



## JOBSHEET 5

### SELECTION

[github.com/tezabae](https://github.com/tezabae)

### **1. Objective**

1. Students are able to solve problems/case studies using simple selection syntax and nested selection.
2. Students are able to apply simple selection syntax and nested selection into Java programs.

### **2. Practicum**

#### **2.1 Experiment 1 : Implementing IF and IF-ELSE to Print KRS**

**Trial Time: 40 minutes**

At the beginning of each semester, students are required to print their KRS (Kartu Rencana Semester or Student Study Plan) and signed by their Academic Advisor. The SIAKAD system will check the student's UKT (or tuition fee) payment status. If the student has paid the UKT, the system displays the KRS for printing. Based on this scenario, a Java program will be created with the following steps.

1. Open your local repository **PraktikumDaspro** cloned previously via Visual Studio Code, then create a **jobsheet5** folder by typing **mkdir jobsheet5** in the terminal .

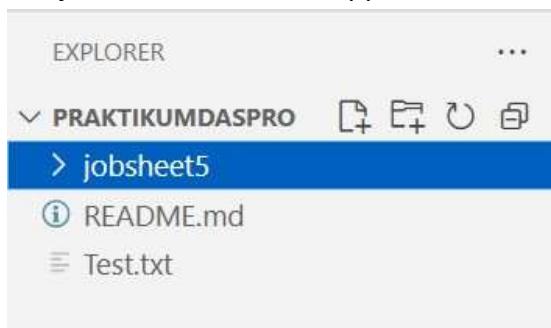
The screenshot shows the Visual Studio Code interface with the terminal tab active. The terminal window displays the command `mkdir jobsheet5` being run in a Microsoft Windows environment. The output shows the creation of the folder:

```
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\POLINEMA\OneDrive\Document\PraktikumDaspro>mkdir jobsheet5

C:\Users\POLINEMA\OneDrive\Document\PraktikumDaspro>
```

The **jobsheet5** folder will appear in the Visual Studio Code Explorer.



2. Create a new file with the name **ifCetakKRSNoPresensi.java** in the **jobsheet5** folder that has been created



3. Create a basic Java program structure with the **main()** function. 4. Import library Scanner and declare scanner with name **sc**

```
Scanner sc = new Scanner(System.in);
```

5. Add the following code to receive input from the keyboard which is then stored in the boolean variable **uktLunas**

```
public static void main (String[] args) {
    System.out.println("----Print KRS SIAKAD----");
    System.out.println("Have the tuition fees been paid in full? (true/false): ");
    boolean uktLunas = sc.nextBoolean();
```

6. Create an IF selection structure to check whether UKT has been paid in full.

```
if (uktLunas) {
    System.out.println("UKT payment has been verified");
    System.out.println("KRS now can be printed and you can ask the academic advisor to sign it.");
}
```

7. Compile and run the program to produce the following display.

```
----Print KRS SIAKAD---
Have the tuition fees been paid in full? (true/false):
true
UKT payment has been verified
KRS now can be printed and you can ask the academic advisor to sign it.
```

8. Commit your program to Github with the message "Attempt 1"

### Question

- Why doesn't the check in the IF structure involve conditions with relational operators?  
Because the variable already boolean
- When the program is run, then you enter the value **false** , what is the result?  
There will be no result on the terminal

```
Print KRS SIAKAD
=====
Have the tuition fee been paid? (true/false)
false
PS C:\code\jobsheet\PraktikumDaspro\PraktikumDaspro>
```

- The system needs to provide information that if the user enters a false value , the output will be "Registration rejected. Please pay UKT first." Modify the program by adding an ELSE!

```
PS C:\code\jobsheet\PraktikumDaspro\PraktikumDaspro> cd .. & cd 'C:\code\jobsheet\PraktikumDaspro\PraktikumDaspro' & 'C:\Program Files\Java\jdk-24\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\hellirook\AppData\Roaming\Code\User\workspaceStorage\8ce317c9f4aa82e5d51a7f63c23da5db\redhat.java\jdt_ws\PraktikumDaspro_5bc6c480\bin' 'ifCetakKRSNoPresensi'
Print KRS SIAKAD
=====
Have the tuition fee been paid? (true/false)
false
Registration rejected. Please pay UKT first.
```

- Commit and push your modifications to Github with the message "Test Modification 1"**



## 2.2. Experiment 2: SWITCH-CASE for Printing KRS

Trial Time: 60 minutes

At the beginning of each semester, students are required to print their KRS (Student Study Plan) for signature by their Academic Advisor. The SIAKAD system will check the student's current semester and then display that semester's KRS for printing. Based on this case, the Java program is created with the following steps.

