



JOBSHEET 11

LOOPING 2

DAVA RIZKY H

254107020246

1. Objective

- Students are able to understand the concept of nested loop
- Students are able to explain nested loop writing format
- Students are able to implement nested loop flowchart using Java programming language

2. Laboratory

2.1 Experiment 1: Loop Review

1. This experiment is aimed at reviewing the loop that had been studied in the previous week. In experiment 1, a program will be made to make a view * N times sideways.
2. Create a new class with the name **Star** and save it in the **Star.java** file
3. Write the basic structure of the Java programming language which contains the **main()** function
4. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
5. In the **main()** function that has been created, declare a **Scanner** object with the name **sc**
6. On the next line, write the instructions for entering the value to be stored in variable **N**

```
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
```

7. On the next line, create a loop syntax using **for**



```
for (int i = 1; i <= N; i++) {
    System.out.print("*");
}
```

Note: please note that the `print` command is used, not `println` because we want to display without any new lines

8. Compile and run the program.
9. Observe the results, match the results of the running programs that you have created according to the following display

Enter the value of N: 5

```
File Edit Selection View Go Run ... ⏪ ⏩
PRAKTIKUMDASPRO-1
star.java U
star.java jobsheet8
EXERCISE 1 DASAR PEMEROGAMAN
jobsheet3
BiayaBensin
CicilanMotor
CicilanMotor.java
ContohOperasi5
GajiKaryawan05
MenghitungLuasPersegiPanjang05
MenghitungTotalBayar05
jobsheet4
jobsheet5
jobsheet6
jobsheet7
jobsheet8
star.java
README.md

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Run star + ⏪ ⏩ ...
PS C:\Users\ASUS R00\OneDrive\TaI_LiE\U\PRAKTIKUMDASPRO\PRAKTIKUMDASPRO-1> & 'C:\Program Files\Java\jdk-24\bin\java.exe' -XX:+ShowCodeDetailsInExceptionMessages' -cp 'C:\Users\ASUS R00\AppData\Roaming\Code\User\workspaceStorage\1e99f81629e843abd52a7ea\ab9e1ff7\nedhat.java\jdt_ws\PRAKTIKUMDASPRO-1_4da0ffe2\bin' 'jobsheet8.star'
Masukkan nilai N: 5
*****
PS C:\Users\ASUS R00\OneDrive\TaI_LiE\U\PRAKTIKUMDASPRO\PRAKTIKUMDASPRO-1>
```

Questions!

1. If in **for** loop, the initialization **i = 1** is changed to **i = 0**, what is the result? How can it be like that?
2. If in **for** loop, condition **i <= N** is changed to **i > N**, what is the result? How can it be like that?
3. If in **for** loop, the condition for step **i++** is changed to **i--** what is the result? How can it be like that?

ANSWER



1. When `i` starts from **0** instead of **1**, the loop will run from **0 to N**, not from **1 to N**. That means the loop executes **N + 1 times**, because **0** is included as a valid iteration.
2. The condition `i > N` means the loop will only execute if **i is greater than N**. However, at the start, `i` is usually **less than or equal to N**, so the condition is **false** from the beginning, and the loop is **skipped**.
3. `i++` means the variable increases, while `i--` decreases it. If the condition is `i <= N`, then `i` will keep getting smaller and **never reach a value greater than N**, so the condition will **always be true** — the loop never stops.

2.2 Experiment 2: Square Star

1. In Experiment 2, an experiment was carried out on nested loops. The case that will be solved is to create a square display `*`, with side lengths of `N`. Suppose `N` is entered as 5, then the results are as follows:

```
*****  
*****  
*****  
*****  
*****
```

2. If you look closely, it's actually like the case of Experiment 1, isn't it? In Experiment 1, for example, the input `N` has a value of 5, then what will be produced is `*****` (we can think of this as an **inner loop** that displays 5 stars `*****`), then for the case of Experiment 2, this is not the result of experiment 1 does it just need to be repeated `N` times? (by adding an **outer loop** to repeat the **inner loop** process `N` times)
3. Create a new class with the name **Square** and save it in the **Square.java** file
4. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
5. Create a **main()** function, and add the same program code as the contents of the **main()** function in Experiment 1



```

Scanner sc = new Scanner(System.in);
System.out.print("Enter the value of N: ");
int N = sc.nextInt();
for (int i = 1; i <= N; i++) {
    System.out.print("*");
}

