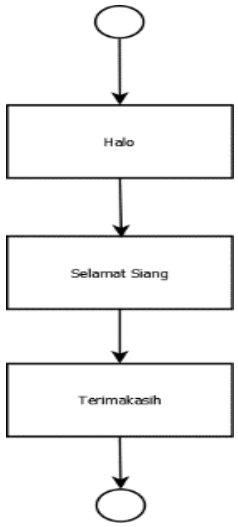
FUNGSI LOGIKA ATAU PERCABANGAN ANDROID STUDIO

Oleh:
Taryana Suryana M.Kom
Teknik Informatika Unikom

taryanarx@email.unikom.ac.id
taryanarx@gmail.com
Line/Telegram: 081221480577

PERCABANGAN

Percabangan merupakan sebuah cara yang digunakan dalam program untuk mengambil suatu keputusan. Didalam pemrograman kita harus dapat menentukan aksi apa yang harus dikerjakan oleh pemroses (processor) ketika sebuah kondisi terpenuhi, dengan menggunakan operasi logic. Percabangan juga dikenal dengan istilah "Control Flow", "Struktur Kondisi", "Struktur IF", "Decision", dsb. Semuanya itu pada dasarnya adalah sama



Gambar 1: Alur Program Sederhana

Cara menulis kode percabangan:
Caranya: menggunakan kata kunci if, else, switch, dan case, dan operator ternary.
Contoh format stuktur IF seperti ini:

```
if(suatu kondisi) {
    // lakukan sesuatu kalau kondisi benar
    // Lakukan ini juga
}
```

suatu_kondisi hanya bernilai true/false saja. Kita bisa gunakan operator relasi dan logika di sini.

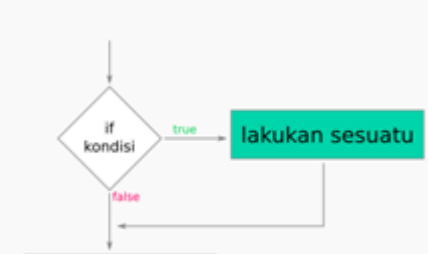
Sebelumnya, kamu perlu tahu dulu tiga bentuk percabangan pada Java:

1. Percabangan IF
2. Percabangan IF..ELSE

3. Percabangan IF..ELSE..IF atau SWITCH..CASE

Percabangan IF

Percabangan ini hanya memiliki satu pilihan. Artinya, pilihan di dalam IF hanya akan dikerjakan kalau kondisinya benar.



Gambar 2. Kondisi IF

Tapi kalau salah… tidak akan melakukan apa-apa. Alias lanjut eksekusi ke perintah berikutnya.

Contoh: Misalnya jika nilai anda lebih dari 70 maka anda lulus

```
nilai=70;
if(nilai>=70)
{
System.out.println("Anda Lulus");
}
```

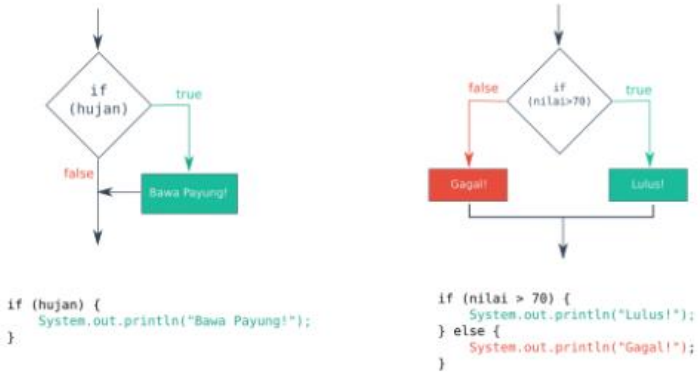
Percabangan if..else

Pernyataan Percabangan dengan mengunakan If-else digunakan untuk mengambil suatu keputusan diantara banyak pernyataan yang ada.

Percabangan IF..ELSE memiliki pilihan alternatif kalau kondisinya salah.

