



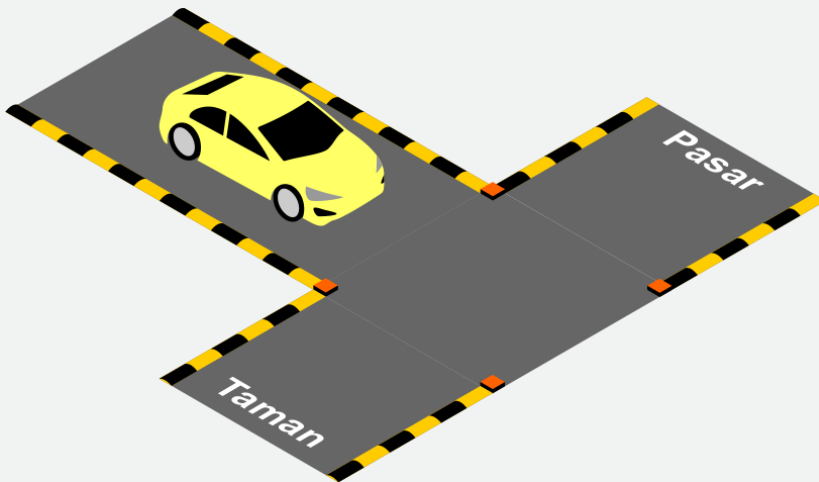
PERCABANGAN

Fundamen Pengembangan Aplikasi



Konsep Percabangan

- Percabangan *adalah* instruksi untuk **mengendalikan aliran proses** ketika terdapat dua atau beberapa instruksi yang dapat **dipilih** berdasarkan suatu **kondisi**.
- Dapat diterapkan pada **analisis kasus** atau **perulangan**
- Pembahasan untuk sesi ini terbatas pada analisis kasus



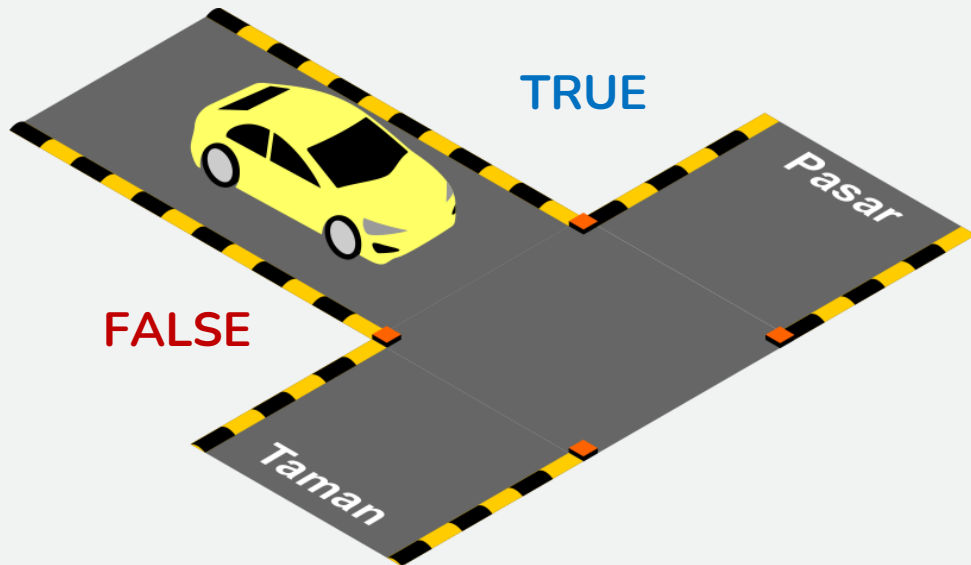
1. Ada Pilihan
2. Ada Kondisi

Kondisi dapat dinyatakan sebagai **sebuah nilai** atau sebuah **ekspresi** bertipe **Boolean** (dapat bernilai true atau false)

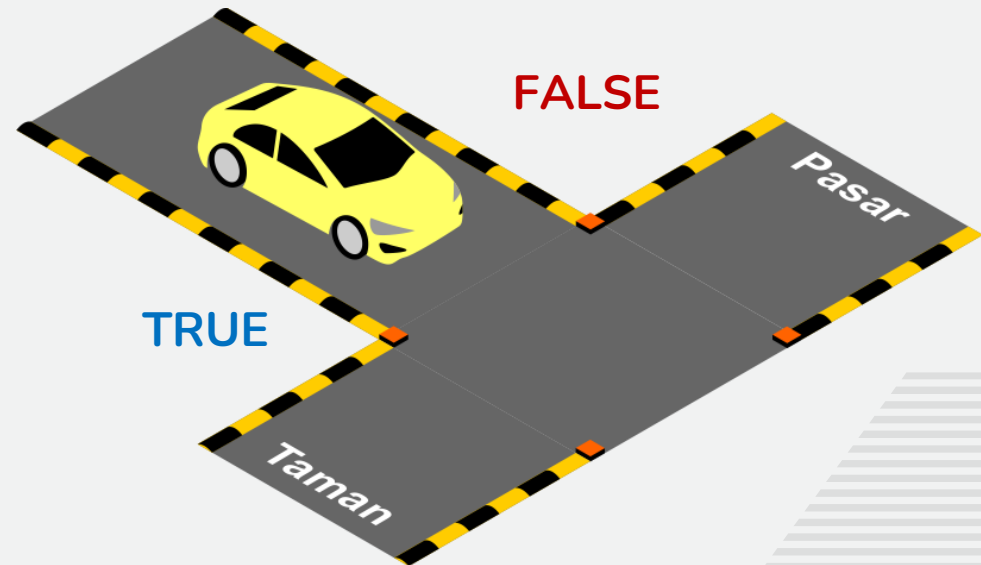
Konsep Percabangan

- Kondisi dibangun sesuai dengan kebutuhan

True → ke Pasar
False → ke Taman



True → ke Taman
False → ke Pasar



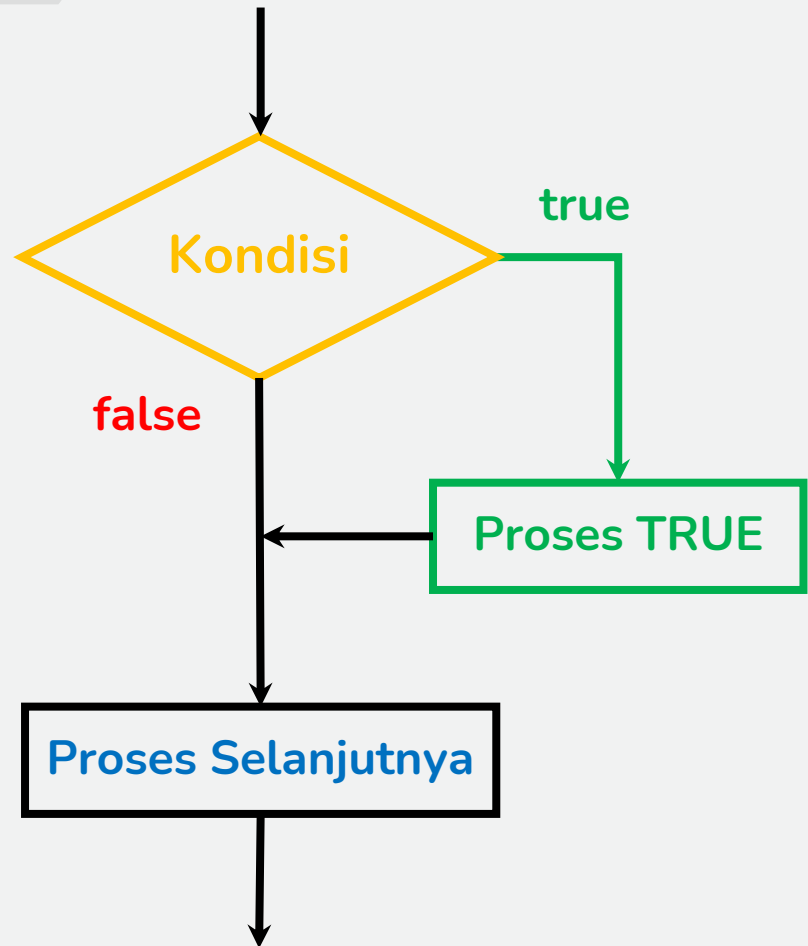
Percabangan IF



Percabangan IF

- Percabangan IF akan **mengeksekusi** suatu proses jika **kondisinya true** dan **tidak akan mengeksekusi proses** tersebut jika **kondisinya false**
- Misal, apakah Anda mau makan?
 - Jika **TRUE**, ambil makanan
 - Jika **FALSE**, diam (tidak melakukan apa-apa)
- 1 Kondisi
- 1 Pilihan

