
4   import model.Product;
5
6   public interface ProductService {
7
8       void addProduct(String name, long price);
9
10      ArrayList<Product> getAllProducts();
11
12  }
13
```

c. ProductServiceDefault.java

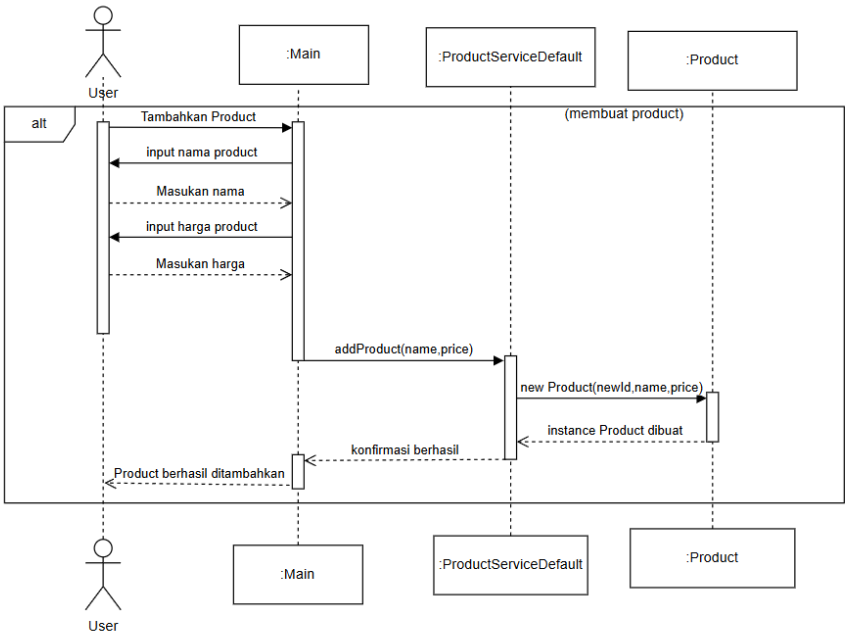
service > ProductServiceDefault.java > ...

```java
package service;

import java.util.ArrayList;
import model.Product;

public class ProductServiceDefault implements ProductService {

    private final ArrayList<Product> productList = new ArrayList<>();

    public ProductServiceDefault() {
        productList.add(new Product(id: 1, name: "Laptop ASUS", price: 9500000));
        productList.add(new Product(id: 2, name: "Monitor Dell", price: 2500000));
    }

    @Override
    public void addProduct(String name, long price) {
        if (price <= 0) {
            throw new IllegalArgumentException(
                s: "Harga harus angka positif lebih dari 0");
        }
        int newId = productList.size() + 1;
        productList.add(new Product(newId, name, price));
    }

    @Override
    public ArrayList<Product> getAllProducts() {
        return productList;
    }
}
```

3. Sequence Diagram
   a. Lihat Semua Product




   b. Tambahkan Product

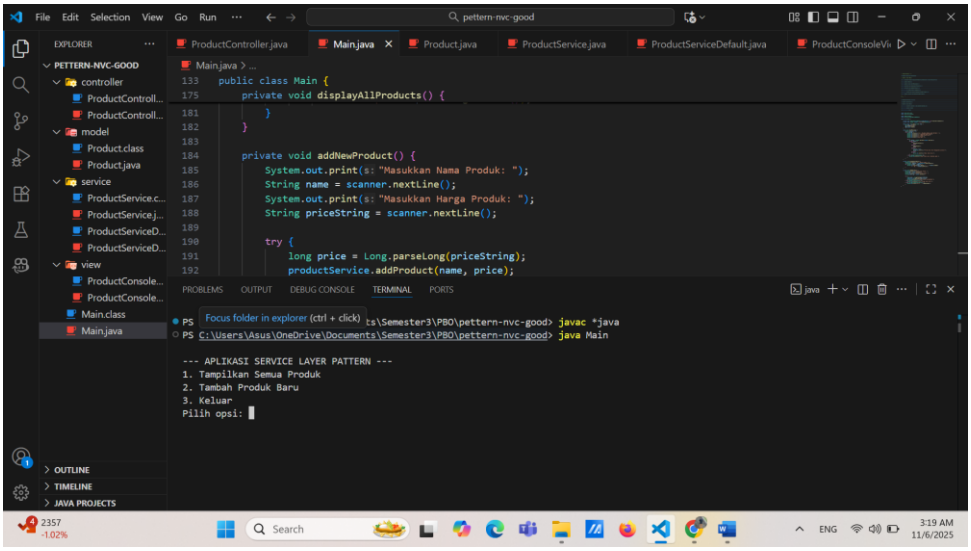



d. Main.java