1. Create a new file named **switchCetakKRSNoPresensi.java** in the **jobsheet5** folder
2. Create a basic Java program structure with the **main()** function.
3. Import library Scanner and declare scanner with name **sc**
4. Add the following code to receive input from the keyboard which is then stored in the **semester** variable of type int.

```
System.out.println(x:"--- Print KRS SIAKAD ---");
System.out.print(s:"Enter current semester: ");
int semester = sc.nextInt();
```

5. Create a SWITCH-CASE selection structure to check the current semester.



```
switch (semester) {  
    case 1:  
        System.out.println(x:"KRS Semester 1 is displayed");  
        break;  
    case 2:  
        System.out.println(x:"KRS Semester 2 is displayed");  
        break;  
    case 3:  
        System.out.println(x:"KRS Semester 3 is displayed");  
        break;  
    case 4:  
        System.out.println(x:"KRS Semester 4 is displayed");  
        break;  
    case 5:  
        System.out.println(x:"KRS Semester 5 is displayed");  
        break;  
    case 6:  
        System.out.println(x:"KRS Semester 6 is displayed");  
        break;  
    case 7:  
        System.out.println(x:"KRS Semester 7 is displayed");  
        break;  
    case 8:  
        System.out.println(x:"KRS Semester 8 is displayed");  
        break;  
    default:  
        System.out.println(x:"Invalid Semester");  
}
```

6. Compile and run the program to produce the following display.

```
--- Print KRS SIAKAD ---  
Enter current semester: 3  
KRS Semester 3 is displayed
```

7. Commit your program to Github with the message “Attempt 2”

### Question

1. What is the function of syntax **break** ?

the break statement is used to exit the switch block once a matching case is found and its code is executed

2. What is the role of syntax **default** on SWITCH-CASE selection structure?

The default case acts as a catch-all. If the value of the variable in the switch statement does not match any of the specified case values, the code inside the default block will be executed



The default case acts as a catch-all. If the value of the variable in the switch statement does not match any of the specified case values, the code inside the default block will be executed

3. Create a new file named **ifElseCetakKRSNoPresensi.java**. This file contains the program resulting from the transformation of the KRS printing program using the SWITCH-CASE structure that has been created into the IF-ELSE IF-ELSE form .

```

File Edit Selection View Go Run Terminal Help
ifElseCetakKRSNoPresensi.java
switchCetakKRSNoPresensi 3 switchCetakKRSNoPresensi 3 main(jar)
1 package assignments;
2 import java.util.Scanner;
3 public class switchCetakKRSNoPresensi {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         System.out.println("Pilih semester (1-8):");
7         System.out.println("1. Semester 1 is displayed");
8         System.out.println("2. Semester 2 is displayed");
9         System.out.println("3. Semester 3 is displayed");
10        System.out.println("4. Semester 4 is displayed");
11        System.out.println("5. Semester 5 is displayed");
12        System.out.println("6. Semester 6 is displayed");
13        System.out.println("7. Semester 7 is displayed");
14        System.out.println("8. Semester 8 is displayed");
15        System.out.println("9. Invalid semester input");
16        int semester = scanner.nextInt();
17        switch (semester) {
18            case 1:
19                System.out.println("Semester 1 is displayed");
20                break;
21            case 2:
22                System.out.println("Semester 2 is displayed");
23                break;
24            case 3:
25                System.out.println("Semester 3 is displayed");
26                break;
27            case 4:
28                System.out.println("Semester 4 is displayed");
29                break;
30            case 5:
31                System.out.println("Semester 5 is displayed");
32                break;
33            case 6:
34                System.out.println("Semester 6 is displayed");
35                break;
36            case 7:
37                System.out.println("Semester 7 is displayed");
38                break;
39            case 8:
40                System.out.println("Semester 8 is displayed");
41                break;
42            default:
43                System.out.println("Invalid semester input");
44                break;
45        }
46    }
47 }

```

4. Commit and push your modifications to Github with the message “Trial Modification 2”

### 2.3. Experiment 3: Nested IF to Check Thesis Examination Requirements

Trial Time: 90 minutes

A student will register for a thesis exam. The SIMTA system will first check the administrative requirements, that the student must be free of compensation. If this requirement is met, the system then checks the guidance log records. To be able to register for the exam, students must have at least 8 guidance sessions with supervisor 1 and at least 4 guidance sessions with supervisor 2. If all requirements are met, the student can proceed to the thesis exam registration process. If not, the system will display the reason for failure.