```

6. Compile and run the program. Make sure the results given are the same as in Experiment 1

7. Pay attention to the iterative syntax used to print * N times sideways.

In step 5, we make **for** loop structure (red box) as an **inner loop**

8. Furthermore, the inner loop needs to be repeated N times to display the * symbol to generate output like step 1. Thus, it is necessary to add an **outer loop**

```

for (int iOuter = 1; iOuter <= N; iOuter++) {
    for (int i = 1; i <= N; i++) {
        System.out.print("*");
    }
    System.out.println("");
}

```

9. Compile and run the program.

10. Observe the results, match the results of the running programs that you have created according to the following display

```

Enter the value of N: 5
*****
*****
*****
*****
*****
```



```

public class square {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Masukkan nilai N: ");
        int N = sc.nextInt();

        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                System.out.print("*");
            }
            System.out.println();
        }
        sc.close();
    }
}

```

The terminal output shows:

```

Masukkan nilai N: 5
*****
*****
*****
*****
*****
```

Questions!

- Pay attention to outer loop. If in **for** syntax, the initialization **iOuter = 1** is changed to **iOuter = 0**, what is the result? How can it be like that?
- Return the program to normal with initialization **iOuter = 1**. Then pay attention to the inner loop. If in **for** syntax, the initialization **i = 1** is changed to **i = 0**, what is the result?
How can it be like that?
- What is the difference between outer loop and inner loop?
- Why is it necessary to add the syntax **System.out.println();** under inner loop? What will happen if the syntax is omitted?
- Commit and push the changes to GitHub

ANSWER

- iOuter** controls **how many rows** will be printed (the outer loop). If it starts from **0**, the loop runs from **0 to N**, not from **1 to N**. That means the outer loop executes **N + 1 times**, so the output will have **one additional line**.
- The **inner loop** controls **how many stars (*)** are printed per row. If it starts from **0**, it runs from **0 to N**, which means **N + 1 iterations**. As a result, each row prints **one extra star**.



3. • **Outer loop:**
Controls **how many rows** or how many times the inner loop will be repeated.
 - **Inner loop:**
Controls **what is printed inside each row**, such as how many stars or numbers appear in one line.
4. **System.out.println();** is used to move to a new line after the inner loop finishes printing one row.
If it is removed, then all the stars will appear on the same line without line breaks.
5. <https://github.com/rizkydava1233-ops/PRAKTIKUMDASPRO/blob/main/jobsheet8/square.java>

2.3 Experiment 3: Triangle Star

1. In Experiment 3, a right-angled triangle with a height of N was carried out. Suppose N is entered as 5, then the results are as follows:

```
*  
**  
***  
****  
*****
```

2. Create a new class with the name **Triangle** and save it in the **Triangle.java** file
3. Because the program requires input from the keyboard, it is necessary to import the **Scanner** class, so add the import syntax in the top line of the program.
4. Create a **main()** function, and fill in the following program code into the **main()** function



```

System.out.print("Enter the value of N: ");
int N = sc.nextInt();
int i = 0;
while (i <= N) {
    int j = 0;
    while (j < i) {
        System.out.print("*");
        j++;
    }
    i++;
}
}

```

5. Compile and run the program. Observe the results!

The screenshot shows a Java development environment with the following details:

- File Structure:** The project is named "PRAKTIKUMDASPRO-1". It contains several packages like "EXERCISE 1 DASAR PEMROGRAMAN", "jobsheet3", "jobsheet4", "jobsheet5", "jobsheet6", "jobsheet7", and "jobsheet8". Each package contains Java files such as "star.java", "triangle.java", and "square.java".
- Code Editor:** The "triangle.java" file is open. It contains the following code:


```

public class triangle {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the value of N: ");
        int N = sc.nextInt();

        // Outer while loop untuk baris
        int i = 0;
        while (i < N) {
            // Inner while loop untuk kolom
            int j = 0;
            while (j < i) {
                System.out.print("*");
                j++;
            }
            // Pindah ke baris baru
            System.out.println();
            i++;
        }
    }
}

```
- Terminal:** The terminal window shows the command "java triangle" being run. The output is "Enter the value of N: 18", followed by a series of asterisks representing the triangle pattern.
- Messages:** A message box indicates that the last command attempted to run the Java class "triangle.java" from the directory "C:\Program FJ". An error message states "Error: Could not find or load main class triangle".

Questions!

