

Selection part 1

Teaching Team of Programming Fundamentals
2023

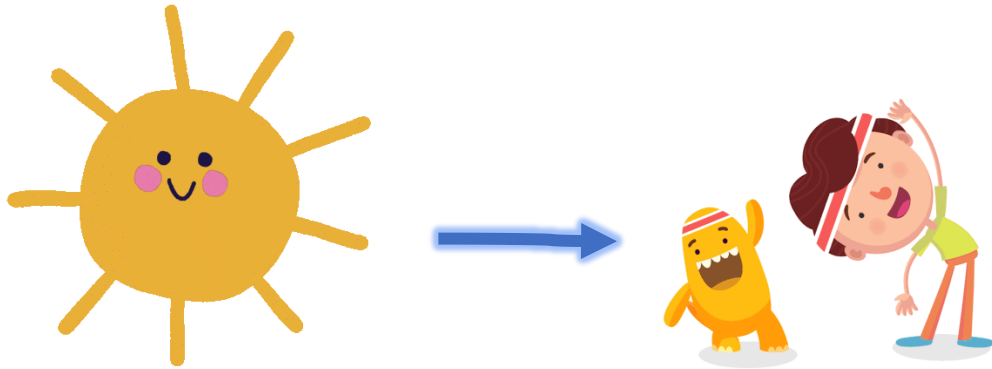
Learning Outcome

After finishing this lesson, students must be able to :

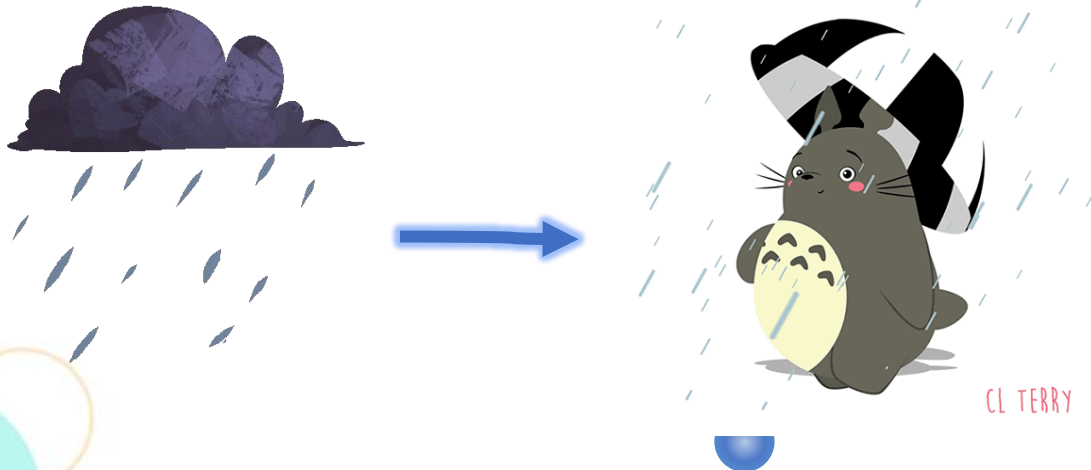
- explain the basic concept of selection,
- Implement selection algorithm to solve a specific logical problem and visualize it by creating flowchart
- Write the source code based on the flowchart to implement selection algorithm