```java
import java.util.List;
import java.util.Scanner;

import model.Product;
import service.ProductServiceDefault;

public class Main {

    private final ProductServiceDefault productService = new ProductServiceDefault();
    private final Scanner scanner = new Scanner(System.in);

    Run | Debug
    public static void main(String[] args) {
        Main app = new Main();
        app.startMenuLoop();
        app.scanner.close();
    }

    public void startMenuLoop() {
        boolean running = true;
        while (running) {
            System.out.println(x: "\n--- APLIKASI SERVICE LAYER PATTERN ---");
            System.out.println(x: "1. Tampilkan Semua Produk");
            System.out.println(x: "2. Tambah Produk Baru");
            System.out.println(x: "3. Keluar");
            System.out.print(s: "Pilih opsi: ");

            try {
                int choice = Integer.parseInt(scanner.nextLine());
                switch (choice) {
```

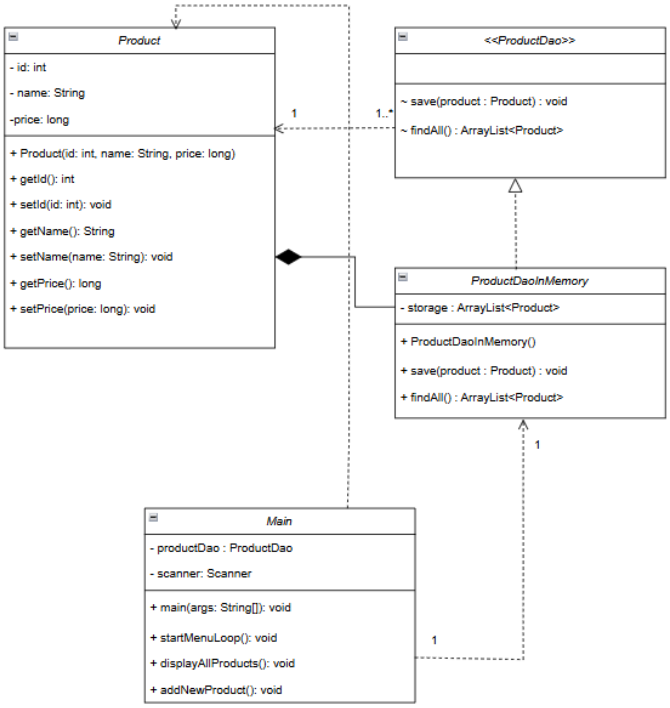
```
                case 1:
                    displayAllProducts();
                    break;
                case 2:
                    addNewProduct();
                    break;
                case 3:
                    running = false;
                    System.out.println(x: "Terima kasih telah menggunakan aplikasi!");
                    break;
                default:
                    System.out.println(x: "Opsi tidak valid.");
            }
        } catch (NumberFormatException e) {
            System.out.println(x: "Input tidak valid. Masukkan angka.");
        }
    }
}

private void displayAllProducts() {
    System.out.println(x: "\n--- Daftar Produk ---");
    List<Product> products = productService.getAllProducts();
    for (Product product : products) {
        System.out.println(product.getId() + " - " + product.getName()
                + " - Rp. " + product.getPrice());
    }
}
```

e. Compile & Run



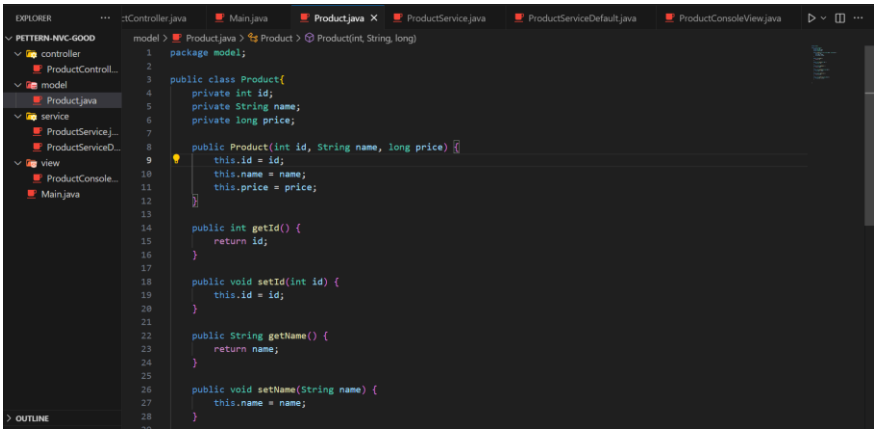

## E. kode program DAO Pattern

### 1. Class Diagram




2. Kode program

   a. Product.java