1. Look at the results, does the output produced with a value of N = 5 match the following display?

```

*
**
***
****
*****

```

IT IS NOT IN ACCORDANCE WITH! There is a blank line at the beginning that shouldn't be there.

2. If not, which parts should be improved or added? Describe any parts that need to be improved or added!

Perbaikan yang dilakukan: Mengubah inisialisasi $i = 0$ menjadi $i = 1$

The screenshot shows a Java code editor in VS Code. The code prints a right-angled triangle pattern based on user input. The code uses nested loops: an outer loop for rows and an inner loop for columns. It includes imports for Scanner and System.out, and handles closing the scanner.

```
public class triangle {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the value of N: ");
        int N = sc.nextInt();

        // Outer while loop untuk baris
        int i = 1;
        while (i <= N) {
            // Inner while loop untuk kolom
            int j = 0;
            while (j < i) {
                System.out.print("*");
                j++;
            }
            // Pindah ke baris baru
            System.out.println();
            i++;
        }
        sc.close();
    }
}
```

The code editor interface includes:

- File, Edit, Selection, View, Run, etc. menu at the top.
- Explorer sidebar showing project structure and files like star.java, triangle.java, square.java, and various jobsheet files.
- Terminal tab showing command-line output: "Enter the value of N: 5".
- Output tab showing the resulting triangle pattern: *
**

- Search, Go Live, and other status indicators at the bottom.

2.4 Experiment 4: Guess the Number Quiz

1. Create a new class with the name **Quiz** and save it in the **Quiz.java** file 2. Add Scanner and Random libraries in the top line of the program

```
import java.util.Scanner;  
import java.util.Random;
```

- ### 3. Create a `main()` function

4. In the **main()** function, make a **Scanner** declaration with the name **input** and **Random** declaration with the name **rand**. In this case, Random is used to randomize the numbers

```
Scanner input = new Scanner(System.in);  
Random rand = new Random();
```

5. Then on the next line, add the following syntax

```
char menu = 'y';
do {
    int number = rand.nextInt(10) + 1;
    boolean success = false;
    do {
        System.out.print("Guess the number (1-10): ");
        int answer = input.nextInt();
        input.nextLine();
        success = (answer == number);
    } while (!success);
    System.out.print("Do you want to repeat the game (Y/N)");
    menu = input.next().charAt(0);
    input.nextLine();
} while (menu == 'Y' || menu == 'y');
```

Note: the `input.nextLine()` syntax in that snippet is used to ignore the new line character

6. Compile and run the program. Observe the results!

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "PRAKTIKUMDASPRO_1". The "jobsheet8" folder contains four files: "quiz.java", "square.java", "star.java", and "triangle.java".
- Code Editor:** Displays the content of the "quiz.java" file. The code is a Java program that generates a random number between 1 and 10 and asks the user to guess it.
- Terminal:** Shows the command "dk-24/bin/java.exe -XX:+ShowCodeDetailsInExceptionMessages -cp 'C:/Users/VISUS/ROG/VippData/Roamin..." followed by several "Guess the number (1-10):" prompts and the user's responses "8", "2", "3", and "5".
- Output:** Shows an error message: "Error: Could not find or load main class quiz Caused by: java.lang.ClassNotFoundException: quiz".
- Problems:** Shows a single error icon next to "quiz.java".
- Status Bar:** Shows "ln 16 Col 1 Spaces 4 UTF-8 CR/LF () Java".

Questions!

- ### 1. Explain the program flow in Experiment 4!

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

```
import java.util.Random;
```



- **Scanner** is used to read input from the keyboard.
- **Random** is used to generate random numbers.

```
public class Quiz {  
    public static void main(String[] args) {
```

- **input** is used to receive user input.
- **rand** is used to generate random numbers between 1 and 10.

```
    char menu = 'y';
```

The variable **menu** is used to store the user's choice — whether they want to play again or not.

```
    do {  
        ...  
    } while (menu == 'Y' || menu == 'y');
```

- This loop makes the game **repeat** as long as the user enters **Y or y**.
- If the user enters **N or n**, the game stops.

```
    int number = rand.nextInt(10) + 1;
```

This line generates a random number between **1 and 10**.
(`nextInt(10)` generates numbers 0–9, so adding 1 shifts the range to 1–10.)

```
    boolean success = false;
```

This variable is used to mark whether the user's guess is correct or not.

```
    do {  
        System.out.print("Guess the number (1-10): ");  
        int answer = input.nextInt();  
        input.nextLine(); // ignore newline character  
  
        success = (answer == number);  
    } while (!success);
```

- The program asks the user to enter a guess (`answer`).