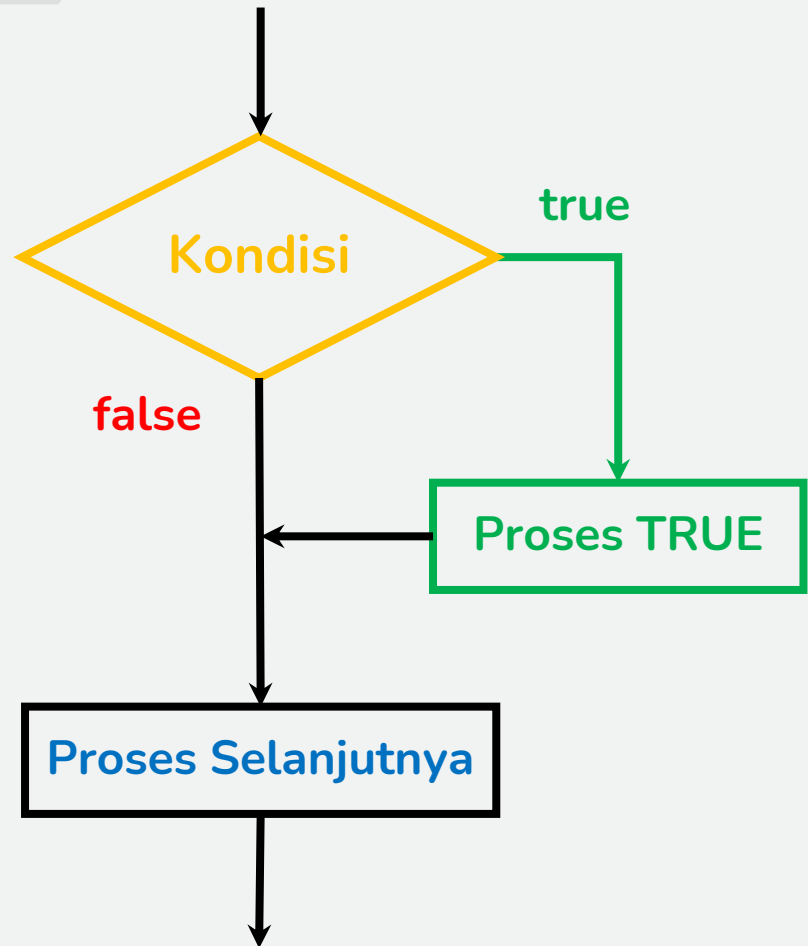
Flowchart Percabangan IF



Ketika ada kondisi percabangan:

- Jika kondisinya **true**, akan dilakukan **Proses TRUE**
- Jika kondisinya **false**, tidak dilakukan proses apapun, (**lompat ke Proses Selanjutnya**)
- **Proses Selanjutnya PASTI** akan dieksekusi, baik itu kondisinya **true** maupun **false** (sudah keluar dari percabangan)

Flowchart Percabangan IF



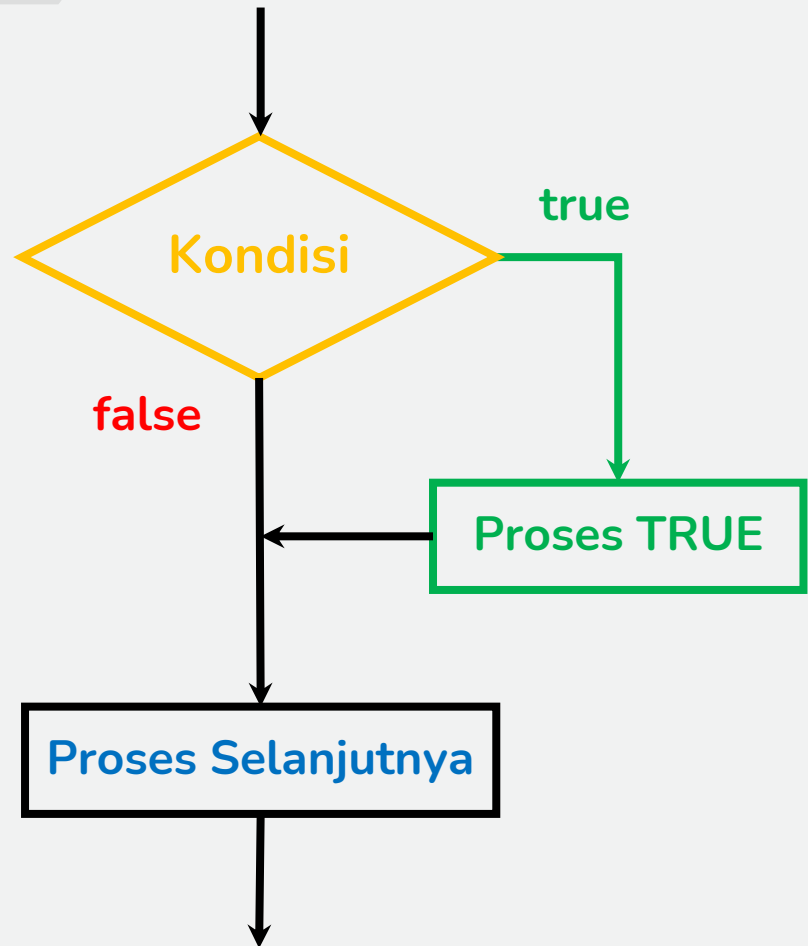
Kasus Percabangan:

Jika total pembelian lebih dari 10.000,
akan terdapat potongan harga sebesar
500

Logika Percabangan:

Jika $(total > 10000)$,
 $total \leftarrow total - 500$

Flowchart Percabangan IF



Kasus Percabangan:

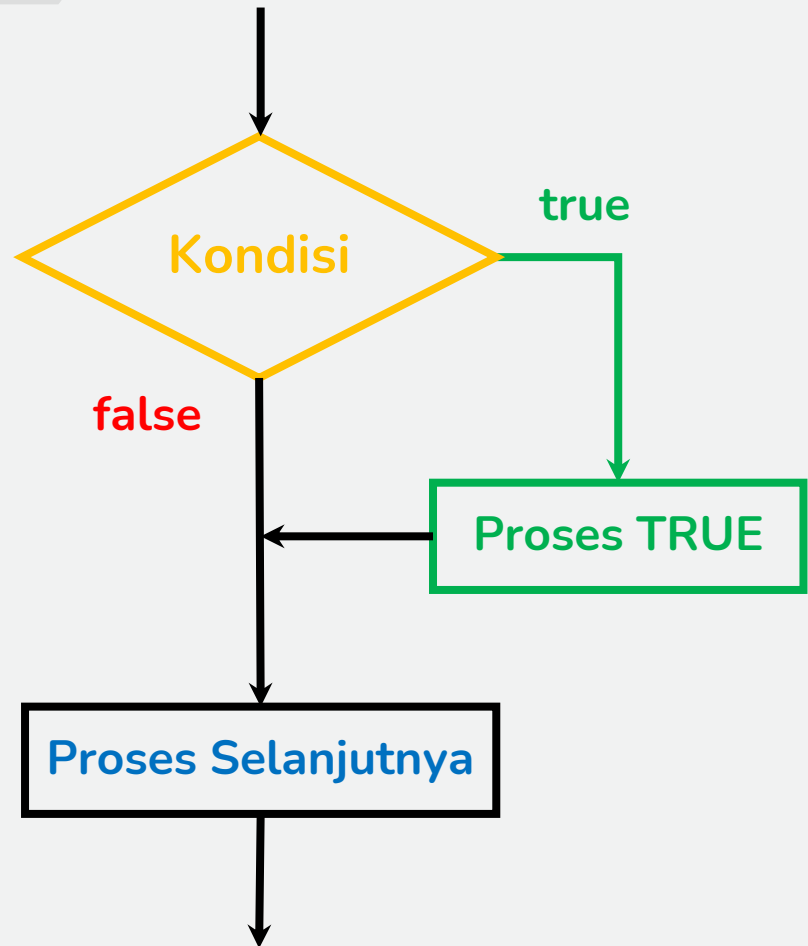
- Jika total pembelian lebih dari 10.000, terdapat potongan harga sebesar 500.
- Setiap transaksi dikenakan biaya admin sebesar 100.