```java
package model;

public class Product{
    private int id;
    private String name;
    private long price;

    public Product(int id, String name, long price) {
        this.id = id;
        this.name = name;
        this.price = price;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
```

```
29
30      public double getPrice() {
31          return price;
32      }
33
34      public void setPrice(long price) {
35          this.price = price;
36      }
37
38
39
40  }
```

b. ProductDao.java

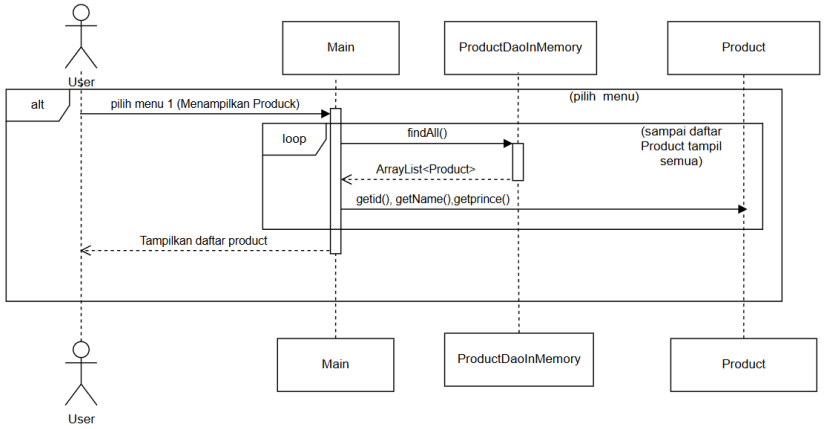
```
dao > ProductDao.java > ...
1   package dao;
2
3   import java.util.ArrayList;
4   import model.Product;
5
6   public interface ProductDao {
7       void save(Product product);
8       ArrayList<Product> findAll();
9   }
10
```

c. ProductDaoMemoy.java

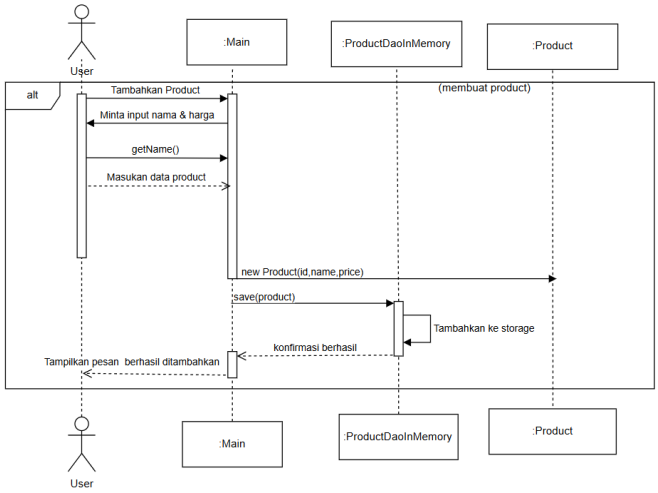
dao > memory > ProductDaoInMemory.java > ...

```java

package dao.memory;

import java.util.ArrayList;

import dao.ProductDao;
import model.Product;

public class ProductDaoInMemory implements ProductDao {

    private final ArrayList<Product> storage = new ArrayList<>();

    public ProductDaoInMemory() {
        storage.add(new Product(id: 1, name: "Laptop ASUS", price: 9500000));
        storage.add(new Product(id: 2, name: "Monitor Dell", price: 2500000));
    }

    @Override
    public void save(Product product) {
        storage.add(product);
    }