- It checks if the guess matches the random number (number).
- If it's **incorrect**, the loop repeats.
- If it's **correct**, the inner loop ends.

```
System.out.print("Do you want to repeat the game (Y/N)? ");
menu = input.next().charAt(0);
```

- After guessing correctly, the program asks the user if they want to play again.
- If the user enters **Y/y**, the game restarts.
- If the user enters **N/n**, the game ends.

```
input.close();
```

This closes the `Scanner` object to free up system resources.

2. What must be done to discontinue (not repeat) the game?

Type "N" or "n" when asked "Do you want to repeat the game (Y/N)?", or

Remove the outer loop so the program runs only once.

3. Modify the program above, so that it can display information about: input the guess value entered by the user, whether it is smaller or greater than the answer (number) randomly determined by the computer!

```
public class quiz {
    public static void main(String[] args) {
        int success = false;
        System.out.print("Guess the number (1-10): ");
        while (!success) {
            int userGuess = Integer.parseInt(input.nextLine());
            if (userGuess < 1 || userGuess > 10) {
                System.out.println("Too Large! Try a lower number.");
            } else if (userGuess < 5) {
                System.out.println("Too Small! Try a higher number.");
            } else {
                System.out.println("Correct!");
                success = true;
            }
        }
        // Ask if user wants to repeat the game
        System.out.print("Do you want to repeat the game (Y/N)? ");
        menu = input.next().charAt(0);
        input.nextLine(); // Used to ignore the newline character
        while (menu == 'Y' || menu == 'y') {
            // Close Scanner
            input.close();
        }
    }
}
```



4. Commit and push the changes to GitHub

<https://github.com/rizkydava1233-ops/PRAKTIKUMDASPRO/blob/main/jobsheet8/square.java>

2.5 Experiment 5: Filling and Displaying Arrays

1. Create a new class with the name **NestedLoopStudentID**
2. Create a **main()** function
3. In the **main()** function, add a declaration for Scanner named **scanner** and a 2dimensional array declaration named **temps** with 5 rows and 7 columns of type **double**
4. Add the following program code

```
for (int i = 0; i < temps.length; i++) {
    System.out.println("City: " + i);
    for (int j = 0; j < temps[0].length; j++) {
        System.out.print("Day " + (j + 1) + ": ");
        temps[i][j] = scanner.nextDouble();
    }
    System.out.println();
}
```

5. Then, also add the following several lines of program code

```
for (int i = 0; i < temps.length; i++) {
    System.out.println("City: " + i);
    for (int j = 0; j < temps[0].length; j++) {
        System.out.print(temps[i][j] + " ");
    }
    System.out.println();
}
```

6. Compile and run the program. Observe the results!

Note: This program example is a case example for storing temperature data in a city.

Check back to last week's material slides.

The screenshot shows a Java code editor with the following code:

```
public class NestedLoopStudentID {
    public static void main(String[] args) {
        // Declaration for Scanner named scanner and 2D array named temps
        Scanner scanner = new Scanner(System.in);
        double[][] temps = new double[5][7];

        // Input temperature data for each city and day
        for (int i = 0; i < temps.length; i++) {
            System.out.println("City: " + i);
            for (int j = 0; j < temps[0].length; j++) {
                System.out.print("Day " + (j + 1) + ": ");
                temps[i][j] = scanner.nextDouble();
            }
            System.out.println();
        }

        // Display temperature data
        System.out.println("----- Temperature Data Display ---\n");
        for (int i = 0; i < temps.length; i++) {
            System.out.println("City: " + i);
        }
    }
}
```

The code implements a nested loop to input temperature data for 5 cities over 7 days. It then displays the data. The code editor interface includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The terminal tab shows the command run: NestedLoopStudentID.

Questions!

- ### 1. Explain the program flow in Experiment 5!

```
double[][] temps = new double[5][7];
```

- The program creates a 2D array named `temps` with **5 rows (cities)** and **7 columns (days)**.
 - Each element stores temperature data in **double** format.

```
for (int i = 0; i < temps.length; i++) { ... }
```

- The **outer loop (i)** represents each **city**.
 - The **inner loop (j)** represents each **day** for that city.
 - The user is asked to enter temperature values one by one.

```
for (int i = 0; i < temps.length; i++) { ... }
```

- The outer loop displays the **city name/number**.
 - The inner loop prints all **temperature values for each day** in that city.
 - After finishing one city, it moves to a new line.