Selection

Daily activities that use selection



If it is sunny
I will exercise outside the house

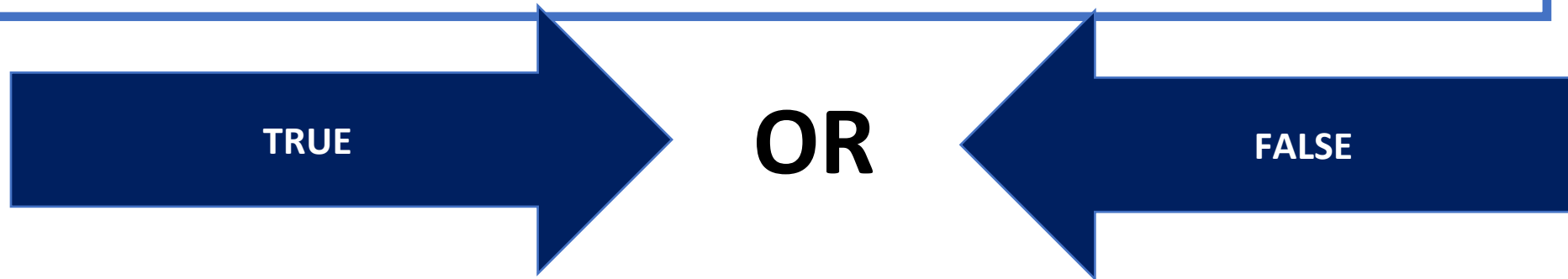


If it is raining
I will bring an umbrella

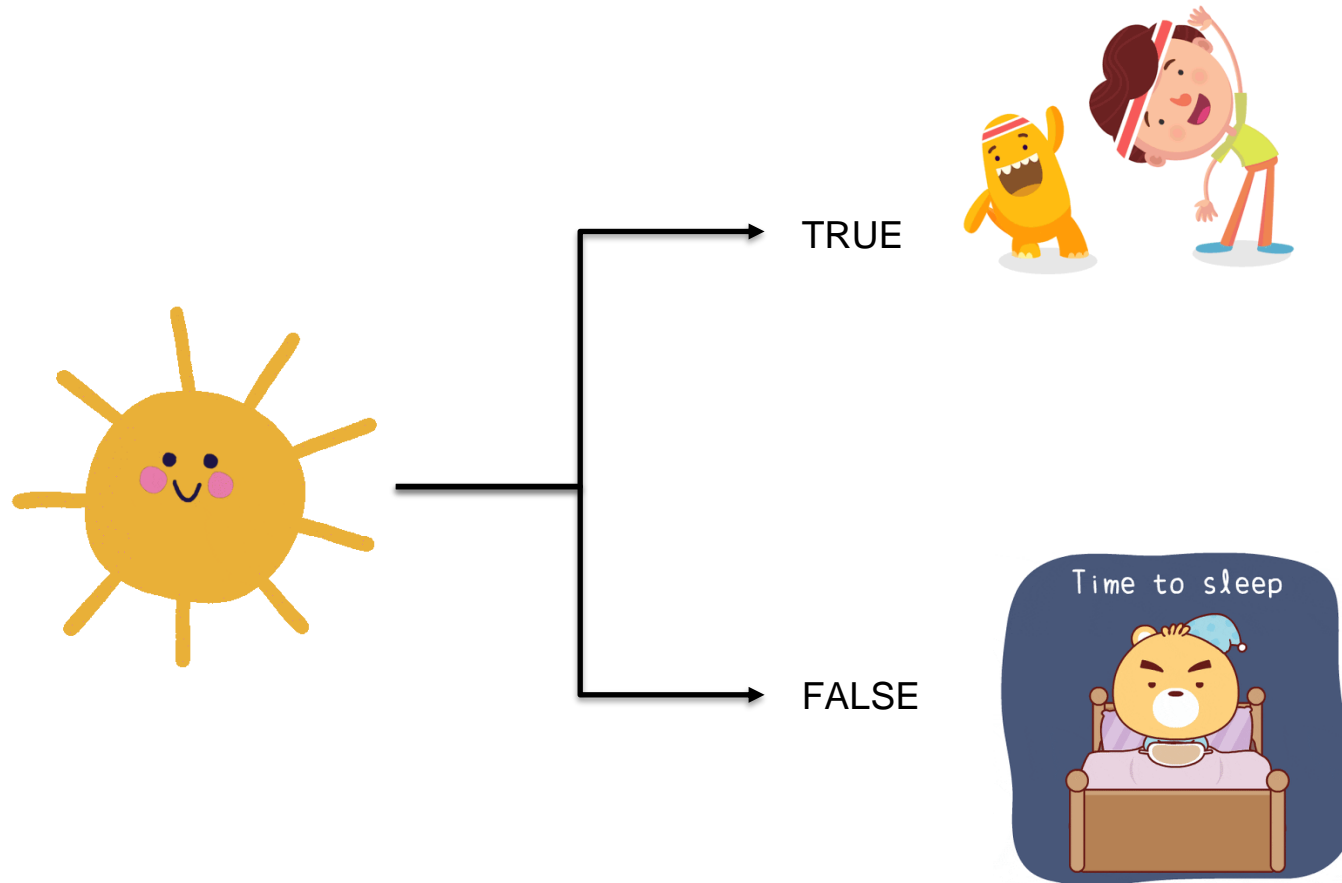
Definition

- Selection is an of basic flow in algorithm that is used to select a specific action, based on certain condition or criteria

Condition : a statement that evaluates a specific condition whether **true** or **false**