    @Override
    public ArrayList<Product> findAll() {
        return storage;
    }
}
```

3. Sequence Diagram
   a. Melihat semua product




   b. Menambahkan Product

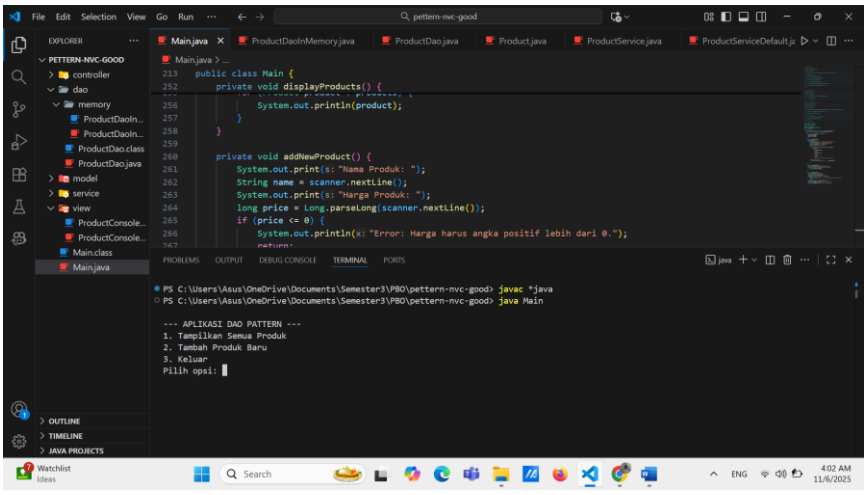



d. Main.java

```
206  import java.util.ArrayList;
207  import java.util.Scanner;
208
209  import dao.ProductDao;
210  import dao.memory.ProductDaoInMemory;
211  import model.Product;
212
213  public class Main {
214      private final ProductDao productDao = new ProductDaoInMemory();
215      private final Scanner scanner = new Scanner(System.in);
216
         Run | Debug
217      public static void main(String[] args) {
218          Main app = new Main();
219          app.startMenuLoop();
220          app.scanner.close();
221      }
222
223      public void startMenuLoop() {
224          boolean running = true;
225          while (running) {
226              System.out.println(x: "\n--- APLIKASI DAO PATTERN ---");
227              System.out.println(x: "1. Tampilkan Semua Produk");
228              System.out.println(x: "2. Tambah Produk Baru");
229              System.out.println(x: "3. Keluar");
230              System.out.print(s: "Pilih opsi: ");
231              try {
232                  int choice = Integer.parseInt(scanner.nextLine());
233                  switch (choice) {
```

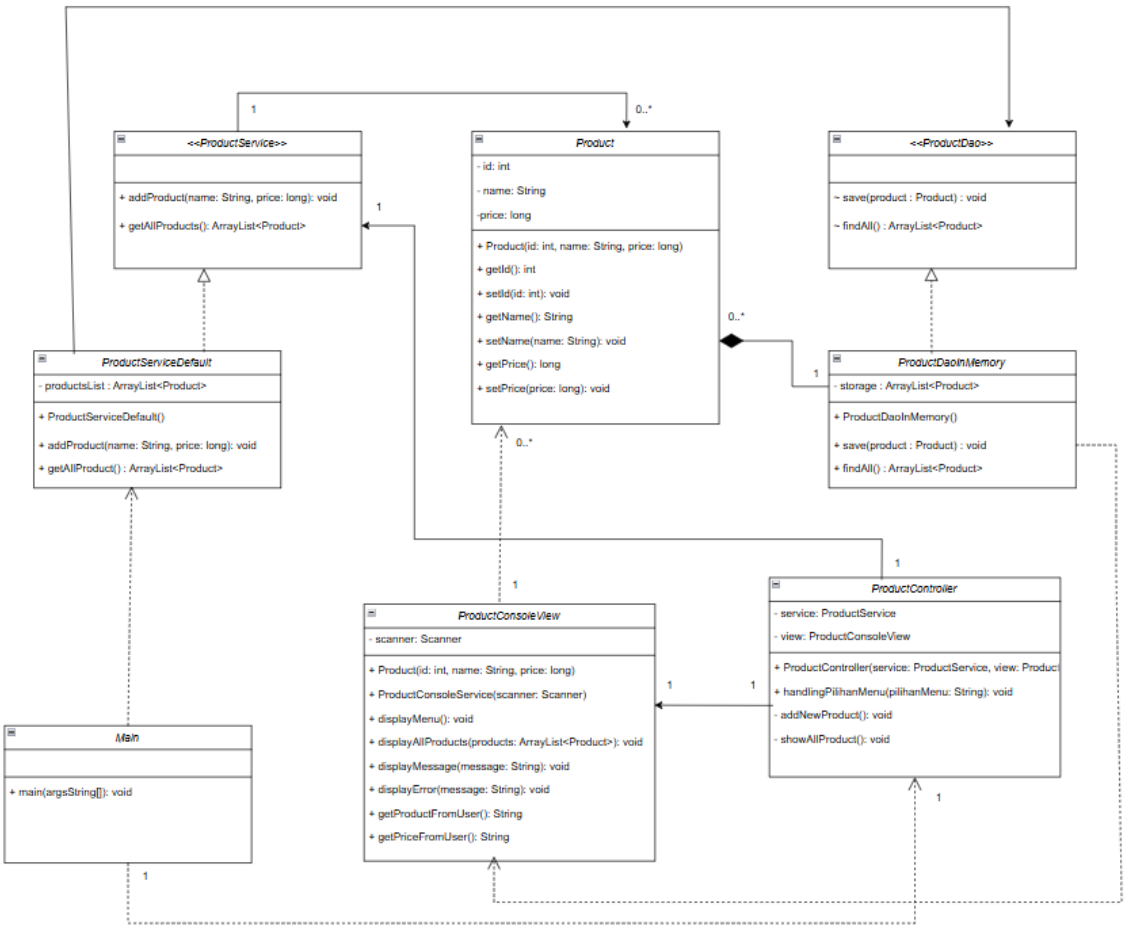
```
                    case 1:
                        displayProducts();
                        break;
                    case 2:
                        addNewProduct();
                        break;
                    case 3:
                        running = false;
                        break;
                    default:
                        System.out.println(x: "Opsi tidak valid.");
                }
            } catch (Exception e) {
                System.out.println("Input error: " + e.getMessage());
            }
        }
    }