2. Modify the program to display an array using foreach!



```

public class NestedLoopStudentID {
    public static void main(String[] args) {
        System.out.println("City: " + i);
        for (int j = 0; j < temps[i].length; j++) {
            System.out.print("Day " + (j + 1) + ": ");
            temps[i][j] = scanner.nextDouble();
        }
        System.out.println();
    }

    // Display temperature data using foreach
    System.out.println("==== Temperature Data Display (Using Foreach) ===");
    int cityIndex = 0;
    for (double[] city : temps) {
        System.out.println("City: " + cityIndex);
        for (double temp : city) {
            System.out.print(temp + " ");
        }
        System.out.println();
        cityIndex++;
    }
}

PS C:\Users\ASUS ROG\OneDrive\Tai liệu\PRAKTIKUMDASPRO\PRAKTIKUMDASPRO-1> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ASUS ROG\AppData\Roaming\Code\User\workspaceStorage\1e99f81829e843aabd52a/e7a7a91ff7\nediat.java;jdt_us\PRAKTIKUMDASPRO-1_4d8fffe21b1n' 'jobsheet8.NestedLoopStudentID'
City: 0
Day 1: 35
Day 2: 35

```

3. Modify the program so that it can display the average value for each city!

```

public static void main(String[] args) {
    System.out.println();
    cityIndex++;
}

// Display average temperature for each city
System.out.println("==== Average Temperature Per City ===");
for (int i = 0; i < temps.length; i++) {
    double sum = 0;
    for (int j = 0; j < temps[i].length; j++) {
        sum += temps[i][j];
    }
    double average = sum / temps[i].length;
    System.out.println("City " + i + " average: " + average + "°C");
}

// Close Scanner
scanner.close();
}

PS C:\Users\ASUS ROG\OneDrive\Tai liệu\PRAKTIKUMDASPRO\PRAKTIKUMDASPRO-1> & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ASUS ROG\AppData\Roaming\Code\User\workspaceStorage\1e99f81829e843aabd52a/e7a7a91ff7\nediat.java;jdt_us\PRAKTIKUMDASPRO-1_4d8fffe21b1n' 'NestedLoopStudentID'
City 0 average: 31.57142857142857°C
City 1 average: 30.714285714285715°C
City 2 average: 77.0°C
City 3 average: 76.14285714285714°C
City 4 average: 25.857142857142858°C

```

4. Commit and push the changes to GitHub

<https://github.com/rizkydava1233-ops/PRAKTIKUMDASPRO/blob/main/jobsheet8/NestedLoopStudentID.java>



3. Assignment

- Create a program to print a numeric triangle display as below based on the N input (minimum N value is 3). Example N = 5

```

1
12
123
1234
12345

```

The screenshot shows a Java IDE interface with the project name "PRAKTIKUMDASPRO-1". The left sidebar displays a file tree with several Java files under "PRAKTIKUMDASPRO-1" and "jobsheet8". The main editor window contains the following Java code:

```

package jobsheets.assignment;
import java.util.Scanner;
public class NumericTriangle {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Input
        System.out.print("Enter N (minimum 3): ");
        int N = sc.nextInt();

        // Validate input
        if (N < 3) {
            System.out.println("Error: N must be at least 3");
        } else {
            // Print the triangle
            for (int i = 1; i <= N; i++) {
                // Print numbers from 1 to i
                for (int j = 1; j <= i; j++) {
                    System.out.print(j);
                }
            }
        }
    }
}

```

The terminal tab shows the output of running the program with input "7":

```

Enter N (minimum 3): 7
1
12
123
1234
12345
123456
1234567

```

- Create a program to print the star triangle view shown below based on the N input (minimum N value is 5). Example N = 7

```

*****
 *****
 ****
 ***
 **
 *

```



The screenshot shows a Java IDE interface with the following details:

