

JOBSHEET 4

Sintaks Pemilihan 1

1. Tujuan

- Mahasiswa memahami tentang operator logika
- Mahasiswa mampu menyelesaikan permasalahan dengan menggunakan sintaks pemilihan
- Mahasiswa mampu membuat sebuah program Java yang memanfaatkan sintaks pemilihan

2. Teori

2.1 Operator

Operator adalah simbol-simbol khusus yang digunakan untuk mengoperasikan suatu nilai data (operand). Operator terdiri dari Assignment Operator, Arithmetic Operator, Relational Operator, dan Logical Operator.

a. Assignment Operator

Operasi yang dijalankan oleh sebuah program merupakan suatu proses yang melibatkan operator (pelaku) dan operand (nilai) untuk melakukan sesuatu. Assignment operator menggunakan tanda '=' sebagai operator utama yang artinya mengisikan/menyimpan suatu nilai ke dalam variabel yang terdapat pada sisi kiri tanda '='.

Contoh: **angka = 11**, artinya nilai 11 akan disimpan ke dalam variabel bernama 'angka'

Jenis-jenis assignment operator dapat dilihat pada Tabel 1 berikut.

Tabel 1. Assignment Operator

Operator	Deskripsi	Contoh
=	Menetapkan nilai dari operand kanan ke operand kiri	C = A sama dengan memasukkan nilai A ke dalam C
+=	Menambah nilai operand kiri dengan kanan dan menetapkan nilainya ke operand sebelah kiri	C += A sama dengan C = C + A
-=	Mengurangi nilai operand kiri dengan kanan dan menetapkan nilainya ke operand sebelah kiri	C -= A sama dengan C = C - A
*=	Mengalikan nilai operand kiri dengan kanan dan menetapkan nilainya ke operand sebelah kiri	C *= A sama dengan C = C * A

Operator	Deskripsi	Contoh
/=	Membagi nilai operand kiri dengan kanan dan menetapkan nilainya ke operand sebelah kiri	$C /= A$ sama dengan $C = C / A$
%=	Menghitung sisa bagi (modulus) operand kiri dengan operand kanan, dan menetapkan nilainya pada operand di sebelah kiri	$C \% = A$ sama dengan $C = C \% A$
<<=	Digit biner operand sebelah kiri bergeser sejumlah operand sebelah kanan ke arah kiri dan menetapkan nilainya pada operand sebelah kiri	$C <<= 2$ sama dengan $C = C << 2$
>>=	Digit biner operand sebelah kiri bergeser sejumlah operand sebelah kanan ke arah kanan dan menetapkan nilainya pada operand sebelah kiri	$C >>= 2$ sama dengan $C = C >> 2$
&=	operand kiri memperoleh nilai dari operand kiri & operand kanan	$C \&= 2$ sama dengan $C = C \& 2$
^=	operand kiri memperoleh nilai dari operand kiri ^ operand kanan	$C \wedge= 2$ sama dengan $C = C \wedge 2$
=	operand kiri memperoleh nilai dari operand kiri operand kanan	$C = 2$ sama dengan $C = C 2$

b. Arithmetic Operator (Binary dan Unary)

Arithmetic operator digunakan untuk melakukan operasi perhitungan matematika.

Terdapat lima operasi dasar aritmatika, yaitu:

- + : tambah
- - : kurang
- * : kali
- / : bagi
- % : sisa hasil bagi (modulo)

Operator ini terdiri dari operator Binary dan Unary. Operator Binary melibatkan dua operand pada setiap operasinya, sedangkan operator Unary hanya melibatkan satu operand pada setiap operasinya. Jenis-jenis arithmetic operator binary dapat dilihat pada Tabel 2 berikut.

Tabel 2. Arithmetic Operator Binary

Operator	Deskripsi
+	Menjumlahkan dua operand
-	Mengurangkan dua operand
*	Mengalikan dua operand
/	Membagikan dua operand
%	Menghitung sisa bagi operand pertama dengan kedua

Kemudian untuk jenis-jenis arithmetic operator unary dapat dilihat pada Tabel 3 berikut.