    private void displayProducts() {
        System.out.println(x: "\n--- Daftar Produk ---");
        ArrayList<Product> products = productDao.findAll();
        for (Product product : products) {
            System.out.println(product);
        }
    }

    private void addNewProduct() {
        System.out.print(s: "Nama Produk: ");
        String name = scanner.nextLine();
        System.out.print(s: "Harga Produk: ");
        long price = Long.parseLong(scanner.nextLine());
        if (price <= 0) {
            System.out.println(x: "Error: Harga harus angka positif lebih dari 0.");
            return;
        }
        int newId = productDao.findAll().size() + 1;
        Product newProduct = new Product(newId, name, price);
        productDao.save(newProduct);
        System.out.println(x: "Produk berhasil ditambahkan!");
    }
}

```

e. Compile & Run



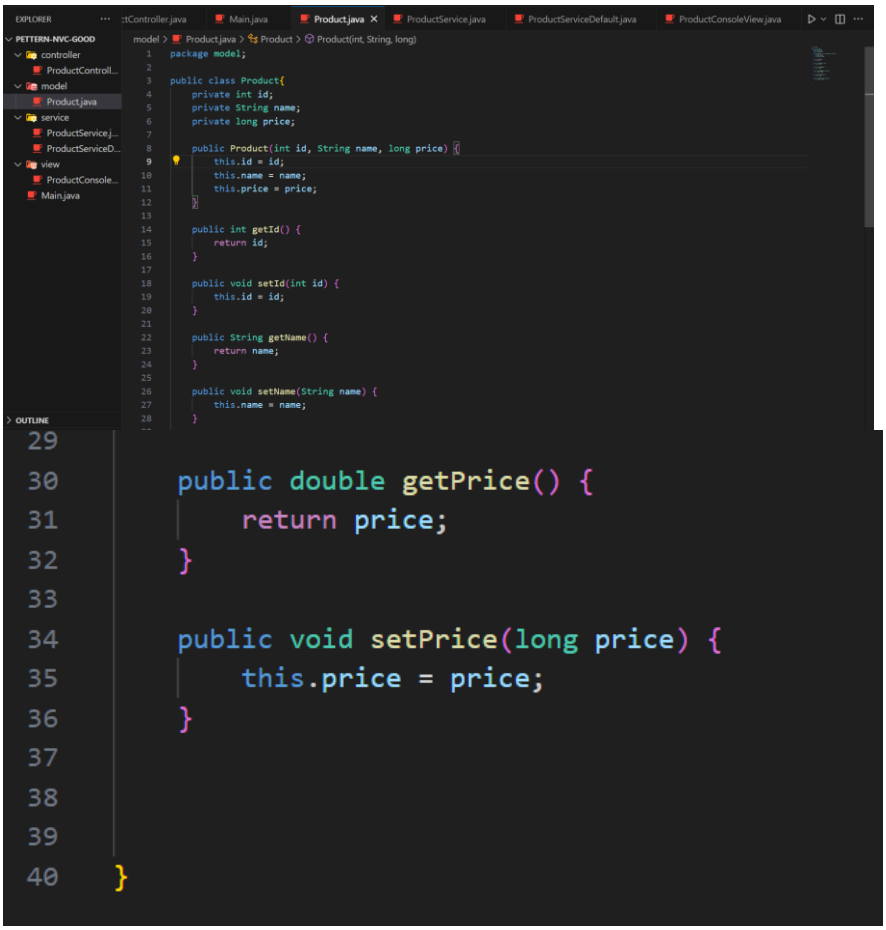
F. kode program MVC + Service Layer + DAO Pattern

1. Class Diagram



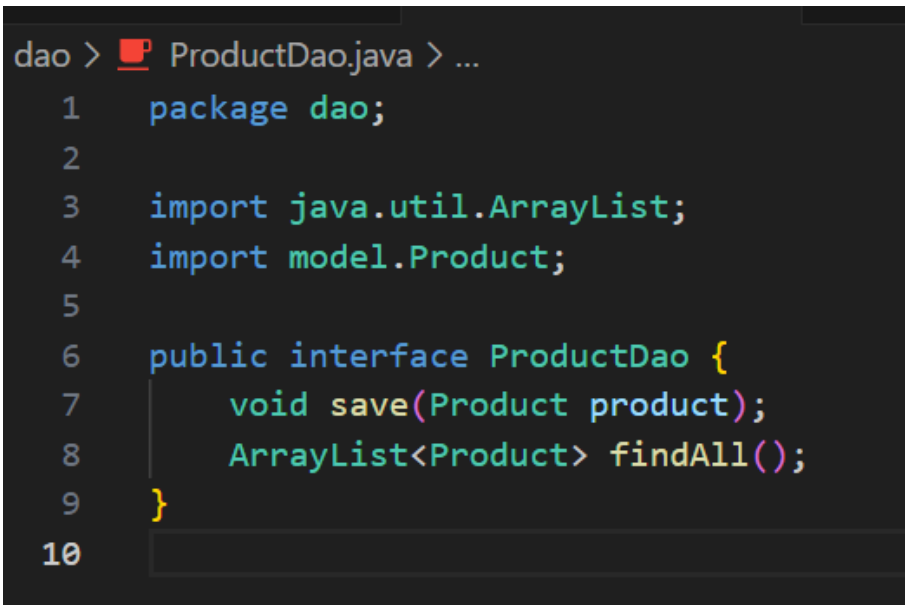

2. Kode program

a. Product.java