Logika Percabangan:

Jika ($\text{total} > 10000$),
 $\text{total} \leftarrow \text{total} - 500$

$\text{total} \leftarrow \text{total} + 100$

Flowchart Percabangan IF



Logika Percabangan:

Jika $(total > 10000)$,
 $total \leftarrow total - 500$

$total \leftarrow total + 100$

Kode Program:

```
if (total > 10000) {  
    total -= 500;  
}
```

$total += 100;$

Blog Program
(Kurung Kurawal)

Percabangan IF-ELSE



Percabangan IF-ELSE

- Percabangan IF - ELSE mirip dengan percabangan IF, hanya saja pada percabangan IF - ELSE, **kondisi false** juga akan melakukan **eksekusi terhadap suatu proses**
- Misal, apakah Anda **mau makan**?
 - Jika **TRUE**, **ambilkan makanan**
 - Jika **FALSE**, **ambilkan minuman**
- 1 Kondisi
- 2 Pilihan

Percabangan IF-ELSE

Kasus percabangan:

Apakah Anda Dosen atau Mahasiswa?

- Jika true, ikut kuliah
- Jika false, bermain

Logika percabangan:

Jika Dosen atau Mahasiswa bernilai true,
ikut kuliah

Selain itu
bermain

Percabangan IF-ELSE

Kasus percabangan:

Apakah **Genap** dan **Lebih dari 0**?

- Jika **true**, **Bilangan Genap Positif**
- Jika **false**, **Bukan Bilangan Genap Positif**

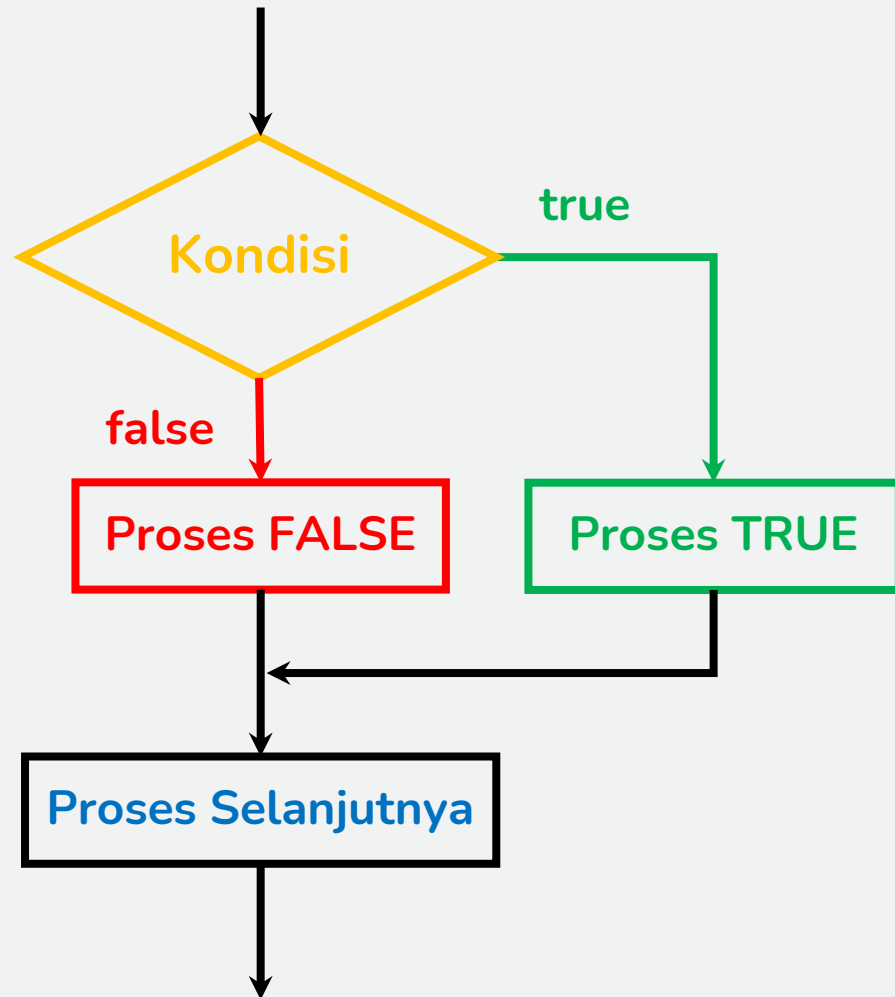
Logika percabangan:

Jika **Genap** dan **Lebih dari 0** bernilai **true**,
Bilangan Genap Positif

Selain itu

Bukan Bilangan Genap Positif

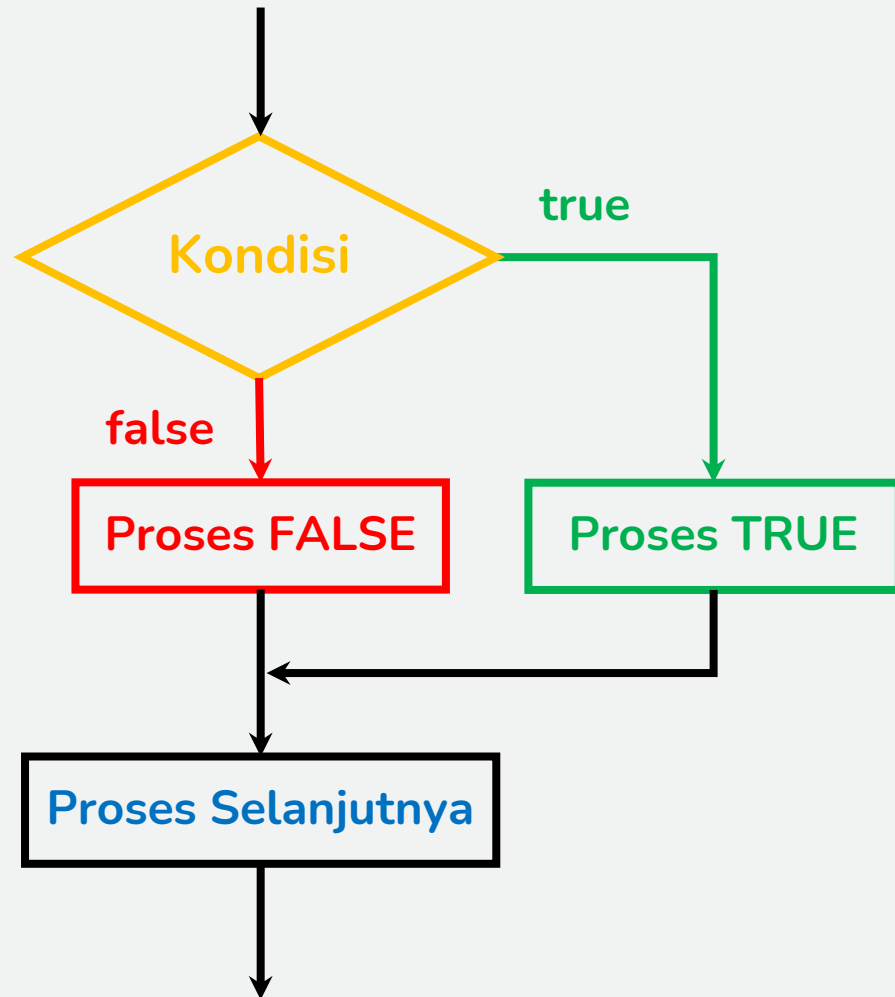
Percabangan IF-ELSE



Ketika ada kondisi percabangan

- Jika kondisinya **true**, akan dilakukan **Proses TRUE**
- Jika kondisinya **false**, akan dilakukan **Proses FALSE**
- **Proses Selanjutnya** akan dieksekusi, baik itu kondisinya **true** maupun **false** (sudah keluar dari percabangan)