Tabel 3. Arithmetic Operator Unary

Operator	Deskripsi
++	Meningkatkan nilai integer satu persatu (increment)
--	Menurunkan nilai integer satu persatu (decrement)

c. Relational Operator

Relational operator (operator perbandingan) akan membandingkan dua buah operand, baik berupa nilai atau suatu variabel. Perbandingan dua buah operand ini memiliki suatu tujuan yaitu untuk memenuhi sebuah kondisi sehingga dapat dilakukan tindakan lanjutan berupa proses atau kondisi lain. Jenis-jenis relational operator dapat dilihat pada Tabel 4 berikut.

Tabel 4. Relational Operator

Operator	Deskripsi
==	Melakukan pengecekan kesamaan dua operand, kondisi menjadi true apabila kondisi tersebut terpenuhi
!=	Melakukan pengecekan ketidak samaan dua operand, kondisi menjadi true apabila kondisi tersebut terpenuhi
>	Melakukan pengecekan kondisi operand kiri lebih besar dari pada operand kanan, apabila terpenuhi maka kondisi menjadi true
<	Melakukan pengecekan kondisi operand kiri lebih kecil dari pada operand kanan, apabila terpenuhi maka kondisi menjadi true
>=	Melakukan pengecekan kondisi operand kiri lebih besar sama dengan dari pada operand kanan, apabila terpenuhi maka kondisi menjadi true
<=	Melakukan pengecekan kondisi operand kiri lebih kecil sama dengan dari pada operand kanan, apabila terpenuhi maka kondisi menjadi true

d. Logical Operator

Logical operator (operator logika) merupakan jenis operator yang akan membandingkan logika hasil dari Relational operator. Terdapat 3 macam operator yang termasuk dalam operator logika yaitu:

- && : AND
- || : OR
- == : HAS A VALUE OF
- ! : NOT

Jenis-jenis logical operator dapat dilihat pada Tabel 5 berikut.

Tabel 5. Logical Operator

Operator	Deskripsi
&&	Melakukan pengecekan kondisi yang harus bernilai true untuk kedua operand kanan dan kiri secara bersamaan
	Melakukan pengecekan kondisi yang dapat bernilai true pada salah satu atau kedua operand kanan dan kiri
==	Melakukan pengecekan kondisi yang dapat bernilai true jika nilai operand kanan sama dengan operand kiri, serta false jika nilai kedua operand tersebut tidak sama
!	Melakukan pengecekan kondisi NOT, atau membalikkan kondisi. Contoh !(A && B)

2.2 Sintaks Pemilihan Sederhana

Dalam kehidupan sehari-hari, kita sering kali menemukan sebuah keadaan yang mengharuskan kita untuk mengambil sebuah keputusan dengan mempertimbangkan berbagai hal. Komputer dapat diprogram untuk membuat sebuah keputusan, yaitu melakukan hal-hal yang berbeda pada kondisi-kondisi yang berbeda.

Sintaks pemilihan merupakan sebuah statement yang digunakan untuk mengatur tindakan yang harus dilakukan oleh komputer pada kondisi-kondisi tertentu. Dengan adanya statement ini, kita bisa mengatur kapan sebuah perintah akan dijalankan, yaitu setelah suatu syarat tertentu sudah terpenuhi. Contohnya:

Jika **suhu kurang dari 16°C** maka **memakai jaket**

Pernyataan tersebut mengandung sebab akibat. Keterangan “suhu kurang dari 16°C” adalah sebuah syarat yang harus terpenuhi, sedangkan “memakai jaket” adalah tindakan yang akan dilakukan apabila syaratnya terpenuhi.