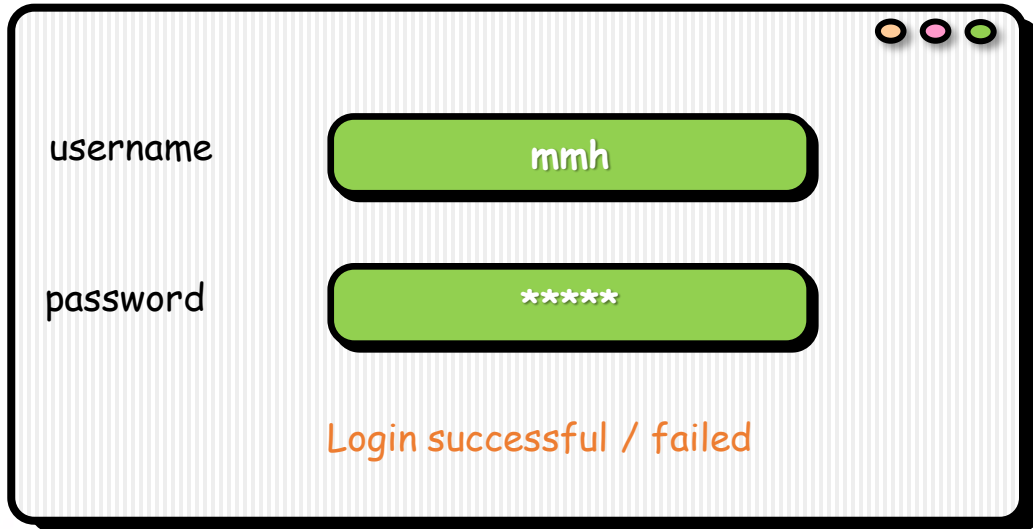


SELECTION → CONDITION



SELECTION → CONDITION

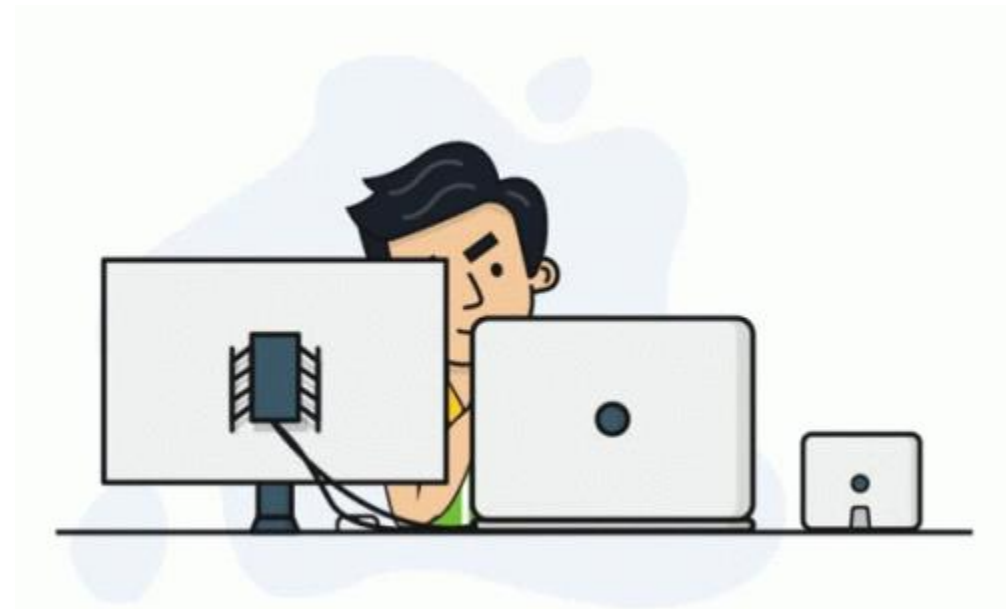
Similar to humans, computers function according to **conditions** set by programmers.



username mmh

password *****

Login successful / failed



SELECTION SYNTAXEX (STATEMENTS)

- 
- 1. IF**
 - 2. IF...ELSE**
 - 3. IF...ELSE IF...ELSE...**
 - 4. SWITCH...CASE**

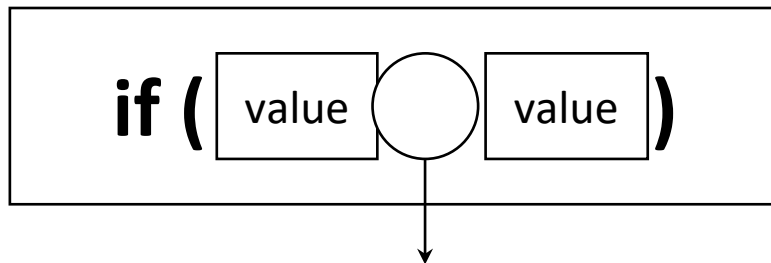
Selection Statement Structure IF....

Selection Statement: IF

- STATEMENT IF

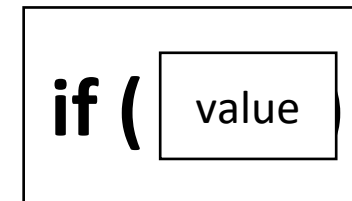
if (condition)

Condition based on 2 values
relationship



Relational Operator

1 value



Selection Statement: IF

- Basic structure:

```
if (condition) {  
    statement1;  
    statement2;  
}
```

- ✓ If the condition is **true**, then the **statement1** and **statement2** will be **executed**
- ✓ If it is **false**, then the statements will **not be executed**

Selection Statement: IF

if (value)

if (value ○ value)

Relational Operator

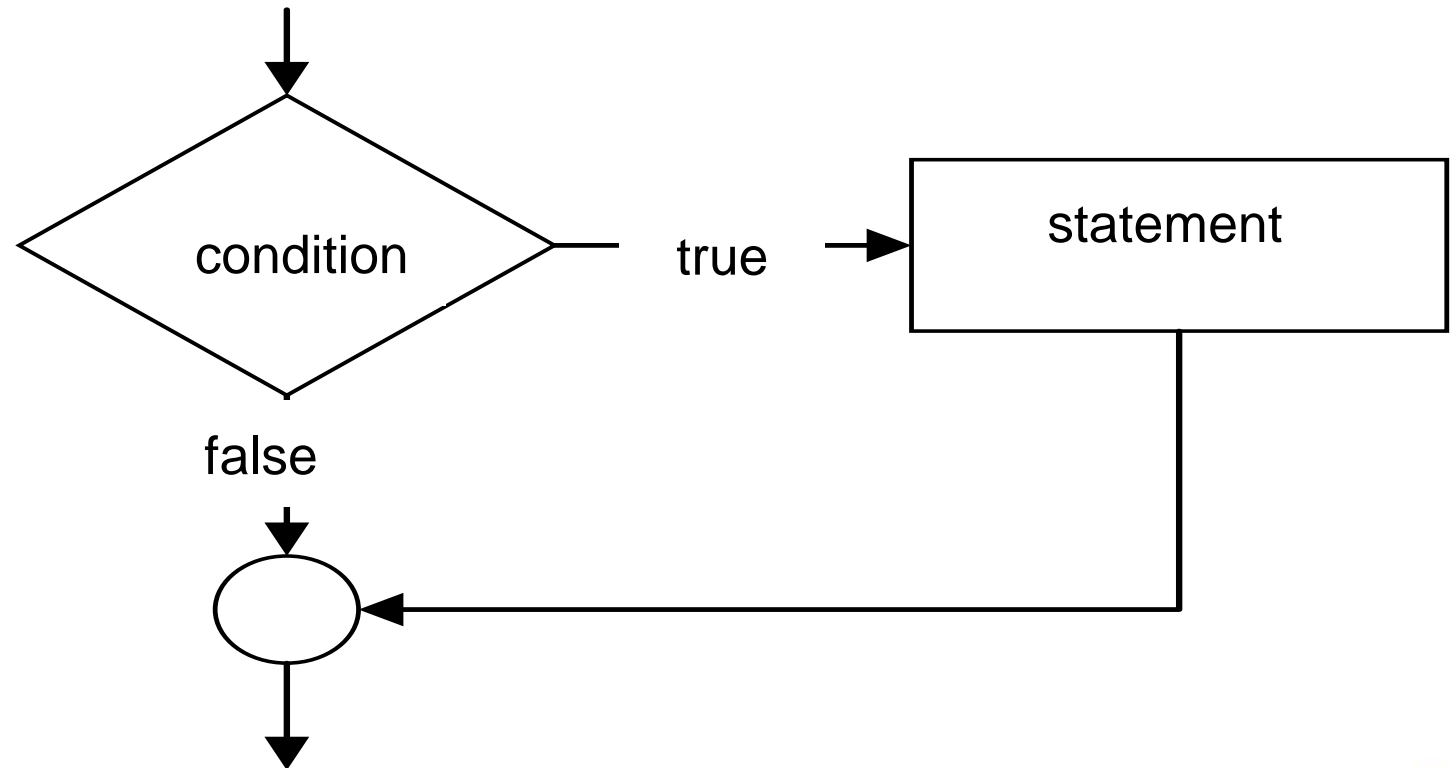
if (condition)

```
if (isLapar) {  
    Print("silakan makan")  
}
```

```
if (nilai < 70 ) {  
    Print("harus remidi")  
}
```