Percabangan IF-ELSE



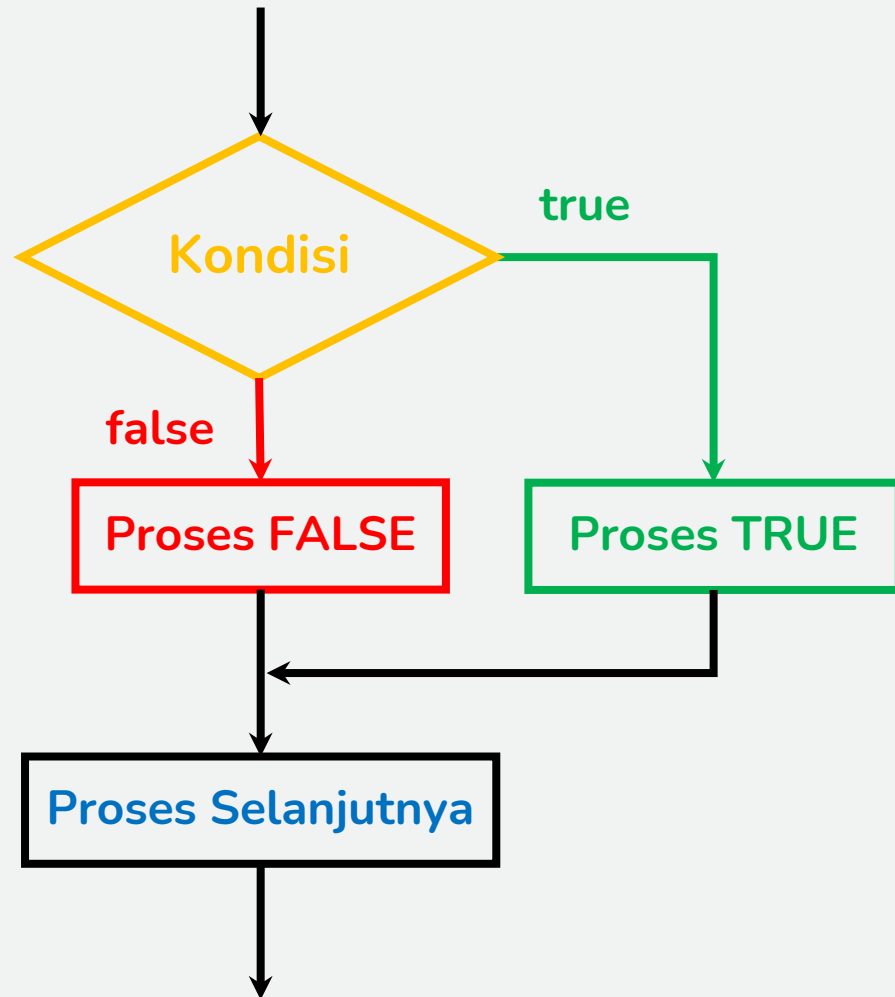
Kasus Percabangan:

- Jika total pembelian lebih dari 10.000, terdapat potongan 500.
- Jika tidak, potongan harganya sebesar 100.

Logika Percabangan:

Jika $(\text{total} > 10000)$,
 $\text{total} \leftarrow \text{total} - 500$
Selain itu
 $\text{total} \leftarrow \text{total} - 100$

Percabangan IF-ELSE



Logika Percabangan:

Jika $(total > 10000)$,
 $total \leftarrow total - 500$
Selain itu
 $total \leftarrow total - 100$

Kode Program:

```
if (total > 10000) {  
    total -= 500;  
}  
else {  
    total -= 100;  
}
```

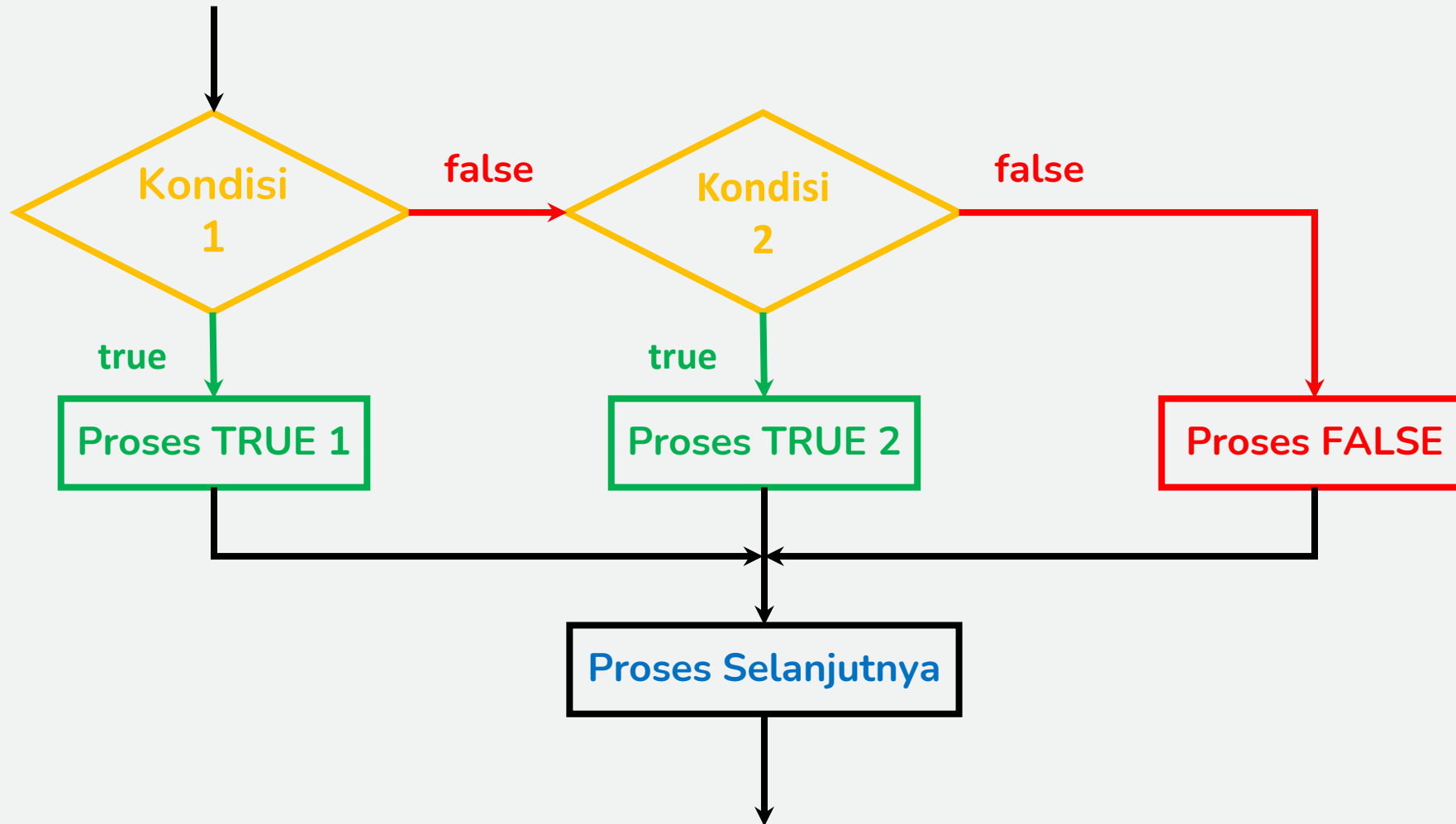

Percabangan IF-ELSE IF



Percabangan IF-ELSE IF

- Mirip dengan percabangan IF - ELSE, hanya saja pada bagian ELSE-nya bisa **ditambahkan** kondisi yang independen
- Misal, apakah Anda **Dosen**?
 - Jika **true**, **mengajar**
 - Jika **false**, apakah Anda **Mahasiswa**?
 - Jika **true**, **belajar**
- Lebih dari 1 kondisi

Percabangan IF-ELSE IF



Kode Program Percabangan IF-ELSE IF

```
if (<kondisi 1>){  
    <Proses TRUE 1>  
}
```

Kasus Percabangan:

- Jika **pekerjaannya Dosen**, akan dituliskan “**Mengajar**”, selain itu, jika pekerjaannya Mahasiswa, akan dituliskan “Belajar”, selain itu semua akan dituliskan “Bermain”.

Kode Program Percabangan IF-ELSE IF

```
if (<kondisi 1>){  
    <Proses TRUE 1>  
}  
else if (<kondisi 2>) {  
    <Proses TRUE 2>  
}
```

Kasus Percabangan:

- Jika pekerjaannya Dosen, akan dituliskan “Mengajar”, selain itu, jika pekerjaannya Mahasiswa, akan dituliskan “Belajar”, selain itu semua akan dituliskan “Bermain”.