Dalam dunia logika, istilah **syarat terpenuhi** dapat diartikan **syarat tersebut bernilai benar atau TRUE**. Selanjutnya pernyataan “Jika ... maka ...” dapat dipakai pada pemrograman dengan menggunakan sebuah statement sintaks pemilihan. Sintaks pemilihan terdiri dari if, if-else, dan switch-case.

a. Sintaks Pemilihan IF

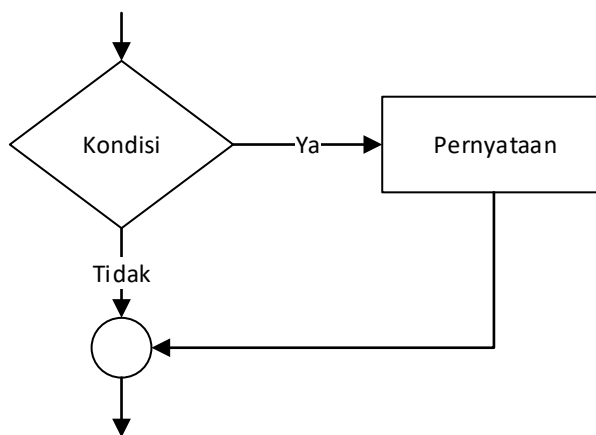
Bentuk umum sintaks pemilihan IF adalah sebagai berikut:

```
if (Kondisi)
{
    Pernyataan;
}
```

Kondisi merupakan ekspresi Boolean, yaitu menghasilkan nilai TRUE atau FALSE

- Jika nilai **Kondisi** bernilai benar (TRUE), maka **Pernyataan** akan dijalankan
- Jika nilai **Kondisi** bernilai salah (FALSE), maka **Pernyataan** tidak akan dijalankan

Flowchart:



Contoh program:

```

1  import java.util.Scanner;
2
3  public class ifkondisi{
4      public static void main(String[] args){
5          Scanner sc = new Scanner(System.in);
6          int suhu;
7
8          System.out.print("Masukkan suhu saat ini: ");
9          suhu = sc.nextInt();
10
11         if(suhu < 16){
12             System.out.println("Silahkan pakai jaket");
13         }
14     }
15 }

```

Hasil:

```

D:\>javac ifkondisi.java
D:\>java ifkondisi
Masukkan suhu saat ini: 11
Silahkan pakai jaket

```

b. Sintaks Pemilihan IF-ELSE

Bentuk umum sintaks pemilihan IF-ELSE adalah sebagai berikut:

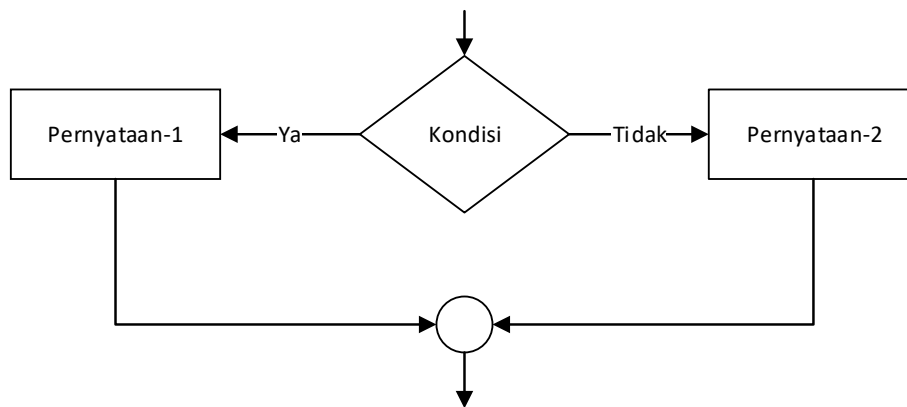
```

if (Kondisi)
{
    Pernyataan-1;
}
else
{
    Pernyataan-2;
}

```

Struktur pemilihan IF-ELSE minimal harus mempunyai 2 pernyataan. Jika **Kondisi** bernilai TRUE atau terpenuhi, maka **Pernyataan-1** akan dijalankan. Namun, jika **Kondisi** bernilai FALSE, maka **Pernyataan-2** yang akan dijalankan.