IF: “Jika kondisi benar maka kerjakan ini, kalau tidak silahkan lanjut”

IF..ELSE: “Jika kondisi benar maka kerjakan ini, kalau salah maka kerjakan yang itu, setelah itu lanjut”



Gambar 3. Kondisi if..else

```
if (kondisi)
{
   //Jika kondisi terpenuhi maka
   Pernyataan1 dieksekusi
}
else
{
   Pernyataan2 dieksekusi
}
```

Contoh Program Menggunakan if..else

```
var nilai=80;
 if(nilai>70)
 {
   System.out.println("Anda Lulus");
 }
 else
 {
      System.out.println("Anda Gagal");
 }
```

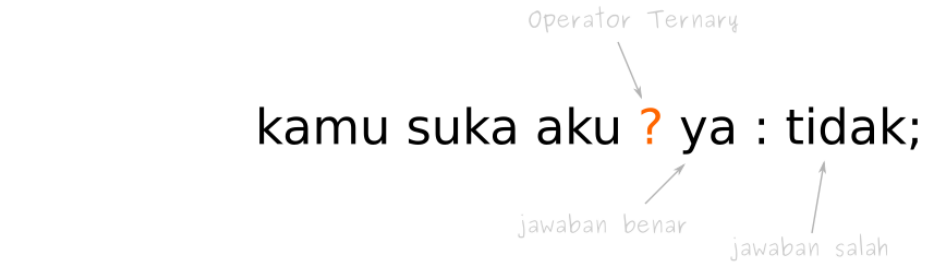
Contoh2:
Misalnya: dalam matakuliah pemrograman , untuk menghitung nilai akhir (NA) digunakan rumus berikut:
NA=40% TUGAS + 30% UTS + 30%UAS,
dan dinyatakan **Lulus** Jika Nilai Akhir lebih besar dari 70

Maka penyelesaiannya adalah sebagai berikut:

```
public class Nilai {
    public static void main(String[] args) {
        // Deklarasi variabel
        int tugas=75;
         int uts=80;
         int uas=70;
         double na=0.4*tugas + 0.3*uts + 0.3*uas;
        System.out.println("Nilai Akhir : " +na);
        // cek apakah na diatas 70
        if (na>70 ) {
          System.out.println("Selamat, anda Lulus");
        }else {
        System.out.println("Maaf Anda Gagal ...");
        }
    }
}
```

Percabangan IF..ELSE dengan Operator Ternary

Selain menggunakan struktur seperti di atas, percabangan ini juga dapat menggunakan operator ternary.
Operator ternary memiliki konsep yang sama seperti percabganan IF..ELSE.



Gambar 4. Operator Ternary

Contoh programnya:

```
public class OperatorTernary {
    public static void main(String[] args) {
        boolean suka = true;
        String jawaban;
        // menggunakan operator ternary
        jawaban = suka ? "iya" : "tidak";
        // menampilkan jawaban
        System.out.println(jawaban);
    }
}
```

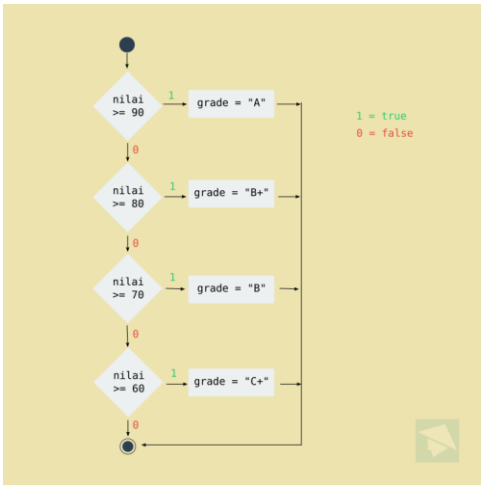
Percabangan IF..ELSE..IF dan SWITCH/CASE

Jika percabangan IF..ELSE hanya memiliki dua pilihan saja. Maka percabangan IF..ELSE..IF memiliki lebih dari dua pilihan.

Formatnya seperti ini:

```
if (suatu kondisi) {
    // maka kerjakan ini
} else if (kondisi lain) {
    // kerjakan ini
} else if (kondisi yang lain lagi) {
    // kerjakan perintah ini
} esle {
    // kerjakan ini kalua tidak ada yg memenuhi diatas
}
```

Coba perhatikan contohnya:



Gambar 5. Kondisi if..else..if

Jika nilainya lebih besar dari 90, maka grade-nya “A”. Sedangkan kalau lebih besar dari 80, maka “B+”. Lebih besar dari 70, maka “B”, dan seterusnya.
Lebih jelasnya, mari kita buat program.