Kode Program Percabangan IF-ELSE IF

```
if (<kondisi 1>){  
    <Proses TRUE 1>  
}  
else if (<kondisi 2>) {  
    <Proses TRUE 2>  
}  
else {  
    <Proses FALSE>  
}
```

Kasus Percabangan:

- Jika pekerjaannya Dosen, akan dituliskan “Mengajar”, selain itu, jika pekerjaannya Mahasiswa, akan dituliskan “Belajar”, selain itu semua akan dituliskan “Bermain”.

Kode Program Percabangan IF-ELSE IF

```
if (pekerjaan == "Dosen"){  
    System.out.println("Mengajar");  
}  
else if (pekerjaan == "Mahasiswa") {  
    System.out.println("Belajar");  
}  
else {  
    System.out.println("Bermain");  
}
```

Membandingkan Nilai String (alt-1)

- Untuk **membandingkan** nilai dua buah **String**, kita dapat menggunakan method **equals()** yang akan menghasilkan nilai **true** jika **sama** dan **false** jika **berbeda**.

- Contoh:

```
String kata1 = "Coba";
```

```
String kata2 = "Tes";
```

```
System.out.println(kata1.equals("Coba"));    // true
```

```
System.out.println(kata1.equals(kata2));    // false
```


Membandingkan Nilai String (alt-2)

- Untuk **membandingkan** nilai dua buah **String**, kita dapat menggunakan operator **==** yang akan menghasilkan nilai **true** jika **sama** dan **false** jika **berbeda**.

- **Contoh:**

```
String kata1 = "Coba";  
String kata2 = "Tes";
```

```
System.out.println(kata1 == "Coba"); // true  
System.out.println(kata1 == kata2);  // false
```

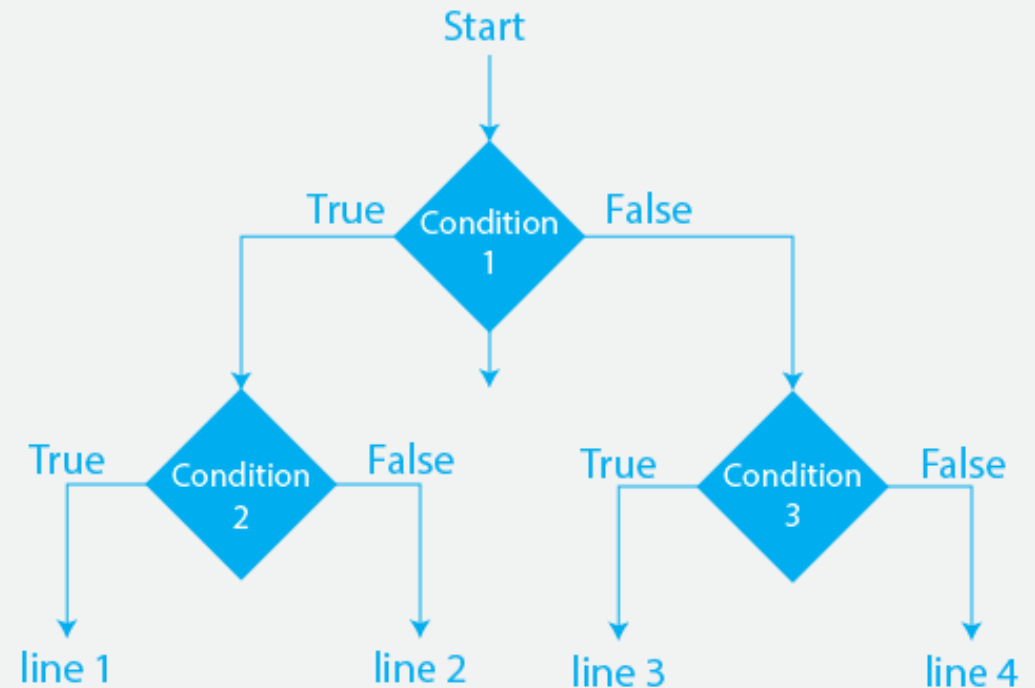
Untuk membandingkan String yang didapat dari proses input menggunakan library **Scanner**, silakan gunakan method **equals**

PERCABANGAN BERSARANG

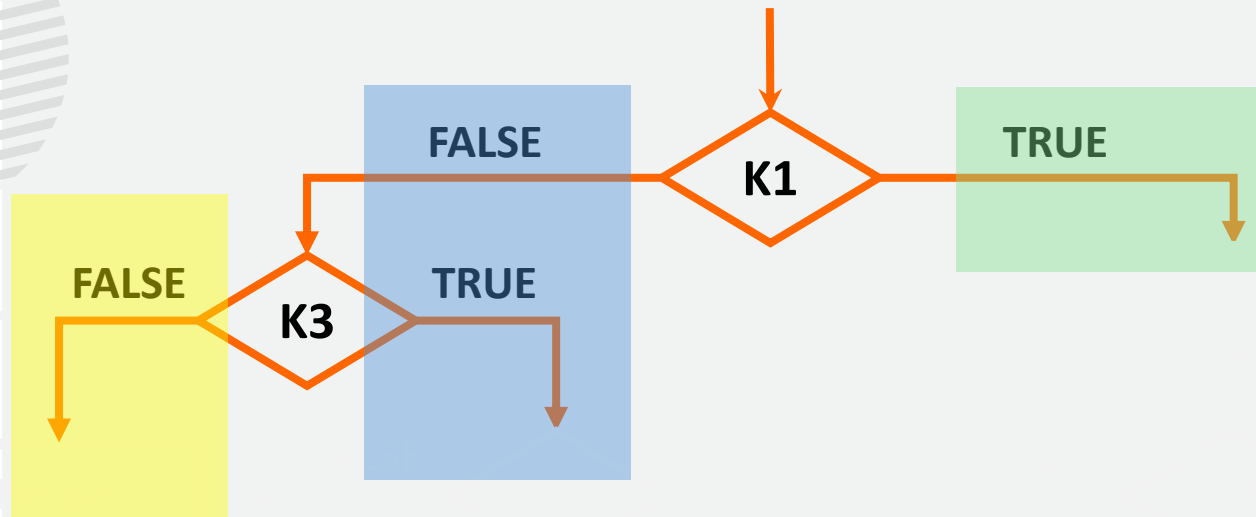


Percabangan Bersarang

- Percabangan bersarang merupakan suatu **konsep** yang dibangun dari **kombinasi percabangan-percabangan** yang telah dipelajari sebelumnya
- Dikatakan “**bersarang**” karena konsep percabangan ini memungkinkan adanya **percabangan di dalam suatu percabangan**
- Konsep percabangan bersarang memungkinkan kita untuk membangun percabangan dengan **lebih fleksibel**



Flowchart Percabangan Bersarang

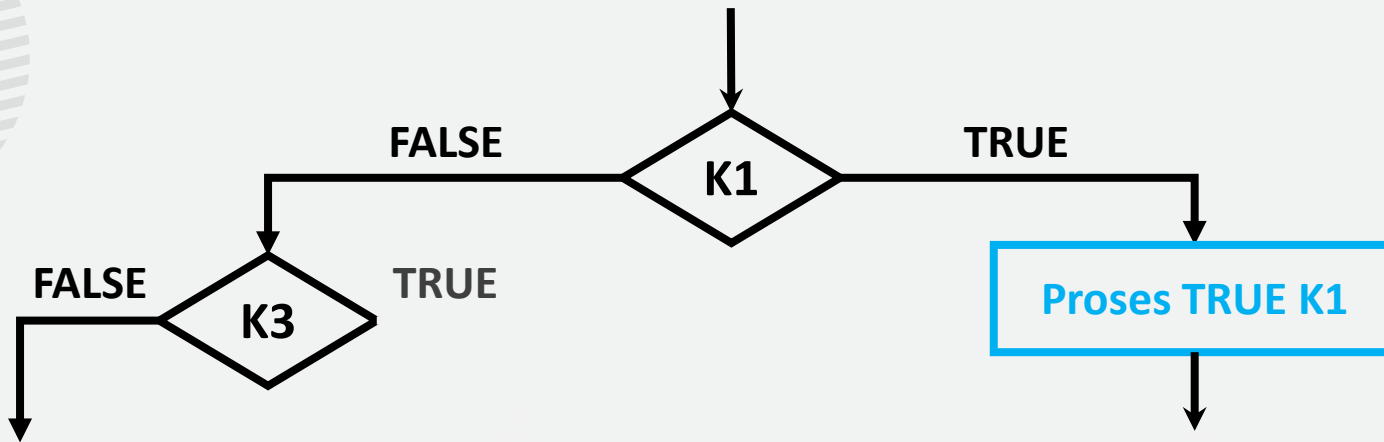


```
if (<K1>) {
```

```
}  
else if (<K3>) {
```

```
}  
else {  
    <Proses FALSE K3>;  
}
```