Flowchart:



Contoh program:

```

1  import java.util.Scanner;
2
3  public class ifelsekondisi{
4      public static void main(String[] args){
5          Scanner sc = new Scanner(System.in);
6          int suhu;
7
8          System.out.print("Masukkan suhu saat ini: ");
9          suhu = sc.nextInt();
10
11         if(suhu < 16){
12             System.out.println("Silahkan pakai jaket");
13         }
14         else{
15             System.out.println("Silahkan pakai topi");
16         }
17     }
18 }
  
```

Hasil:

```

D:\>javac ifelsekondisi.java

D:\>java ifelsekondisi
Masukkan suhu saat ini: 20
Silahkan pakai topi
  
```

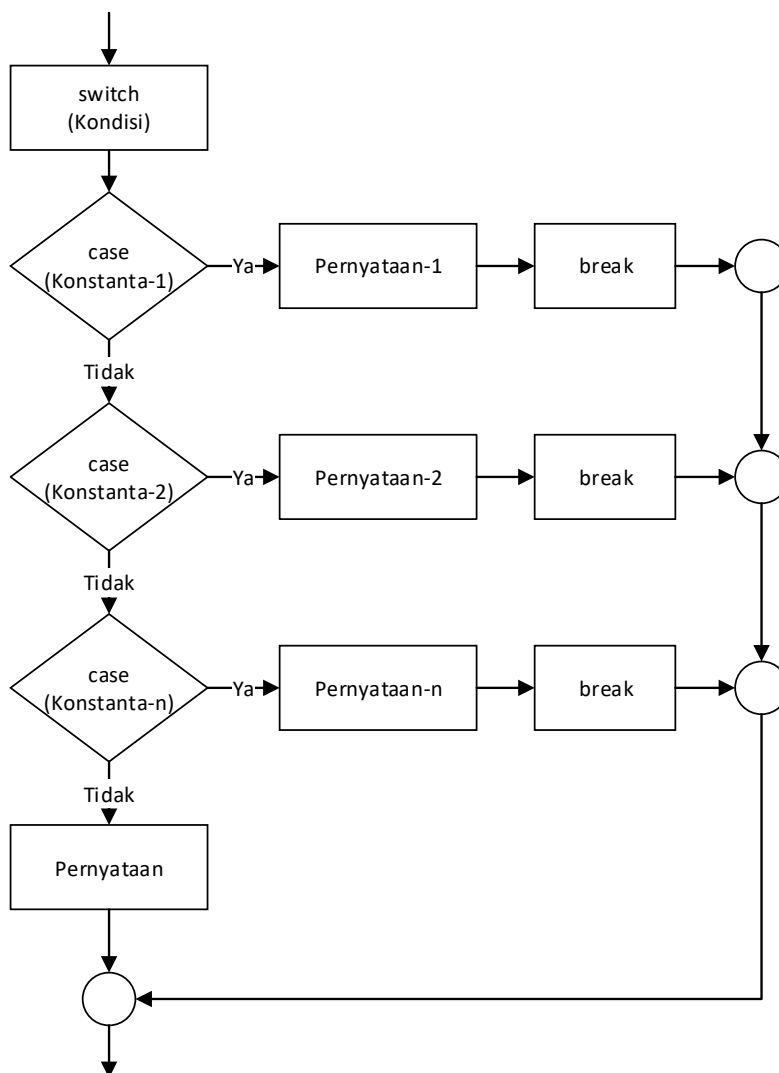
c. Sintaks Pemilihan SWITCH-CASE

Sintaks pemilihan ini digunakan ketika kita mempunyai banyak kemungkinan tindakan yang harus dilakukan pada kondisi yang berbeda-beda. Sintaks pemilihan ini akan menjalankan salah satu dari beberapa pernyataan “case” sesuai dengan nilai kondisi yang ada di dalam “switch”. Selanjutnya proses akan dilanjutkan sampai ditemukan pernyataan “break”. Namun, jika tidak ada nilai pada case yang sesuai dengan nilai kondisi, maka proses akan dilanjutkan ke pernyataan yang ada di dalam “default”.

```

switch (Kondisi)
{
    case Konstanta-1:
        Pernyataan-1;
        break;
    case Konstanta-2:
        Pernyataan-2;
        break;
    ...
    ...
    case Konstanta-n:
        Pernyataan-n;
        break;
    default:
        Pernyataan;
}
    
```