Program HitungGrade

Silahkan buat sebuah class baru bernama HitungGrade, kemudian ikuti kode program berikut.

```
import java.util.Scanner;

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

        // membuat variabel dan scanner
        int nilai;
        String grade;
        Scanner scan = new Scanner(System.in);

        // mengambil input
        System.out.print("Inputkan nilai: ");
        nilai = scan.nextInt();

        // higung gradenya
        if ( nilai >= 90 ) {
            grade = "A";
        } else if ( nilai >= 80 ){
            grade = "B+";
        } else if ( nilai >= 70 ){
            grade = "B";
        } else if ( nilai >= 60 ){
            grade = "C+";
        } else if ( nilai >= 50 ){
            grade = "C";
        } else if ( nilai >= 40 ){
            grade = "D";
        } else {
            grade = "E";
        }

        // cetak hasilnya
        System.out.println("Grade: " + grade);
    }
}
```

Percabangan SWITCH..CASE

Percabangan SWITCH..CASE sebenarnya adalah bentuk lain dari IF..ELSE..IF.
Bedanya, percabangan ini menggunakan kata kunci switch dan case.
Formatnya juga berbeda, tapi cara kerjanya sama.

```
switch(variabel){
   case 1:
      // kerjakan kode ini
      // kode ini juga
      break;
   case 2:
      // kerjakan kode ini
```

```
      // kode ini juga
      break;
   case 3:
      // kerjakan kode ini
      // kode ini juga
      break;
   default:
      // kerjakan kode ini
      // kode ini juga
      break;
}
```

Perhatikan: case 1 artinya nilai variabel yang akan dibandingkan, apakah nilainya sama dengan 1 atau tidak.
Kalau iya, maka kerjakan kode yang ada di dalam case 1.
Bisa juga betuknya berbeda, misalnya seperti ini:

```
switch (variabel) {
    case 'A':
        // lakukan sesuatu
        break;
    case 'B':
        // lakukan ini
        break;
    default:
        // lakukan ini
}
```

Perlu diperhatikan juga: di sana ada kata kunci break dan default.

- break artinya berhenti. Ini untuk memerintahkan komputer untuk berhenti mengecek case yang lainnya.
- default artinya jika nilai variabel tidak ada yang sama dengan pilihan case di atas, maka kerjakan kode yang ada di dalam default.

Pilihan default bisa juga tidak memiliki break, karena dia adalah pilihan terakhir. Artinya pengecekan akan berakhir di situ.

Contoh program dengan percabangan SWITCH..CASE

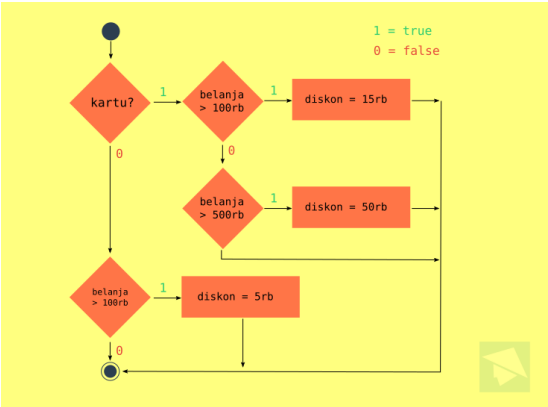
```
import java.util.Scanner;

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

        // membuat variabel dan Scanner
        String lampu;
        Scanner scan = new Scanner(System.in);

        // mengambil input
        System.out.print("Inputkan nama warna: ");
        lampu = scan.nextLine();

        switch(lampu){
            case "merah":
                System.out.println("Lampu merah, berhenti!");
                break;
            case "kuning":
                System.out.println("Lampu kuning, harap hati-hati!");
                break;
            case "hijau":
                System.out.println("Lampu hijau, silahkan jalan!");
                break;
            default:
                System.out.println("Warna lampu salah!");
        }
    }
}
```