Must be **boolean**
(true/false)
statement

Flowchart If



Contoh Kode Program

```
1 public class IfKondisi {  
2  
3     public static void main(String[] args) {  
4         Scanner sc00 = new Scanner(System.in);  
5  
6         System.out.print("Masukkan Suhu :");  
7         int suhu = sc00.nextInt();  
8  
9         if (suhu<16){  
10             System.out.println("silakan menggunakan jaket");  
11         }  
12     }  
13 }
```

Input :
suhu = 15

Output :
"Silahkan pakai jaket"

Input :
suhu = 18

Output :

Selection Statement IF....ELSE



Selection Statement IF...ELSE

Selection Structure **IF-ELSE** at least must have **2 statements**.

If the **condition** is **TRUE**, then **statement1** will be executed. If the **condition** is **FALSE**, then **statement2** will be executed.

```
if ( condition )  
{  
    Statement1;  
}  
else  
{  
    Statement2;  
}
```

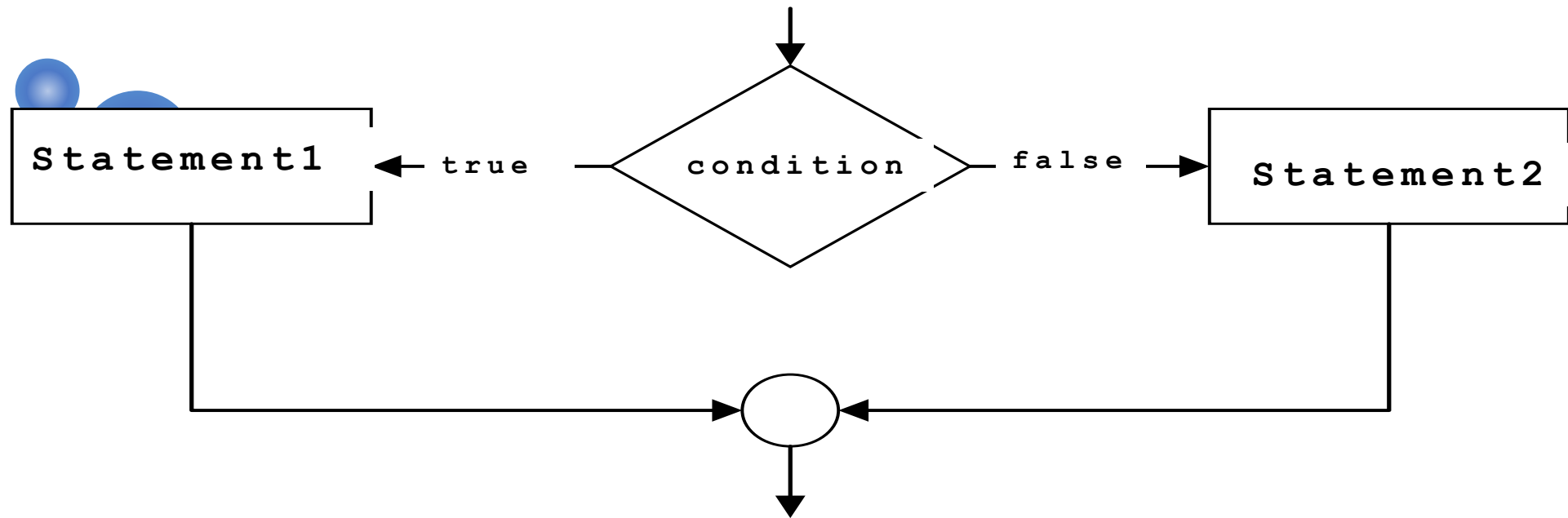


Selection Statement IF...ELSE

```
if (isHungry) {  
    Print("Please take eat!!")  
}  
else{  
    Print("Lets study!!!!")  
}
```

```
if (score < 70 ) {  
    Print("Retake!!!")  
}  
else{  
    Print("Passed!!")  
}
```


Flowchart If-Else



Example

```
public class IfKondisi {  
    Run | Debug  
    public static void main(String[] args) {  
        Scanner sc00 = new Scanner(System.in);  
  
        System.out.print(s:"Input temperature = ");  
        int temperature = sc00.nextInt();  
  
        if(temperature<16){  
            System.out.println(x:"Put on your jacket!!");  
        }else{  
            System.out.println(x:"Wear your hat!!");  
        }  
    }  
}
```

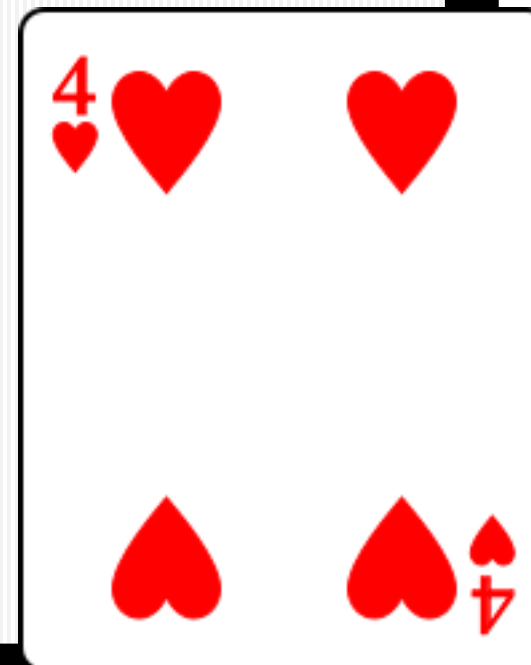
Input :
Temperature = 15

Output :
"Put on your jacket!!"