Flowchart:



Contoh program:

```
1  import java.util.Scanner;
2
3  public class switchcasekondisi{
4      public static void main(String[] args){
5          Scanner sc = new Scanner(System.in);
6          int angka;
7
8          System.out.print("Masukkan angka: ");
9          angka = sc.nextInt();
10
11         switch(angka){
12             case 1:
13                 System.out.println("Hari Senin");
14                 break;
15             case 2:
16                 System.out.println("Hari Selasa");
17                 break;
18             case 3:
19                 System.out.println("Hari Rabu");
20                 break;
21             case 4:
22                 System.out.println("Hari Kamis");
23                 break;
24             case 5:
25                 System.out.println("Hari Jumat");
26                 break;
27             case 6:
28                 System.out.println("Hari Sabtu");
29                 break;
30             case 7:
31                 System.out.println("Hari Minggu");
32                 break;
33
34             default:
35                 System.out.println("Maaf, angka yang Anda masukkan salah");
36         }
37     }
38 }
```

Hasil:

```
D:\>javac switchcasekondisi.java
D:\>java switchcasekondisi
Masukkan angka: 5
Hari Jumat
```

2.3 Sintaks Pemilihan Multi Kondisi

Untuk melakukan sebuah tindakan, terkadang kita tidak hanya diwajibkan untuk memenuhi sebuah syarat saja, tetapi ada beberapa syarat yang harus terpenuhi. Contoh kasusnya adalah sebagai berikut:

Seorang dosen sedang menghitung nilai akhir mahasiswa yang terdiri dari nilai tugas, nilai kuis, nilai UTS, dan nilai UAS. Untuk menentukan kelulusan mahasiswa, nilai akhir mahasiswa

harus di atas 60 agar bisa dinyatakan lulus. Selain itu, nilai UAS juga harus berada di atas 75 agar mahasiswa tersebut dapat dinyatakan lulus. Dengan demikian jika terdapat mahasiswa yang mempunyai nilai akhir di atas 60, namun nilai UAS-nya di bawah 75, maka mahasiswa tersebut tetap dinyatakan tidak lulus.

Berdasarkan kasus tersebut, dapat kita ketahui bahwa terdapat dua kondisi yang harus terpenuhi agar sebuah tindakan dapat dilakukan, yaitu:

- Kondisi 1: nilai akhir harus lebih dari 60
- Kondisi 2: nilai UAS harus lebih dari 75

Agar komputer dapat menentukan tindakan yang harus dilakukan jika terdapat multi kondisi, kita perlu menambahkan operator logika pada sintaks pemilihan. Operator logika yang dapat digunakan adalah sebagai berikut:

- `&&` : semua syarat pada kondisi 1 dan kondisi 2 harus terpenuhi agar bernilai TRUE. Jika hanya salah satu yang terpenuhi maka nilainya FALSE
- `||` : minimal salah satu syarat pada kondisi 1 atau kondisi 2 harus terpenuhi agar bernilai TRUE. Jika kedua kondisi terpenuhi, maka nilainya juga TRUE

Contoh program:

```
1  import java.util.Scanner;
2
3  public class Kelulusan{
4      public static void main(String[] args){
5          Scanner sc = new Scanner(System.in);
6          double tugas, kuis, uts, uas, nilaiAkhir;
7
8          System.out.print("Masukkan nilai tugas: ");
9          tugas = sc.nextDouble();
10         System.out.print("Masukkan nilai kuis: ");
11         kuis = sc.nextDouble();
12         System.out.print("Masukkan nilai uts: ");
13         uts = sc.nextDouble();
14         System.out.print("Masukkan nilai uas: ");
15         uas = sc.nextDouble();
16
17         nilaiAkhir = (tugas + kuis + uts + uas) / 4;
18
19         if((nilaiAkhir > 60) && (uas > 75)){
20             System.out.println(nilaiAkhir);
21             System.out.println("Selamat Anda lulus!");
22         }
23         else{
24             System.out.println(nilaiAkhir);
25             System.out.println("Maaf, Anda tidak lulus");
26         }
27     }
28 }
```

