

Nama : Marlina Yunus
Nim : 2109106143
Kelas : Infomatika C2 2021

Screenshot Tugas Posttest 6 Praktikum PBO

MainMenu.java

```
//package : mengelompokkan berbagai hal (kelas, interface, enum, dsb) yang memiliki fungsi berkaitan
package MainMenu;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;

//public : pengaksesan suatu variabel instan diluar class
public class MainMenu {
    static String fileName;
    static ArrayList<String> todoLists;
    static boolean isEditing = false;
    static Scanner input;

    //static : sebuah property atau method diakses langsung tanpa object
    Run | Debug
    public static void main(String[] args) {
        // Abstract & Keyword Final
        Nama nama = new Nama();
        nama.Nama();
        nama.Boneka();
        nama.JenisBoneka();
    }
}
```

```
System.out.println("FILE: " + fileName);

// run the program (main loop)
while (true) {
    showMenu();
}

static void clearScreen(){
    try {
        final String os = System.getProperty(key:"os.name");
        if (os.contains(s:"Windows")) {
            // clear screen untuk windows
            new ProcessBuilder(..command:"cmd", "/c", "cls")
                .inheritIO()
                .start()
                .waitFor();
        } else {
            // clear screen untuk Linux, Unix, Mac
            Runtime.getRuntime().exec(command:"clear");
            System.out.print(s:"\033[H\033[2J");
            System.out.flush();
        }
    } catch (final IOException | InterruptedException e) {
        System.out.println("Error karena: " + e.getMessage());
    }
}
```



```

static void showTodoList() {
    clearScreen();
    readTodoList();
    if (todoLists.size() > 0) {
        System.out.println(x: "");
        System.out.println(x: "Boneka yang Ditambahkan : ");
        int index = 0;
        for (String data : todoLists) {
            System.out.println(String.format(format: "%d) %s", index, data));
            index++;
        }
    } else {
        System.out.println(x: "Sorry! No Data Here");
    }

    if (!isEditing) {
        backToMenu();
    }
}

```

```

static void addTodoList() {
    clearScreen();
    System.out.println(x: " ");
    System.out.println(x: "Nama Boneka Impian Apa?");
    System.out.print(s: "Jawab : ");
    String newTodoList = input.nextLine();

    try {
        try ( // tulis file
            FileWriter fileWriter = new FileWriter(fileName, append: true)) {
            fileWriter.append(String.format(format: "%s\n", newTodoList));
        }
        System.out.println(x: " ");
        System.out.println(x: "----- ");
        System.out.println(x: "        Berhasil Ditambahkan!        ");
        System.out.println(x: "----- ");
    } catch (IOException e) {
        System.out.println("Oops! Kenali Error : " + e.getMessage());
    }
    backToMenu();
}

```

```

static void editTodoList() {
    isEditing = true;
    showTodoList();

    try {
        System.out.println(x: " ");
        System.out.print(s: "Pilih Indeks : ");
        int index = Integer.parseInt(input.nextLine());

        if (index > todoLists.size()) {
            throw new IndexOutOfBoundsException(s: "Data Kamu Salah!");
        } else {
            System.out.println(x: "Nama Boneka Baru : ");
            String newData = input.nextLine();

            // update data
            String set = todoLists.set(index, newData);
            System.out.println(todoLists.toString());
        }
    }
}

```

```

    try {
        // write new data
        try (FileWriter fileWriter = new FileWriter(fileName, append:false)) {
            // write new data
            for (String data : todolists) {
                fileWriter.append(String.format(format:"%s\n", data));
            }
        }

        System.out.println(x:"");
        System.out.println(x:"-----");
        System.out.println(x:"          Berhasil Diubahkan!          ");
        System.out.println(x:"-----");
    } catch (IOException e) {
        System.out.println("Oops! Know the Error : " + e.getMessage());
    }
}
} catch (IndexOutOfBoundsException | NumberFormatException e) {
    System.out.println(e.getMessage());
}
}

isEditing = false;
backToMenu();
}

```

```

static void deleteTodoList() {
    isEditing = true;
    showTodoList();

    System.out.println(x:" ");
    System.out.println(x:"Pilih Indeks : ");
    int index = Integer.parseInt(input.nextLine());

    try {
        if (index > todolists.size()) {
            throw new IndexOutOfBoundsException(s:"Kamu memasukan data yang salah!");
        } else {
            System.out.println(x:"");
            System.out.println(x:"Indeks yang Kamu Hapus Adalah Indeks : ");
            System.out.println(String.format(format:"(%d) %s", index, todolists.get(index)));
            System.out.println(x:"");
            System.out.println(x:"Adakah Kamu Yakin Hapus?");
            System.out.print(s:"Jawab (Y/T) : ");
            String jawab = input.nextLine();

```

```

        if (jawab.equalsIgnoreCase(anotherString:"Y")) {
            // hapus data
            String remove = todolists.remove(index);

            // tulis ulang file
            try {
                // write new data
                try (FileWriter fileWriter = new FileWriter(fileName, append:false)) {
                    // write new data
                    for (String data : todolists) {
                        fileWriter.append(String.format(format:"%s\n", data));
                    }
                }

                System.out.println(x:"");
                System.out.println(x:"-----");
                System.out.println(x:"          Berhasil Dihapus!          ");
                System.out.println(x:"-----");
            } catch (IOException e) {
                System.out.println("Oops! Know the Error : " + e.getMessage());
            }
        }
    } catch (IndexOutOfBoundsException e) {
        System.out.println(e.getMessage());
    }
}

//back
isEditing = false;
backToMenu();
}
}

```

Nama.java

```
package MainMenu;

public class Nama extends Boneka {
    public void Boneka() {
        // Penamaan Boneka ()
        System.out.println(x:"Barbie");
        System.out.println(x:"Teddy Bear");
        System.out.println(x:" ");
    }

    // Override Boneka ()
    @Override
    public void JenisBoneka() {
        System.out.println(x:"Jenis Boneka itu adalah Barbie Australian");
        System.out.println(x:"Jenis Boneka itu adalah Scrianize");
    }
}
```

Boneka.java

```
package MainMenu;

abstract class Boneka {
    public abstract void JenisBoneka();
    public void Nama() {
        System.out.println(x:"Barbie from Australian Island");
    }
}
```