- Project Structure:** The project is named "PRAKTIKUMDASPRO-1". It contains several packages: "jobsheet3", "assignment", and "PRAKTIKUMDASPRO-1". Under "PRAKTIKUMDASPRO-1", there are sub-folders like "jobsheet3", "MenghitungLuasPersegiPanjang05", "jobsheet4", "jobsheet5", "jobsheet6", "jobsheet7", "jobsheet8", and "assignment". The "assignment" folder contains three files: "NumencTriangle.java", "StarTriangle.java", and "Quiz.java".
- Code Editor:** The code editor shows the "StarTriangle.java" file. The code prints a star triangle pattern based on user input. It includes imports for `java.util.Scanner` and `java.util.*`. The main method reads an integer `n` from the user and prints a pattern of `n` rows of stars.
- Terminal:** The terminal window shows the command `java StarTriangle` being run. It prompts the user for input: "Enter the value of N (minimum 5):".
- Output:** The output window shows the resulting star triangle pattern for N=5:

```

      *
     * *
    * * *
   * * * *
  * * * * *
 * * * * * *

```

3. Create a program to print a square numeric display like the one below based on N input (minimum N value is 3). Example N = 3 and N = 5

```

      5 5 5 5 5
      5       5
  3 3 3       5       5
  3   3       5       5
  3 3 3       5 5 5 5 5

```

The screenshot shows a Java IDE interface with the following details:

- Project Structure:** The project is named "PRAKTIKUMDASPRO-1". It contains several packages: "jobsheet3", "assignment", and "PRAKTIKUMDASPRO-1". Under "PRAKTIKUMDASPRO-1", there are sub-folders like "EXERCISE 1 DASAR PEMROGRAMAN", "jobsheet3", "jobsheet4", "jobsheet5", "jobsheet6", "jobsheet7", "jobsheet8", and "assignment". The "assignment" folder contains three files: "NumencTriangle.java", "SquareNumericDisplay.java", and "StarTriangle.java".
- Code Editor:** The code editor shows the "SquareNumericDisplay.java" file. The code prints a square numeric display pattern based on user input. It includes imports for `java.util.Scanner` and `java.util.*`. The main method reads an integer `n` from the user and prints a pattern of `n` rows of numbers.
- Terminal:** The terminal window shows the command `java SquareNumericDisplay` being run. It prompts the user for input: "Enter the value of N (minimum 3):".
- Output:** The output window shows the resulting square numeric display pattern for N=5:

```

      5 5 5 5 5
      5   5
  5   5   5
  5   5   5
  5 5 5 5 5

```

4. In 2024, Malang State Polytechnic will host the Porseni national event. There are several sports that are competed in, such as **badminton**, **table tennis**, **basketball**, and **volleyball**. Each sport sends its **5 best athletes** from all polytechnics throughout Indonesia to take part in this biannual event. Create a data storage to display information on the **names of athletes** from the various branches mentioned in **ascending order**.

```

public class PorseniAthleteSystem {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        PorseniAthleteSystem system = new PorseniAthleteSystem();

        // Sample data
        system.addAthlete("Ahmad Rizki", "sport:Badminton", "polytechnic:Malang");
        system.addAthlete("Siti Nurhaliza", "sport:Badminton", "polytechnic:Malang");
        system.addAthlete("Budi Santoso", "sport:Table Tennis", "polytechnic:Malang");
        system.addAthlete("Dewi Lester", "sport:Basketball", "polytechnic:ITS");
        system.addAthlete("Eko Prasetyo", "sport:Volleyball", "polytechnic:ITS");

        while (true) {
            System.out.println("1. Add Athlete");
            System.out.println("2. Display All Athletes (Sorted)");
            System.out.println("3. Display Athletes by Sport");
            System.out.println("4. Display All Sports");
            System.out.println("5. Exit");
        }
    }
}

```

5. Implement the flowchart of the features you created in the previous theory assignment about nested loops!

```

public static void fitur2_laporanBooking() {
    System.out.println("1. LAPORAN BOOKING HOTEL");

    // REATED LOOP: loop luar untuk bulan, loop dalam untuk tipe kamar
    for (int bulan = 1; bulan <= jumlahBulan; bulan++) {
        System.out.println("Bulan: " + namabulan(bulan));
        int totalPerBulan = 0;

        for (int tipe = 1; tipe <= jumlahTipeKamar; tipe++) {
            System.out.print(" " + namatipe(tipe) + " : ");
            jumlahBooking[bulan][tipe];
            totalPerBulan += jumlahBooking[bulan][tipe];
        }

        System.out.println(" Total bulan ini: " + totalPerBulan);
        System.out.println();
    }
}

```

6. Don't forget, all program code must be pushed to your repository.

<https://github.com/rizkydava1233-ops/PRAKTIKUMDASPRO/blob/main/jobsheet8/assignment/SistemHotelLengkap.java>