Input :
Temperature = 18

Output :
"Wear your hat!!"

```
if (cardColor=="Red" && cardValue>=5) {  
    "Clap your hand!!!"  
}  
else{  
    "Stomp your feet!!!"  
}
```



Selection Statement

IF...ELSE IF...ELSE...

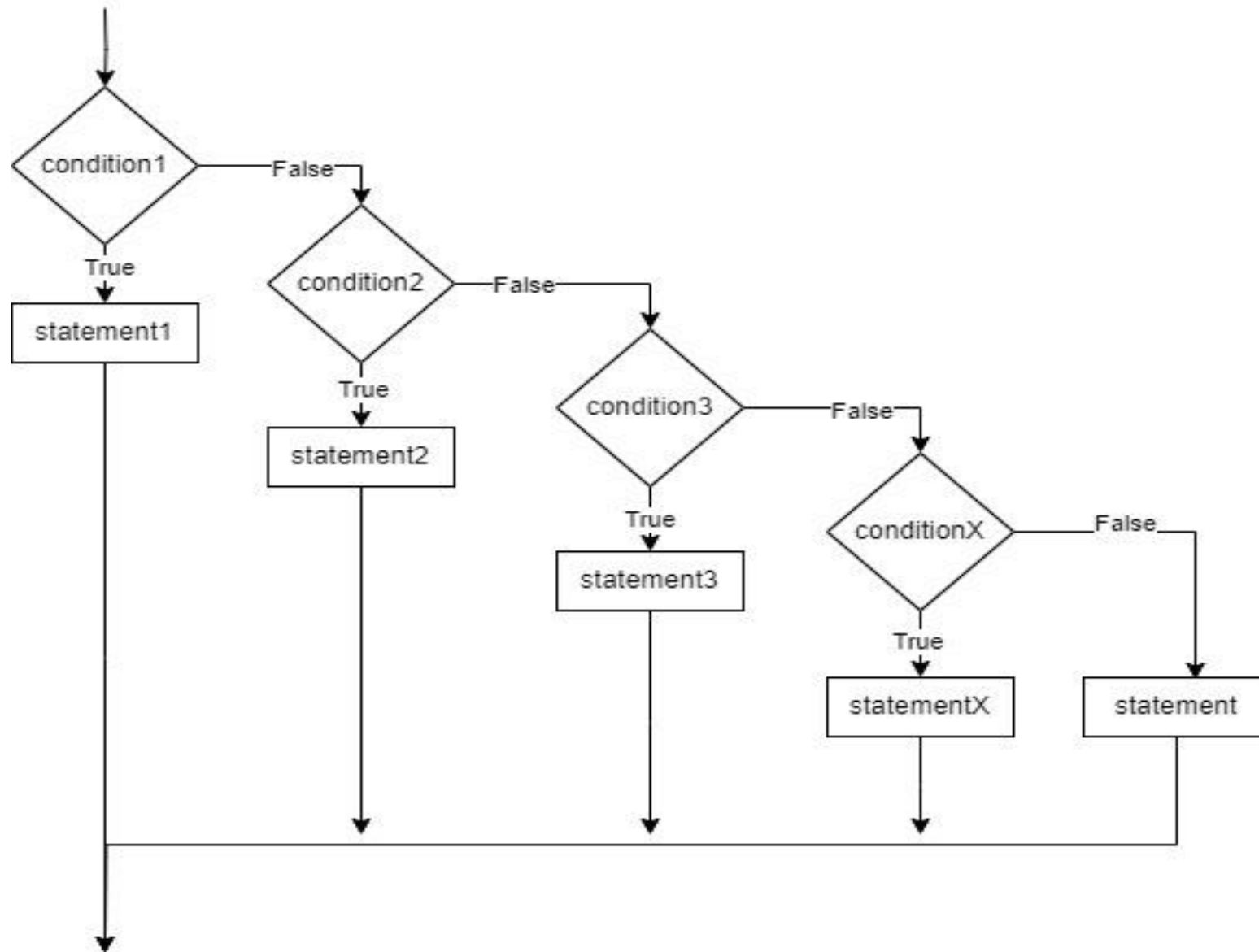
Selection Statement: If...else if...else

- Bentuk umum:

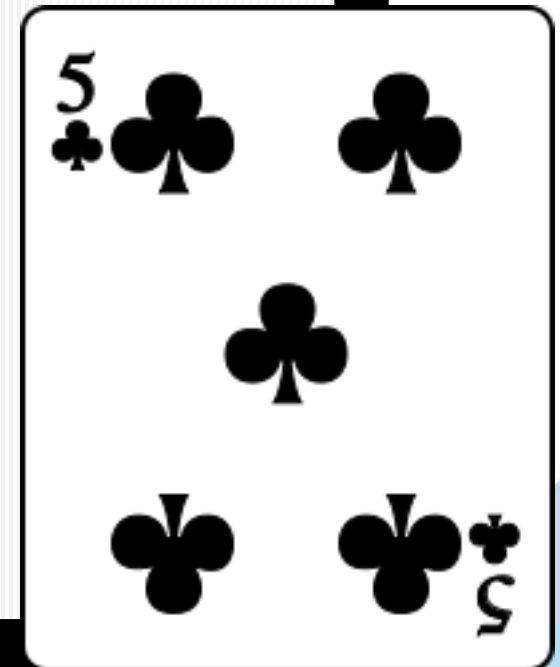
```
if (condition1) {  
    statement1;  
} else if (condition2) {  
    statement2;  
} else if (condition3) {  
    statement3;  
}  
...  
...  
else if (conditionX) {  
    statementX;  
} else {  
    statement;  
}
```