Flowchart Percabangan Bersarang



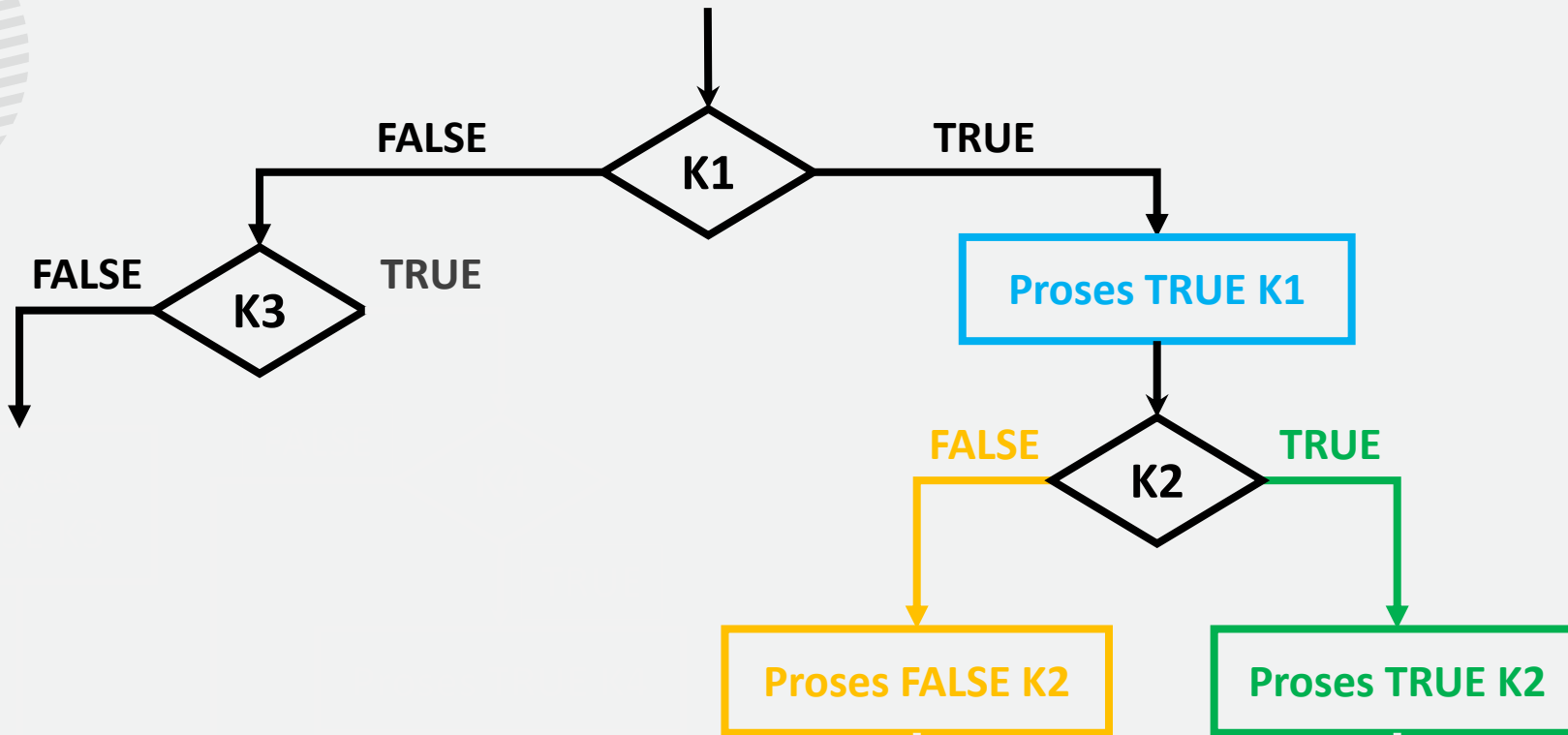
```
if (<K1>) {  
    <Proses TRUE K1>;  
}
```

```
}  
else if (<K3>) {
```

```
}  
else {
```