Based on this case, a Java program is created with the following steps.

1. Create a new file named **nestedUjianSkripsiNoPresensi.java** in the **jobsheet5** folder
2. Create a basic Java program structure with the **main()** function.
3. Import library Scanner and declare scanner with name **sc**
4. Declare a message variable of type String to store the output result of the condition.

**String message;**



5. Add the following code to receive input from the keyboard regarding component information, which is then stored in the **compensationStatus** variable with data type String.

```
System.out.print(s:"Has the student been cleared of compensation? (Y/T or Yes/No): ");
String compensationStatus = sc.nextLine().trim();
```

*Note: trim() function is used to remove spaces before and after a String.*

6. Add the following code to receive input from the keyboard regarding guidance log information which is then stored in the variables **supervisor1** and **supervisor2** of type int

```
System.out.print(s:"Enter the number of guidance logs with Supervisor 1: ");
int supervisor1 = sc.nextInt();
System.out.print(s:"Enter the number of guidance logs with Supervisor 2: ");
int supervisor2 = sc.nextInt();
```

7. Nested -IF selection structure to check the condition of the component at the first level and the number of guidance logs at the second level.

```
if (free) {
    if (supervisor1 >= 8 && supervisor2 >= 4) {
        message = "All requirements are met. The student may register for the thesis exam.";
    } else if (supervisor1<8 && supervisor2<4) {
        message = "Failed: Guidance logs with Supervisor 1 are fewer than 8 AND Supervisor 2 fewer than 4.";
    }
    else if (supervisor1 < 8 && supervisor2 >= 4) {
        message = "Failed: Guidance logs with Supervisor 1 are fewer than 8.";
    } else if (supervisor1 >= 8 && supervisor2 < 4) {
        message = "Failed: Guidance logs with Supervisor 2 are fewer than 4.";
    }
} else {
    message = "Failed: The student has not been cleared of compensation.";
}

System.out.println(message);
```

8. Compile and run the program to produce the following display.

```
Has the student been cleared of compensation? (Y/T or Yes/No): yes
Enter the number of guidance logs with Supervisor 1: 9
Enter the number of guidance logs with Supervisor 2: 9
All requirements are met. The student may register for the thesis exam.
```

9. Commit your program to Github with the message “Attempt 3”

## Question

1. What happens if a student answers " No " to **compensationStatus** question? Why is that?

There will be display the message “Failed: The student has not been cleared of compensation.”

2. Explain the meaning of the following code snippet!

```
if (supervisor1 >= 8 && supervisor2 >= 4) {
    if (supervisor1 >= 8 && supervisor2 >= 4) {
```



`supervisor1 >= 8`: Checks if the value of the supervisor1 variable is greater than or equal to 8.  
`supervisor2 >= 4`: Checks if the value of the supervisor2 variable is greater than or equal to 4.  
`&&`: This is the logical AND operator. It means the entire condition is only true if both the left side and the right side are true. The code within the `{...}` will only run if the student has had at least 8 guidance sessions with Supervisor 1 AND at least 4 sessions with Supervisor 2.

3. What is the student eligibility process from start to finish? Explain it in detail for all conditions.

The eligibility process follows these steps:

**Check Compensation:** The system first asks if the student is cleared of compensation.

**Initial Decision:** If the student is not cleared, they are immediately deemed ineligible, and the process stops.

**Check Guidance Logs:** If the student is cleared of compensation, the system proceeds to check the number of guidance logs with two supervisors.

**Final Decision:**

If logs with Supervisor 1 are 8 or more AND logs with Supervisor 2 are 4 or more, the student is eligible to register.

If either or both of the guidance log requirements are not met, the student is deemed ineligible, and a specific message is shown indicating which requirement was failed.