```
29
30      public double getPrice() {
31          return price;
32      }
33
34      public void setPrice(long price) {
35          this.price = price;
36      }
37
38
39
40  }
```

b. ProductDao.java



```
dao > ProductDao.java > ...
1   package dao;
2
3   import java.util.ArrayList;
4   import model.Product;
5
6   public interface ProductDao {
7       void save(Product product);
8       ArrayList<Product> findAll();
9   }
10
```

c. ProductDaoMemoy.java

dao > memory > ProductDaoInMemory.java > ...

```java

package dao.memory;

import java.util.ArrayList;

import dao.ProductDao;
import model.Product;

public class ProductDaoInMemory implements ProductDao {

    private final ArrayList<Product> storage = new ArrayList<>();

    public ProductDaoInMemory() {
        storage.add(new Product(id: 1, name: "Laptop ASUS", price: 9500000));
        storage.add(new Product(id: 2, name: "Monitor Dell", price: 2500000));
    }

    @Override
    public void save(Product product) {
        storage.add(product);
    }

    @Override
    public ArrayList<Product> findAll() {
        return storage;
    }
}
```

d. ProductService.java

service > ProductService.java > ...

```java
package service;

import java.util.ArrayList;
import model.Product;

public interface ProductService {

    void addProduct(String name, long price);

    ArrayList<Product> getAllProducts();

}

```

e. ProductServicedefault.java

service > ProductServiceDefault.java > ...

```java
package service;

import java.util.ArrayList;
import model.Product;

public class ProductServiceDefault implements ProductService {

    private final ArrayList<Product> productList = new ArrayList<>();

    public ProductServiceDefault() {
        productList.add(new Product(id: 1, name: "Laptop ASUS", price: 9500000));
        productList.add(new Product(id: 2, name: "Monitor Dell", price: 2500000));
    }

    @Override
    public void addProduct(String name, long price) {
        if (price <= 0) {
            throw new IllegalArgumentException(
                s: "Harga harus angka positif lebih dari 0");
        }
        int newId = productList.size() + 1;
        productList.add(new Product(newId, name, price));
    }

    @Override
    public ArrayList<Product> getAllProducts() {
        return productList;
    }
}
```

f. ProductServiceDefault.java

service > ProductServiceDefault.java > ...