Percabangan dalam Percabangan (Nested Loop)

Kita sudah tahu tiga bentuk dasar percabangan di Java. Selanjutnya, kita coba bahas percabangan yang ada di dalam perbangan (percabangan bersarang).
Jadi, percabangan itu bisa dibuat di dalam percabangan. Kadang teknik ini disebut juga *nested if*.
Contoh kasus:
Misalnya ada model bisinis seperti ini di sebuah toko. Ketika orang membayar di kasir, biasanya ditanya ada kartu member untuk mendapatkan diskon dan sebagainya.

```
Apakah anda punya kartu member?
   - ya
       * Apakah belanjaan anda lebih dari 500rb?
           # ya : mendapatkan diskon 50rb
           # tidak : tidak mendapatkan diskon
       * Apakah belanjaan anda lebih dari 100rb?
           # ya : mendapatkan diskon 15rb
           # tidak: tidak mendapatkan diskon
   - tidak
       * Apakah belanjaan anda lebih dari 100rb?
           # ya : mendapatkan diskon 10rb
           # tidak: tidak mendapatkan diskon
```

Perhatikan flow chart-nya:



Gambar 6. If bersarang

Perhatikan Contoh Program Berikut:
Silahkan buat class baru bernama Kasir dan ikuti kode program berikut ini.

```
import java.util.Scanner;
public class Kasir {
    public static void main(String[] args) {
        // deklarasi variabel dan Scanner
        int belanjaan, diskon, bayar;
        String kartu;
        Scanner scan = new Scanner(System.in);

        // mengambil input
        System.out.print("Apakah ada kartu member: ");
        kartu = scan.nextLine();
        System.out.print("Total belanjaan: ");
        belanjaan = scan.nextInt();

        // proses
        if (kartu.equalsIgnoreCase("ya")) {
            if (belanjaan > 500000) {
                diskon = 50000;
            } else if (belanjaan > 100000) {
                diskon = 15000;
            } else {
                diskon = 0;
            }

        } else {
            if (belanjaan > 100000) {
                diskon = 5000;
            } else {
                diskon = 0;
            }
        }

        // total yang harus dibayar
        bayar = belanjaan - diskon;

        // output
        System.out.println("Total Bayar: Rp " + bayar);
    }
}
```

Cobalah untuk mengubah nilai yang dimasukkan dan perhatikan hasilnya.
Mungkin di sana ada yang perlu diperhatikan:

- Fungsi equalsIgnoreCase("ya") digunakan untuk membandingkan String dengan tidak memperdulikan huruf besar dan kecilnya.
- Ada juga Fungsi equals(), fungsinya sama. Tapi equals() akan memperhatikan *case* hurufnya.

Kenapa tidak menggunakan operator == atau !=?
Di Java memang seperti itu.
Kalau kita ingin membandingkan nilai String, ya… menggunakan fungsi yang dua tadi.
Tapi, kalau membandingkan selain String, maka bisa pakai operator == atau !=.

Menggunakan Operator Logika dalam Percabangan

Operator logika dalam percabangan sebenarnya bisa membuat percabangan menjadi lebih singkat.
Misal ada program *Tilang* dengan logika seperti ini:

```
public class Tilang {
    public static void main(String[] args) {
        boolean SIM = false;
        boolean STNK = true;

        // cek apakah dia akan ditilang atau tidak
        if(SIM == true){
            if( STNK == true ) {
                System.out.println("Tidak ditilang!");
            }
        } else {
            System.out.println("Anda ditilang!");
        }
    }
}
```

Perhatikan: di sana kita menggunakan percabangan bersarang untuk mengecek, apakah dia ditilang atau tidak.
Hal ini sebenarnya bisa disingkat dengan operator logika, sehingga menjadi seperti ini:

```
public class Tilang {
    public static void main(String[] args) {
        boolean SIM = false;
        boolean STNK = true;

        // cek apakah dia akan ditilang atau tidak
        if(SIM == true && STNK == true){
            System.out.println("Tidak ditilang!");
        } else {
            System.out.println("Anda ditilang!");
        }
    }
}
```

Pada kode di atas, kita menggunakan operator AND (&&).
Karena logikanya: Si pengendara tidak akan ditilang kalau punya SIM dan STNK.