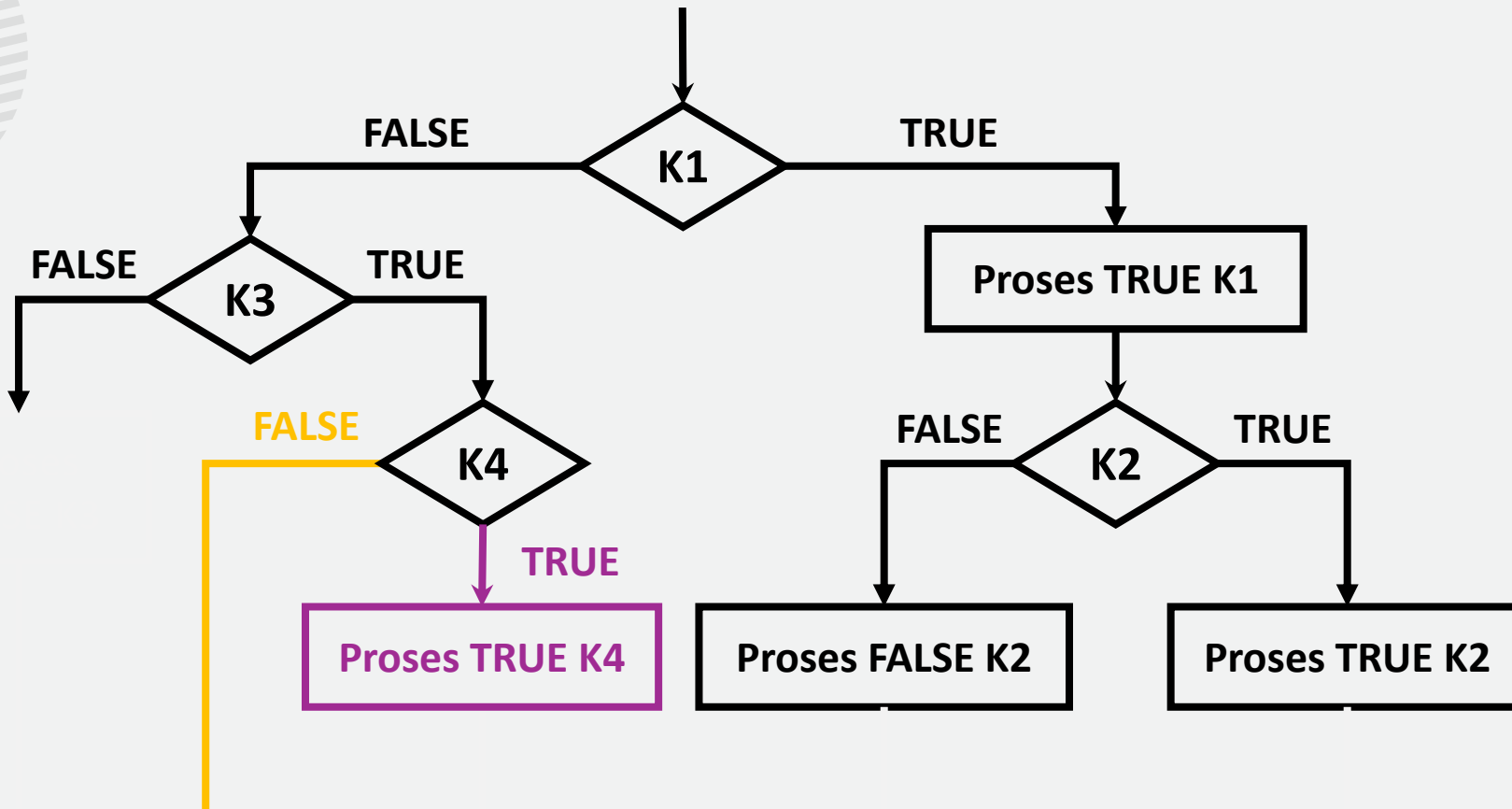
```
}
```

Flowchart Percabangan Bersarang



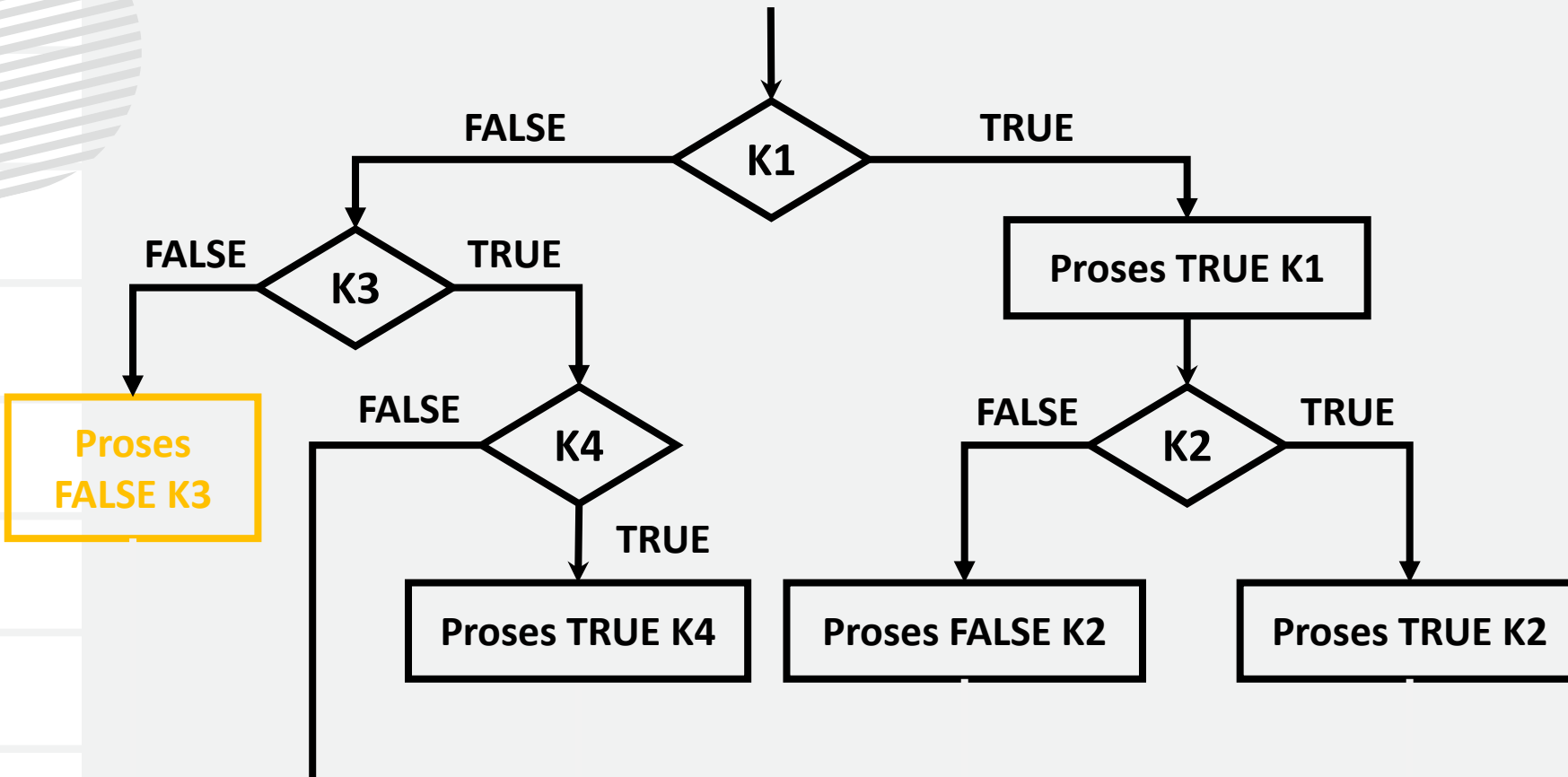
```
if (<K1>) {  
    <Proses TRUE K1>;  
    if (<K2>) {  
        <Proses TRUE K2>;  
    }  
    else {  
        <Proses FALSE K2>;  
    }  
}  
else if (<K3>) {  
  
}  
else {  
  
}
```

Flowchart Percabangan Bersarang



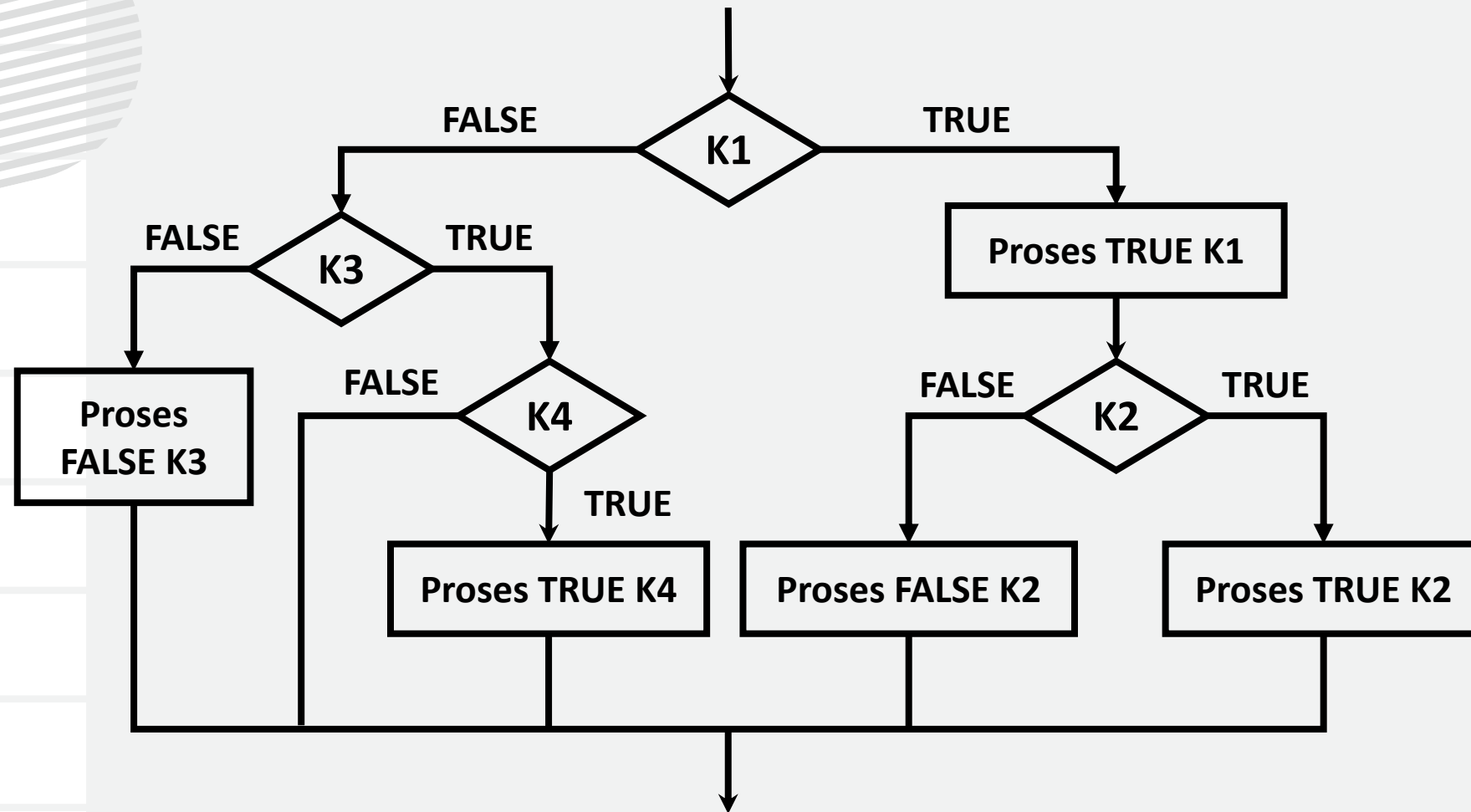
```
if (<K1>) {  
    <Proses TRUE K1>;  
    if (<K2>) {  
        <Proses TRUE K2>;  
    }  
    else {  
        <Proses FALSE K2>;  
    }  
}  
else if (<K3>) {  
    if (<K4>) {  
        <Proses TRUE K4>;  
    }  
    else {  
        }  
}
```