```java
package service;

import java.util.ArrayList;
import model.Product;

public class ProductServiceDefault implements ProductService {

    private final ArrayList<Product> productList = new ArrayList<>();

    public ProductServiceDefault() {
        productList.add(new Product(id: 1, name: "Laptop ASUS", price: 9500000));
        productList.add(new Product(id: 2, name: "Monitor Dell", price: 2500000));
    }

    @Override
    public void addProduct(String name, long price) {
        if (price <= 0) {
            throw new IllegalArgumentException(
                s: "Harga harus angka positif lebih dari 0");
        }
        int newId = productList.size() + 1;
        productList.add(new Product(newId, name, price));
    }

    @Override
    public ArrayList<Product> getAllProducts() {
        return productList;
    }
}
```

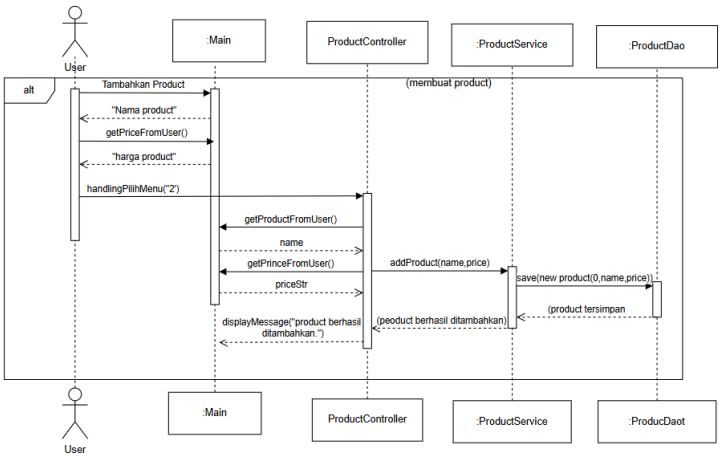
g. ProductConsoleView.java

```java
package view;

import java.util.ArrayList;
import java.util.Scanner;
import model.Product;

public class ProductConsoleView {
    private final Scanner scanner;

    public ProductConsoleView(Scanner scanner) {
        this.scanner = scanner;
    }

    public void displayMenu() {
        System.out.println("--- APLIKASI + SERVICE LAYER + DAO PATTERN ---");
        System.out.println("1. Tampilkan semua produk");
        System.out.println("2. Tambah produk baru");
        System.out.println("3. Keluar");
        System.out.print("Pilih opsi: ");
    }

    public void displayAllProducts(ArrayList<Product> products) {
        System.out.println("\n--- Daftar Produk ---");
        if (products.isEmpty()) {
            System.out.println("Tidak ada produk tersedia.");
        } else {
            for (Product product : products) {
                System.out.println(product.getId() + " - " + product.getName()
                    + " - Rp " + product.getPrice());
            }
        }
    }

    public void displayMessage(String message) {
        System.out.println("INFO: " + message);
    }

    public void displayError(String message) {
        System.out.println("ERROR: " + message);
    }

    public String getProductNameFromUser() {
        System.out.print("Masukkan Nama Produk: ");
        return scanner.nextLine();
    }

    public long getProductPriceFromUser() {
        System.out.print("Masukkan Harga Produk: ");
        return Long.parseLong(scanner.nextLine());
    }

    public int getMenuChoiceFromUser() {
        return Integer.parseInt(scanner.nextLine());
    }
}
```

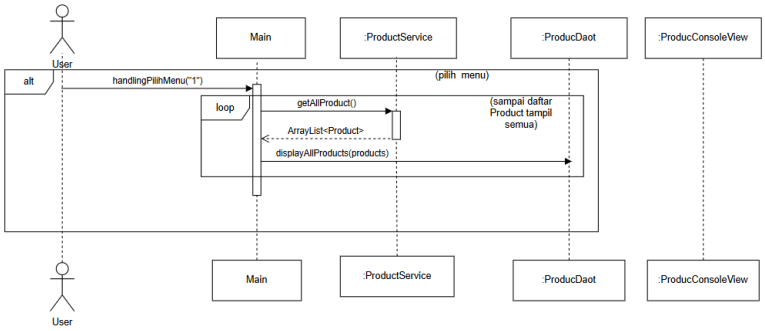
h. ProductController.java