Hasil:

```
D:\>java Kelulusan
Masukkan nilai tugas: 58
Masukkan nilai kuis: 75
Masukkan nilai uts: 81
Masukkan nilai uas: 56
67.5
Maaf, Anda tidak lulus
```

3. Praktikum

3.1 Percobaan 1

1. Buka text Editor (Sublime, Notepad++, atau Atom)
2. Buat file baru dengan nama Bilangan.java
3. Buat class dengan nama Bilangan
4. Tuliskan struktur dasar bahasa Java yang berisi fungsi main()
5. Tambahkan import library Scanner di bagian paling atas kode program

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

6. Deklarasikan variabel sc dengan tipe Scanner

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

7. Buatlah variabel bertipe int, kemudian beri nama variabel tersebut dengan angka

```
int angka;
```

8. Tambahkan kode berikut ini untuk menerima masukan dari keyboard

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

9. Buat struktur kondisi untuk mengecek angka yang dimasukkan, apakah termasuk bilangan genap atau ganjil.

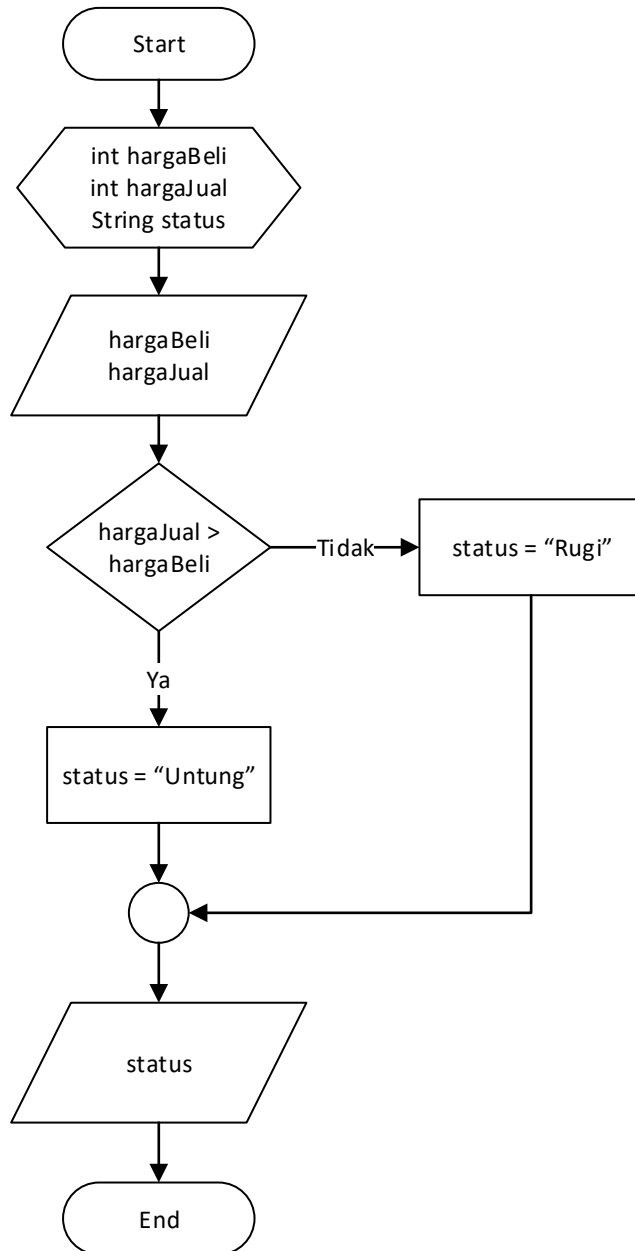
```
if(angka % 2 == 0){
    System.out.println("Angka " + angka + " adalah bilangan genap");
}
else{
    System.out.println("Angka " + angka + " adalah bilangan ganjil");
}
```

10. Lakukan kompilasi dengan menggunakan perintah javac, kemudian jalankan program.

Amati apa yang terjadi!