- **statement1** will be **executed** when “**condition1**” is **TRUE**.
- If “**condition1**” is **FALSE**, then “**condition2**” will be evaluated. If “**condition2**” is **TRUE** then **statement2** will be executed, and so on.
- If **all conditions** do not match (**all conditions** are **FALSE**) then it will go to **else** block and **statement** will be executed.

Flowchart



```
if (cardColor=="Red" && cardValue>=5) {  
    "Clap your hand!!!"  
}  
else if (cardColor=="Black" && cardValue>=5) {  
    "Point your friend!!!"  
}  
else{  
    "Stomp your feet!!!"  
}
```



Example

```
public class ElseIfKondisi {  
    Run | Debug  
    public static void main(String[] args) {  
        Scannerer sc00 = new Scanner(System.in);  
  
        System.out.print(s:"Input temperature = ");  
        int temperature = sc00.nextInt();  
  
        if(temperature<16){  
            System.out.println(x:"Put on your jacket!!");  
        }else if(temperature<20){  
            System.out.println(x:"Put on your warm clothes!!");  
        }else{  
            System.out.println(x:"Wear your hat!!");  
        }  
    }  
}
```

Input :
Temperature = 18

Output :
"Put on your warm clothes!!"

Selection Statement SWITCH-CASE

Selection Statement: SWITCH-CASE

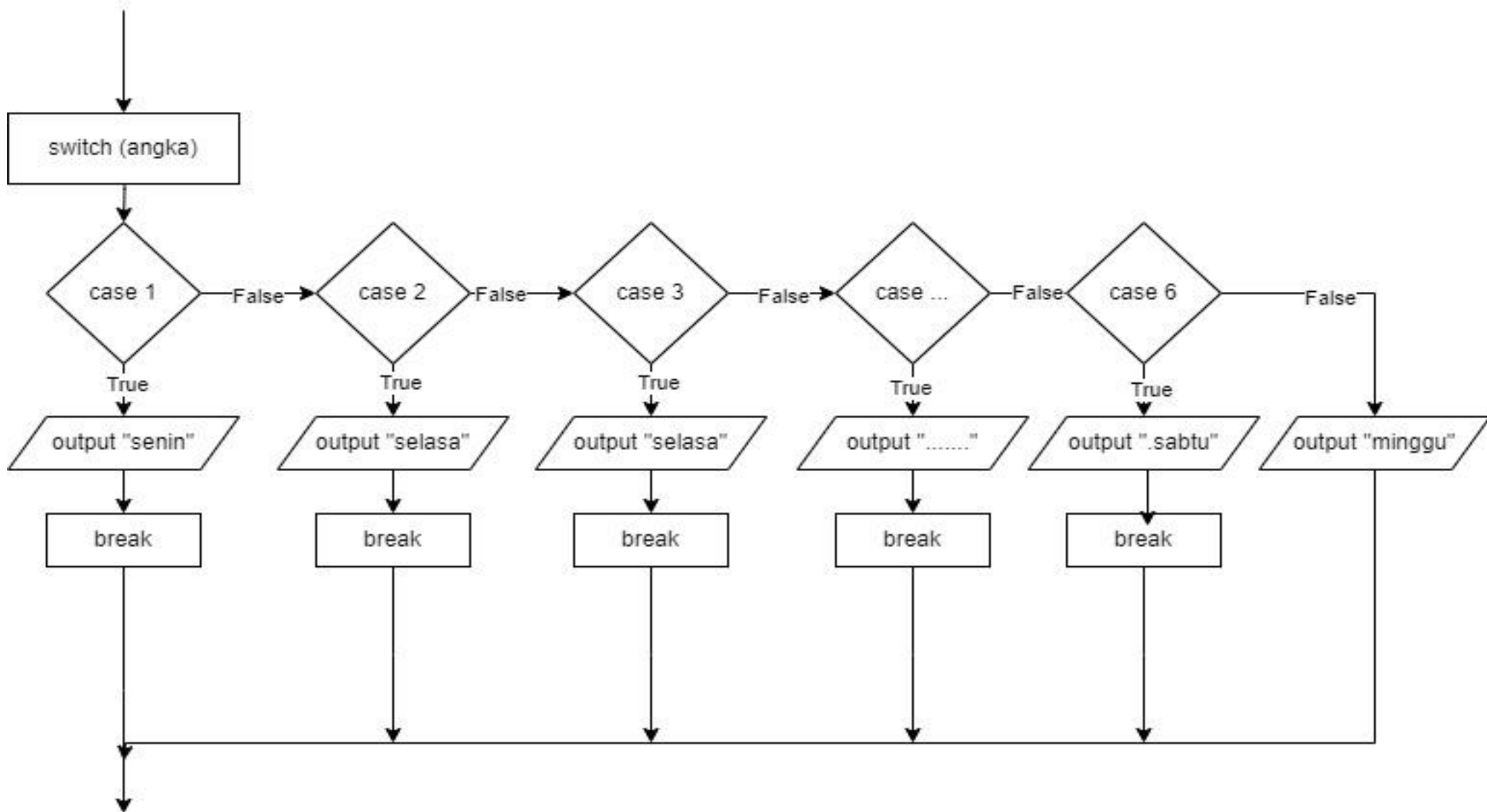
- **Basic Structure:**

```
switch (condition) {  
    case constant1:  
        statement1;  
        break;  
    case constant2:  
        statement1;  
        break;  
    ...  
    ...  
    case constantN:  
        statementN;  
        break;  
    default:  
        statement;  
}
```