### 3. Task Assignment Completion Time: 120 minutes

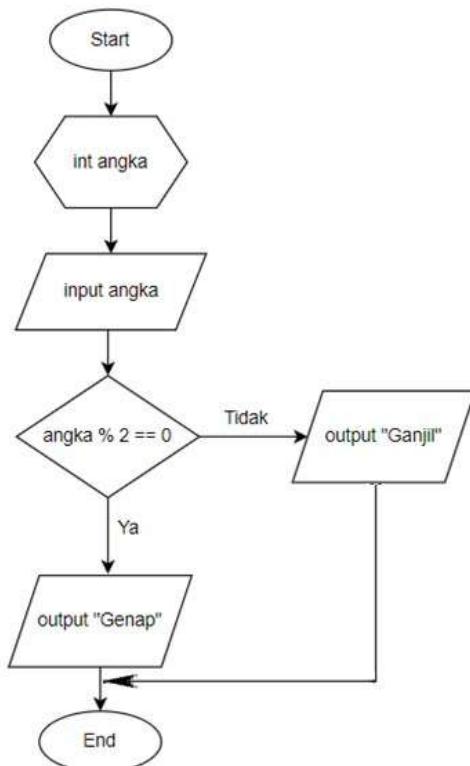
1. Reopen the file `ifCetakKRSNoPresensi.java`, add a new line in the main function to transform the modified KRS printing program using the IF-ELSE structure that has been created into the Ternary Operator form! **Commit and push your program to Github with the message “Task 1”**

```

File Edit Selection View Go Run Terminal Help ← → ⌘, infoSheet5
EXPLORER OPEN EDITORS j_ifCetakKRSNoPresensi.java 2 j_ifCetakKRSNoPresensi.java 2
C:\>java>file name>TUGAS KULIAH SEMESTER 1>DAFTAR PAK IMAM>TesPraktikumDasar>j_ifCetakKRSNoPresensi.java
import java.util.Scanner;
public class ifCetakKRSNoPresensi {
    // This class is intentionally left empty.
    Run Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Print KRS STAMA without attendance data");
        System.out.println("Do you want to print KRS (Y/N)?");
        boolean aktrumas = sc.nextLine().equalsIgnoreCase("Y");
        if (aktrumas) {
            System.out.println("Tuition payment confirmed.");
            System.out.println("KRS now can be printed.");
        } else {
            System.out.println("Tuition payment not confirmed.");
            System.out.println("KRS cannot be printed.");
        }
    }
}

```

2. Pay attention to the following flowchart .



Implement the flowchart into Java code using an IF-ELSE selection structure! Commit and push your program to GitHub with the message "Task 2"

```

ganjilgenap.java
1 package Jobsheets;
2 import java.util.Scanner;
3 public class ganjilgenap {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Enter a number: ");
7         int angka = sc.nextInt();
8         if (angka % 2 == 0) {
9             System.out.println(angka + " genap.");
10        } else {
11            System.out.println(angka + " ganjil.");
12        }
13    }
14 }
  
```

The screenshot shows a Java IDE interface with multiple tabs open. The current tab is 'ganjilgenap.java'. The code in the editor is as follows:

```

ganjilgenap.java
1 package Jobsheets;
2 import java.util.Scanner;
3 public class ganjilgenap {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.println("Enter a number: ");
7         int angka = sc.nextInt();
8         if (angka % 2 == 0) {
9             System.out.println(angka + " genap.");
10        } else {
11            System.out.println(angka + " ganjil.");
12        }
13    }
14 }
  
```

3. Implement the flowchart you created for the Exercise 5 of the Basic Programming Course regarding the library system and campus WIFI access into program code! **Commit and push your program to GitHub with the message "Assignment 3."**

The screenshot shows a Java code editor in VS Code. The left sidebar displays the file structure with several Java files listed under 'OPEN EDITORS' and 'JOBSHEETS'. The main editor area shows the code for 'WiFiAccess.java'. The code uses a Scanner to read user input and determine if WiFi access is granted based on the user type ('lecturer' or 'student') and their credits.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ 🔍 jobsheets

EXPLORER
OPEN EDITORS
JOBSHEETS
JAVA PROJECTS
OUTLINE

j iCetakKRSNoPresensi.java 2 j WiFiAccess.java 1 ×
j WiFiAccess.java > () jobsheets
1 package Jobsheets;
2 import java.util.Scanner;
3 public class WiFiAccess {
4
5     Run|Debug
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9             System.out.print("Enter user type (lecturer/student): ");
10            String userType = scanner.nextLine();
11
12            if (userType.equalsIgnoreCase("lecturer")) {
13                System.out.println("WiFi access granted (lecturer)");
14            }
15
16            else if (userType.equalsIgnoreCase("student")) {
17                // If the user is a student, ask for their credits
18                System.out.print("Input credits: ");
19                int credits = scanner.nextInt();
20
21                // 4. Check if the student has 12 or more credits
22                if (credits >= 12) {
23                    System.out.println("WiFi access granted (active student)");
24                } else {
25                    System.out.println("WiFi access denied (inactive student - credits < 12)");
26                }
27
28            // Handle cases where the input is neither "lecturer" nor "student"
29            else {
30                System.out.println("Invalid user type. Access denied.");
31            }
32        scanner.close();
33    }
34 }
```