3.2 Percobaan 2

1. Buka text Editor (Sublime, Notepad++, atau Atom)
2. Perhatikan flowchart berikut ini



3. Buat file baru dengan nama Penjualan.java
4. Buat class dengan nama Penjualan
5. Tuliskan struktur dasar bahasa Java yang berisi fungsi main()
6. Tambahkan import library Scanner di bagian paling atas kode program
7. Deklarasikan variabel sc dengan tipe Scanner

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

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



8. Buatlah dua variabel bertipe int dan satu variabel bertipe String, kemudian beri nama tiga variabel tersebut dengan hargaBeli, hargaJual, dan status

```
int hargaBeli;  
int hargaJual;  
String status;
```

9. Tambahkan kode berikut ini untuk menerima masukan dari keyboard

```
System.out.print("Masukkan harga beli barang: ");  
hargaBeli = sc.nextInt();  
System.out.print("Masukkan harga jual barang: ");  
hargaJual = sc.nextInt();
```

10. Buat struktur kondisi untuk mengecek untung dan rugi dengan syarat jika hargaJual lebih besar dari hargaBeli, maka dinyatakan untung, selain dari itu dinyatakan rugi

```
if(hargaJual > hargaBeli){  
    status = "Untung";  
}  
else{  
    status = "Rugi";  
}
```

11. Tambahkan kode berikut untuk menampilkan keluaran status penjualan

```
System.out.println("Anda " + status);
```

12. Lakukan kompilasi dengan menggunakan perintah javac, kemudian jalankan program. Amati apa yang terjadi!

3.3 Percobaan 3

1. Buka text Editor (Sublime, Notepad++, atau Atom)
2. Buat file baru dengan nama Kalkulator.java
3. Buat class dengan nama Kalkulator
4. Tuliskan struktur dasar bahasa Java yang berisi fungsi main()
5. Tambahkan import library Scanner di bagian paling atas kode program

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

6. Deklarasikan variabel sc dengan tipe Scanner

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

7. Buatlah tiga variabel bertipe double dan satu variabel bertipe char, kemudian beri nama empat variabel tersebut dengan angka1, angka2, hasil, dan operator

```
double angka1, angka2, hasil;  
char operator;
```



8. Tambahkan kode berikut ini untuk menerima masukan dari keyboard

```
System.out.print("Masukkan angka pertama: ");
angka1 = sc.nextDouble();
System.out.print("Masukkan angka kedua: ");
angka2 = sc.nextDouble();
System.out.print("Masukkan operator (+ - * /): ");
operator = sc.next().charAt(0);
```

9. Buat struktur kondisi untuk melakukan operasi perhitungan dua angka yang dimasukkan sesuai dengan operator yang dipilih

```
switch(operator) {
    case '+':
        hasil = angka1 + angka2;
        System.out.println(angka1 + " + " + angka2 + " = " + hasil);
        break;
    case '-':
        hasil = angka1 - angka2;
        System.out.println(angka1 + " - " + angka2 + " = " + hasil);
        break;
    case '*':
        hasil = angka1 * angka2;
        System.out.println(angka1 + " * " + angka2 + " = " + hasil);
        break;
    case '/':
        hasil = angka1 / angka2;
        System.out.println(angka1 + " / " + angka2 + " = " + hasil);
        break;
    default:
        System.out.println("Operator yang Anda masukkan salah");
}
```