Flowchart Percabangan Bersarang



```
if (<K1>) {  
    <Proses TRUE K1>;  
    if (<K2>) {  
        <Proses TRUE K2>;  
    }  
    else {  
        <Proses FALSE K2>;  
    }  
}  
else if (<K3>) {  
    if (<K4>) {  
        <Proses TRUE K4>;  
    }  
    else {  
        <Proses FALSE K3>;  
    }  
}  
else {  
    <Proses FALSE K3>;  
}
```


Flowchart Percabangan Bersarang



```
if (<K1>) {  
    <Proses TRUE K1>;  
    if (<K2>) {  
        <Proses TRUE K2>;  
    }  
    else {  
        <Proses FALSE K2>;  
    }  
}  
else if (<K3>) {  
    if (<K4>) {  
        <Proses TRUE K4>;  
    }  
    else {  
        <Proses FALSE K3>;  
    }  
}
```

Contoh Kasus Percabangan Bersarang

- Memasukkan satu angka (boleh positif atau negatif)
- Jika angka bernilai 0 (nol), muncul teks “Angka nol”
- Jika angka positif, muncul teks “Bilangan Positif”
- Jika angka positif antara 1 – 9, muncul teks “Bilangan Satuan”
- Jika angka positif antara 10 – 99, muncul teks “Bilangan Puluhan”
- Jika angka di luar rentang 1 – 99, muncul teks “Bukan Bilangan Satuan atau Puluhan”
- Jika angka negatif, muncul teks “Bilangan Negatif”
- Jika angka negatif merupakan bilangan genap, muncul teks “Bilangan Genap”
- Jika angka negatif merupakan bilangan ganjil, muncul teks “Bilangan Ganjil”

Contoh Kasus Percabangan Bersarang

- Memasukkan satu angka (boleh positif atau negatif)
- Jika angka bernilai 0 (nol), muncul teks “Angka nol”
- Jika angka positif, muncul teks “Bilangan Positif”
 - Jika angka positif antara 1 – 9, muncul teks “Bilangan Satuan”
 - Jika angka positif antara 10 – 99, muncul teks “Bilangan Puluhan”
 - Jika angka di luar rentang 1 – 99, muncul teks “Bukan Bilangan Satuan atau Puluhan”
- Jika angka negatif, muncul teks “Bilangan Negatif”
 - Jika angka negatif merupakan bilangan genap, muncul teks “Bilangan Genap”
 - Jika angka negatif merupakan bilangan ganjil, muncul teks “Bilangan Ganjil”



Latihan Mandiri





Membeli Makanan

- Memasukkan angka sesuai dengan angka menu yang dipilih
- Memasukkan berapa porsi yang ingin dibeli
- Jika pilihan menu sesuai, akan ditampilkan banyak pesanan dan total harga yang harus dibayar, namun jika pilihan tidak sesuai, akan ditampilkan informasi pilihan tidak valid
- Menu makanan:
 1. Nasi Goreng : Rp. 10.000
 2. Mie Goreng : Rp. 11.000
 3. Mie Rebus : Rp. 12.000



Membeli Makanan

Menu Makanan:

1. Nasi Goreng : Rp. 10.000
2. Mie Goreng : Rp. 11.000
3. Mie Rebus : Rp. 12.000

Pilihan : 1

Jumlah : 3

3 Bungkus Nasi Goreng
Total harga: Rp. 30000

Menu Makanan:

1. Nasi Goreng : Rp. 10.000
2. Mie Goreng : Rp. 11.000
3. Mie Rebus : Rp. 12.000

Pilihan : 2

Jumlah : 5

5 Bungkus Mie Goreng
Total harga: Rp. 55000

Menu Makanan:

1. Nasi Goreng : Rp. 10.000
2. Mie Goreng : Rp. 11.000
3. Mie Rebus : Rp. 12.000

Pilihan : 3

Jumlah : 4

4 Bungkus Mie Rebus
Total harga: Rp. 48000

Menu Makanan:

1. Nasi Goreng : Rp. 10.000
2. Mie Goreng : Rp. 11.000
3. Mie Rebus : Rp. 12.000

Pilihan : 0

Jumlah : 2

Pilihan Anda tidak valid. Coba lagi!

Menu Makanan:

1. Nasi Goreng : Rp. 10.000
2. Mie Goreng : Rp. 11.000
3. Mie Rebus : Rp. 12.000

Pilihan : 4

Jumlah : 1

Pilihan Anda tidak valid. Coba lagi!

Percabangan Dengan Satu Baris Perintah

- Blok Program (tanda kurung kurawal buka dan tutup) bisa tidak digunakan apabila ada kondisi percabangan yang **hanya terdiri dari SATU BARIS perintah**

- Contoh:

```
if (n > 0)
    status = "Bilangan Positif";
else if (n < 0)
    status = "Bilangan Negatif";
else
    status = "Angka Nol";
```

Tidak Menggunakan
Blok Program

Percabangan Operator Ternary

- Percabangan Operator Ternary (Operator Kondisi/ *Conditional Operator*) merupakan **bentuk singkat** dari Percabangan IF - ELSE
- Percabangan ini merupakan sebuah operator yang dapat menghasilkan suatu nilai
- Format kode program:

<Kondisi> ? **<Nilai TRUE>** : **<Nilai FALSE>**;

- Contoh:

String hasil = **(5 > 3)** ? **"Benar"** : **"Salah"**;

System.out.println(**(4 == 4)** ? **"Sama"** : **"Beda"**);



Percabangan Operator Ternary

- Membaca nilai yang dimasukkan dalam bentuk angka.
- Jika nilai yang dimasukkan bernilai ganjil, ditampilkan teks “<nilai> adalah bilangan Ganjil”.
- Jika nilai yang dimasukkan bernilai genap, ditampilkan teks “<nilai> adalah bilangan Genap”.

```
Masukkan nilai: 248
248 adalah bilangan Genap
```

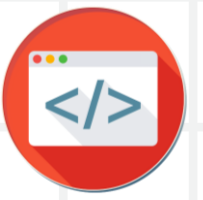
```
Masukkan nilai: 113
113 adalah bilangan Ganjil
```

Percabangan Switch Case

- Percabangan **Switch Case** merupakan bentuk lain dari Percabangan IF – ELSE IF
- **HANYA** dapat diterapkan pada **variabel** yang **nilainya** bersifat **ordinal**
- Bentuk kode programnya adalah sebagai berikut:

```
switch (<variabel>){  
    case <nilai_1> :  
        <Proses 1>;  
        break;  
    case <nilai_2> :  
        <Proses 2>;  
        break;  
    ...  
    case <nilai_#> :  
        <Proses 3>;  
        break;  
    default :  
        <Proses FALSE>  
}
```

- **<nilai_1>** dan seterusnya adalah nilai yang mungkin dari “variabel”
- “**case**” bisa dibuat sebanyak mungkin sesuai kebutuhan
- “**break;**” wajib ditulis untuk tiap “case”
- “**default**” bersifat seperti “ELSE”, bisa digunakan atau tidak digunakan



Percabangan Switch Case

```
switch (pilihan){  
    case 1 :  
        System.out.println("Manis");  
        break;  
    case 2 :  
        System.out.println("Prabroro");  
        break;  
    case 3 :  
        System.out.println("Janggar");  
        break;  
    default :  
        System.out.println("Golput");  
}
```



Percabangan Switch Case

- Memilih ingin makan apa (dengan memasukkan angka pilihannya):
1. Bakso | 2. Mie Ayam | 3. Soto | 4. Rawon | 5. Opor
- Menampilkan harga dari makanan yang dipilih
 - Bakso: Rp. 12.000,-
 - Mie Ayam: Rp. 10.000,-
 - Soto: Rp. 12.000,-
 - Rawon: Rp. 14.000,-
 - Opor: Rp. 14.000,-
- Jika pilihan tidak sesuai, akan ditampilkan teks “Terima Kasih”.

Menu makanan:

1. Bakso
2. Mie Ayam
3. Soto
4. Rawon
5. Opor

Masukkan pilihan Anda (Angka): 3
Rp. 12.000,-

TERIMA KASIH