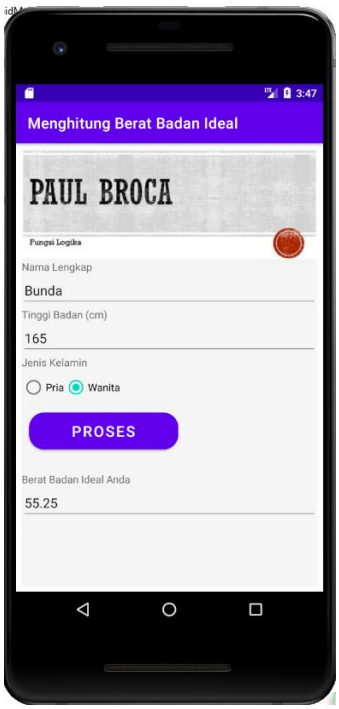
LATIHAN

Dengan Menggunakan Android Buat Program Untuk Menghitung Berat Badan Ideal:

1. Rumus yang ditemukan oleh Paul Broca ini membedakan cara penghitungan antara pria dan wanita.
   Hal ini disebabkan karena pria dan wanita memiliki komposisi tubuh yang berbeda.

   Berat Badan Ideal Menurut Broca
   Pria= tb-100 * 90%
   Wanita = tb-100 * 85

   Contoh Hasil:

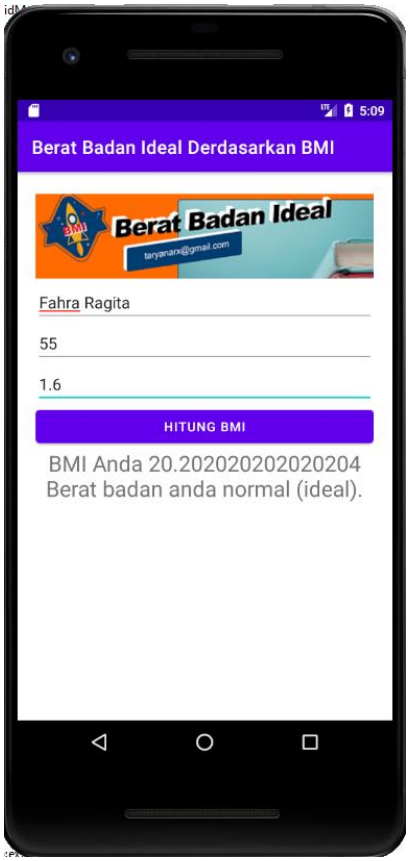


2. *Body Mass Index* (BMI) adalah cara menghitung berat badan ideal berdasarkan tinggi dan berat badan, BMI dibedakan berdasarkan usia.
   a. Angka BMI normal berada pada kisaran 18,5-25.
   b. Jika angka BMI melebihi 25, kamu memiliki berat badan berlebih.
   c. Sedangkan, jika angka BMI berada di bawah 18 berarti berat badanmu kurang.
   d. Jika angka BMI sudah melebihi angka 40, sebaiknya dilakukan penanganan secepatnya karena angka ini menunjukkan tanda bahaya.

Cara menghitungnya:

Berat badan ideal = Berat badan (kilogram): Tinggi badan (meter)
Sebagai contohnya, jika berat badan kamu 47 kilogram dan tinggi badan 1,63 meter, nilai BMI kamu adalah $47:(1,63)^2= 17.8$.
Berarti, berat badan kamu termasuk di bawah rata-rata.

Contoh Output



setelah semua program selesai dibuat dan hasilnya sesuai dengan hasil hitungan manual anda :

Dengan menggunakan Microsoft Word Copy Paste Kode program beserta tangkapan layar hasil keluarannya (screenshot), kemudian kirim ke Modul Tugas yang ada di kuliahonline.unikom.ac.id, **Nama Filenya: Tugasx-nama-nim-kelas.pdf**
Dikumpulkan Paling Lambat Setiap Hari Minggu Jam 18:00

Ok…selamat mencoba

Reff:https://petanikode.com