10. Lakukan kompilasi dengan menggunakan perintah javac, kemudian jalankan program. Amati apa yang terjadi!

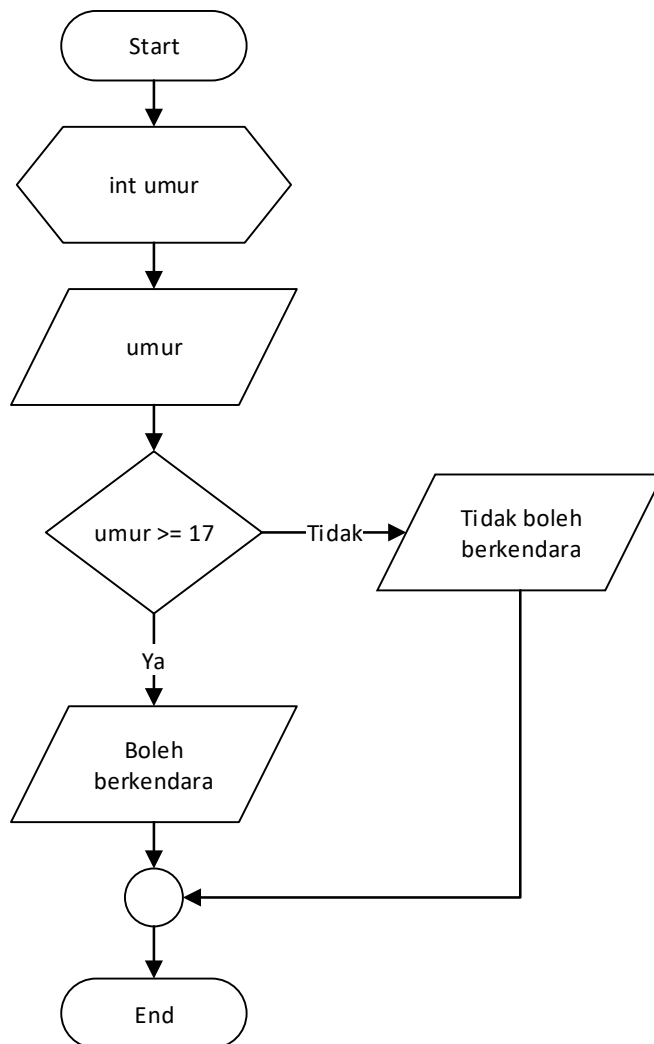
Pertanyaan

1. Apakah yang dimaksud dengan operator?
2. Jelaskan fungsi dari:
 - else
 - break
 - default
3. Jelaskan fungsi dari perintah berikut:
 - operator = sc.next().charAt(0);
 - if(angka % 2 == 0)

4. Pada Percobaan 3, tambahkan kode program untuk kondisi menghitung hasil bagi kedua angka masukan, sehingga keluaran program yang ditampilkan adalah sisa hasil pembagian antara angka pertama dengan angka kedua!

Tugas

1. Buatlah sebuah program untuk memasukkan dua buah bilangan bulat, kemudian salah satu bilangan bulat yang mempunyai nilai terbesar akan dicetak di layar!
2. Perhatikan flowchart berikut ini.



Buatlah kode program sesuai dengan flowchart tersebut!

3. Setiap akhir bulan, pegawai keuangan menghitung gaji karyawan di perusahaannya. Total gaji yang diterima karyawan berasal dari perhitungan gaji pokok, uang makan, dan uang transportasi sebagai berikut:

Total gaji = gaji pokok + uang makan + uang transportasi



-
- Jika total gaji yang diterima oleh karyawan lebih dari sama dengan Rp 1.500.000, maka gaji dipotong pajak sebesar 5%, sedangkan karyawan dengan total gaji di bawah Rp 1.500.000 tidak mendapat potongan pajak. Buatlah program untuk membantu pegawai keuangan dalam menghitung gaji bersih yang diterima oleh setiap karyawan!
4. Buatlah kode program untuk menentukan kelas mahasiswa dengan syarat jika nilai UTS-nya di atas 90 atau nilai UAS-nya sama dengan 100, maka mahasiswa tersebut masuk ke kelas 'A', selain dari itu mahasiswa akan masuk ke kelas 'B' (Masukan hanya berupa nilai UTS dan UAS)!