Selection Statement: SWITCH-CASE

- **Switch-case** statements are used in programming when we have a **single expression** or **variable** or **condition** whose value you want to compare against **multiple possible values** or **conditions**, and we want to execute different blocks of code based on the matched condition.
- Here's how a switch-case statement works:
 - **Expression Evaluation:** The switch-case begins with an **expression** (or **variable**) that is evaluated once. This expression should produce a result that can be compared to various **cases**.
 - **Comparison:** The value produced by the **expression** is compared to the values associated with different **case** labels. Each case label represents a possible value or condition that the expression can match.
 - **Execution:** When a match is found between the expression value and one of the case labels, the code block associated with that case label is executed. This allows you to perform specific actions or logic corresponding to the matched case.
 - **Break Statement:** After executing the code block for a matched case, a break statement is typically used to exit the switch-case construct. Without a break, the control flow would "fall through" to subsequent cases, and their associated code blocks would also be executed. The break statement is used to prevent this.
 - **Default Case (Optional):** You can include an optional default case label at the end of the switch-case. If none of the case labels match the expression value, the code block associated with the default case is executed. This provides a way to handle cases that don't have a specific match.

Pemilihan Switch-Case





Example

```
public static void main(String[] args){
    Scanner sc = new Scanner(System.in);
    int angka;

    System.out.print("Masukkan angka: ");
    angka = sc.nextInt();

    switch(angka){
        case 1:
            System.out.println("Hari Senin");
            break;
        case 2:
            System.out.println("Hari Selasa");
            break;
        case 3:
            System.out.println("Hari Rabu");
            break;
        case 4:
            System.out.println("Hari Kamis");
            break;
        case 5:
            System.out.println("Hari Jumat");
            break;
        case 6:
            System.out.println("Hari Sabtu");
            break;
        case 7:
            System.out.println("Hari Minggu");
            break;

        default:
            System.out.println("Maaf, angka yang Anda masukkan salah");
    }
}
```

Ternary Operators

Ternary Operator

- This operator could be used for condition selection as well
- Basic Structure:

syntax (condition) ? (statement for true) : (statement for false)

Example

```
public static void main(String[] args) {  
    double angka=5.5;  
  
    String hasil;  
  
    if (angka>0.0){  
        hasil="Bilangan positif";  
    }  
    else{  
        hasil="Bilangan negatif";  
    }  
    System.out.println(angka+ " adalah " + hasil);  
}
```

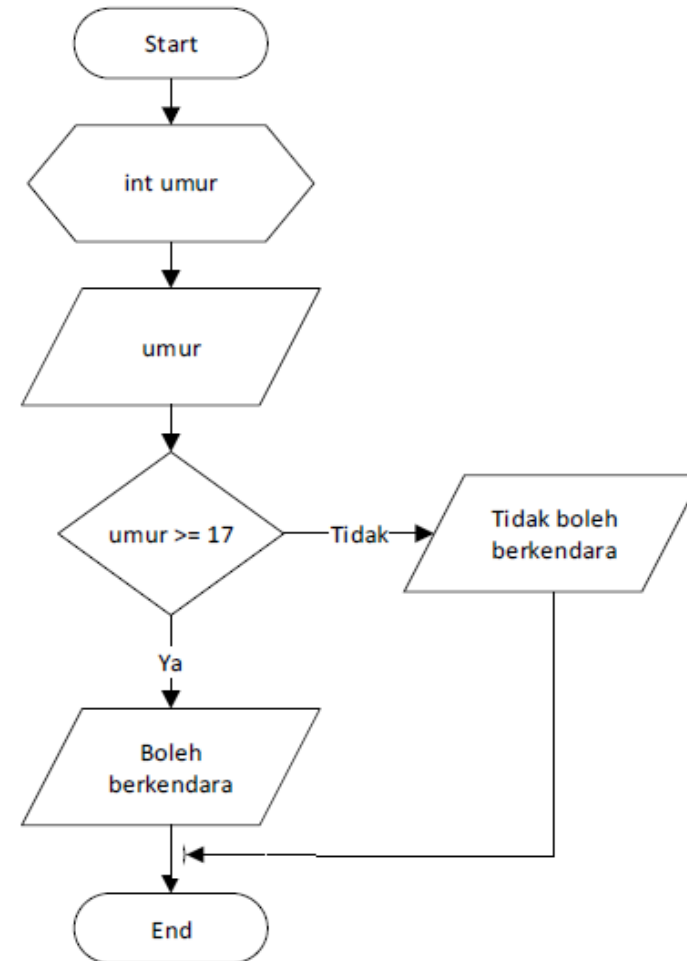
```
public static void main(String[] args) {  
  
    Double angka = 5.5;  
    String hasil;  
    hasil = (angka > 0.0) ? "Bilangan positif" : "Bilangan Negatif";  
    System.out.println(angka + " adalah " + hasil);  
}
```


Case Studies

Case Study #1

There is a rule for driving, People who are allowed to drive must be at least 17 years old