```java
package controller;

import service.ProductService;
import view.ProductConsoleView;

public class ProductController {
    private final ProductService service;
    private final ProductConsoleView view;

    public ProductController(ProductService service, ProductConsoleView view) {
        this.service = service;
        this.view = view;
    }

    public void handleMenuChoice(String menuChoice) {
        try {
            int menu = Integer.parseInt(menuChoice);
            switch (menu) {
                case 1:
                    showAllProducts();
                    break;
                case 2:
                    addNewProduct();
                    break;
                case 3:
                    view.displayMessage(message: "Aplikasi ditutup.");
                    System.exit(status: 0);
                    break;
                default:
                    view.displayError(message: "Opsi tidak valid.");
                    break;
            }
        } catch (NumberFormatException e) {
            view.displayError(message: "Input tidak valid. Masukkan angka.");
        }
    }

    private void addNewProduct() {
        String name = view.getProductNameFromUser();
        long price = view.getProductPriceFromUser();
        try {
            service.addProduct(name, price);
            view.displayMessage(message: "Produk berhasil ditambahkan!");
        } catch (Exception e) {
            view.displayError("Gagal menambah produk: " + e.getMessage());
        }
    }

    private void showAllProducts() {
        view.displayAllProducts(service.getAllProducts());
    }
}
```

3. Sequence Diagram
   a. Tambahkan Product



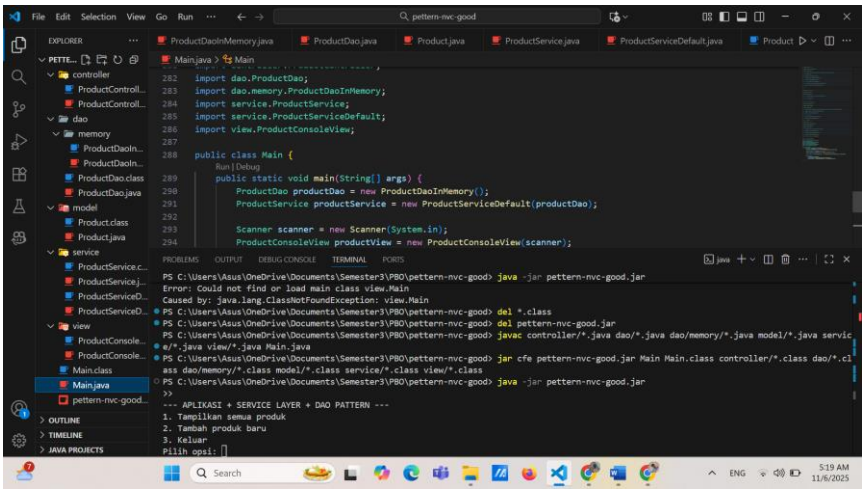

b. Lihat semua Product




i. Main.java

```java
import java.util.ArrayList;
import java.util.Scanner;

import dao.ProductDao;
import dao.memory.ProductDaoInMemory;
import model.Product;

public class Main {
    private final ProductDao productDao = new ProductDaoInMemory();
    private final Scanner scanner = new Scanner(System.in);

    Run | Debug
    public static void main(String[] args) {
        Main app = new Main();
        app.startMenuLoop();
        app.scanner.close();
    }

    public void startMenuLoop() {
        boolean running = true;
        while (running) {
            System.out.println(x: "\n--- APLIKASI DAO PATTERN ---");
            System.out.println(x: "1. Tampilkan Semua Produk");
            System.out.println(x: "2. Tambah Produk Baru");
            System.out.println(x: "3. Keluar");
            System.out.print(s: "Pilih opsi: ");
            try {
                int choice = Integer.parseInt(scanner.nextLine());
                switch (choice) {
                    case 1:
                        displayProducts();
                        break;
                    case 2:
                        addNewProduct();
                        break;
                    case 3:
                        running = false;
                        break;
                    default:
                        System.out.println(x: "Opsi tidak valid.");
                }
            } catch (Exception e) {
                System.out.println("Input error: " + e.getMessage());
            }
        }
    }

    private void displayProducts() {
        System.out.println(x: "\n--- Daftar Produk ---");
        ArrayList<Product> products = productDao.findAll();
        for (Product product : products) {
            System.out.println(product);
        }
    }

    private void addNewProduct() {
        System.out.print(s: "Nama Produk: ");
        String name = scanner.nextLine();
        System.out.print(s: "Harga Produk: ");
        long price = Long.parseLong(scanner.nextLine());
        if (price <= 0) {
            System.out.println(x: "Error: Harga harus angka positif lebih dari 0.");
            return;
        }
        int newId = productDao.findAll().size() + 1;
        Product newProduct = new Product(newId, name, price);
        productDao.save(newProduct);
        System.out.println(x: "Produk berhasil ditambahkan!");
    }
}

```

j. Compile & Run




Repository
https://github.com/OhLanns/patternTugas2403099.git