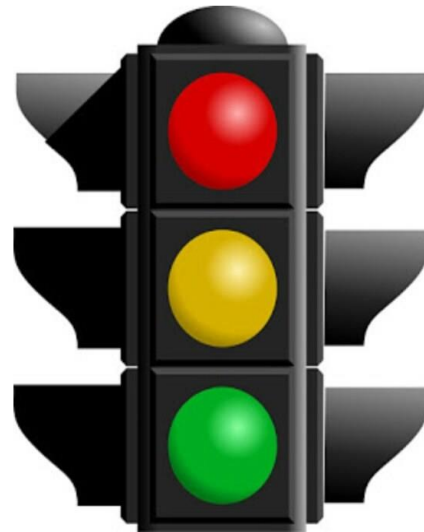
- ✓ Condition: **Usia ≥ 17**
- ✓ If the **condition** is **True**, then **Boleh berkendara**
- ✓ If the **condition** is **False**, then **Tidak Boleh berkendara**



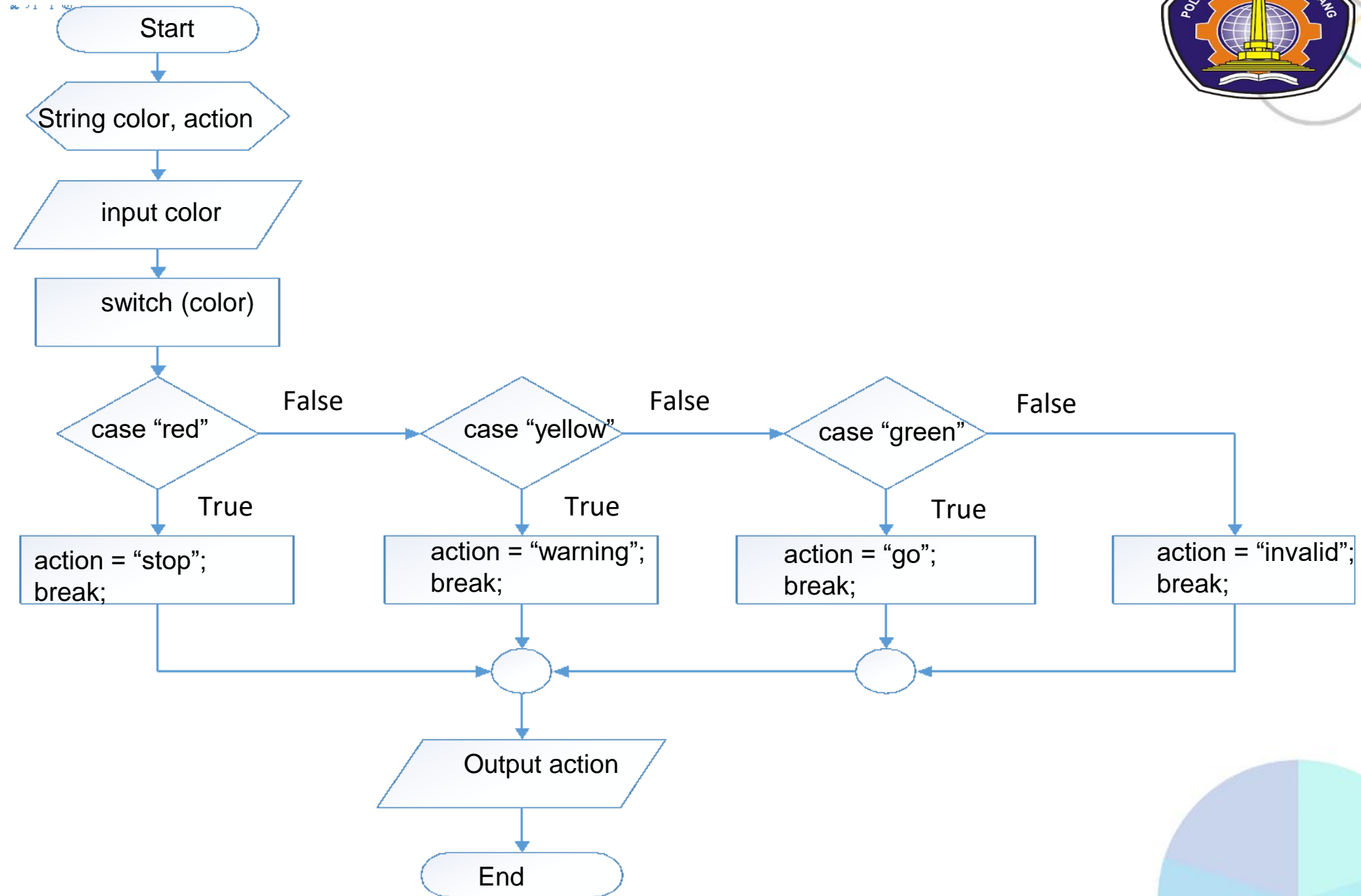
```
if (usia >= 17) {  
    System.out.print("Boleh Berkendara")  
}  
else {  
    System.out.print("Tidak Boleh Berkendara")  
}
```

Case Study #2

Create an algorithm that illustrates the rule of traffic lights.



Jawab



Any Question???

Assignment

Self Learning #1

David Martinez is a teenager currently studying an FPS (First Person Shooter) game tutorial. In the tutorial, he is given instructions on how to use melee weapons and ranged weapons. Melee weapons are used for close combat, typically within a range of 5 meters or less. On the other hand, ranged weapons are employed for distances ranging from 5 meters to over 1000 meters. Create a flowchart that explains the use of both types of weapons!

Self Learning #2

A system must ensure that users who access it are authenticated, so the system requires a login feature to verify the users entering the system. Inputs for this system are the username and password. If the username and password match what is stored in the system, the user can access the system. However, if the password and username do not match, the system will display an 'incorrect username or password' warning. Create a flowchart for this scenario!

ASSIGNMENT

1. Identify, according to each project, which features require the concept of selection!
2. Determine the type of selection to be used, as well as the required conditions for each!
3. Create an algorithm in the form of a flowchart as needed, based on tasks No. 1 and 2 that you have identified!



Good Luck!