




Back End Development 1 ○

+

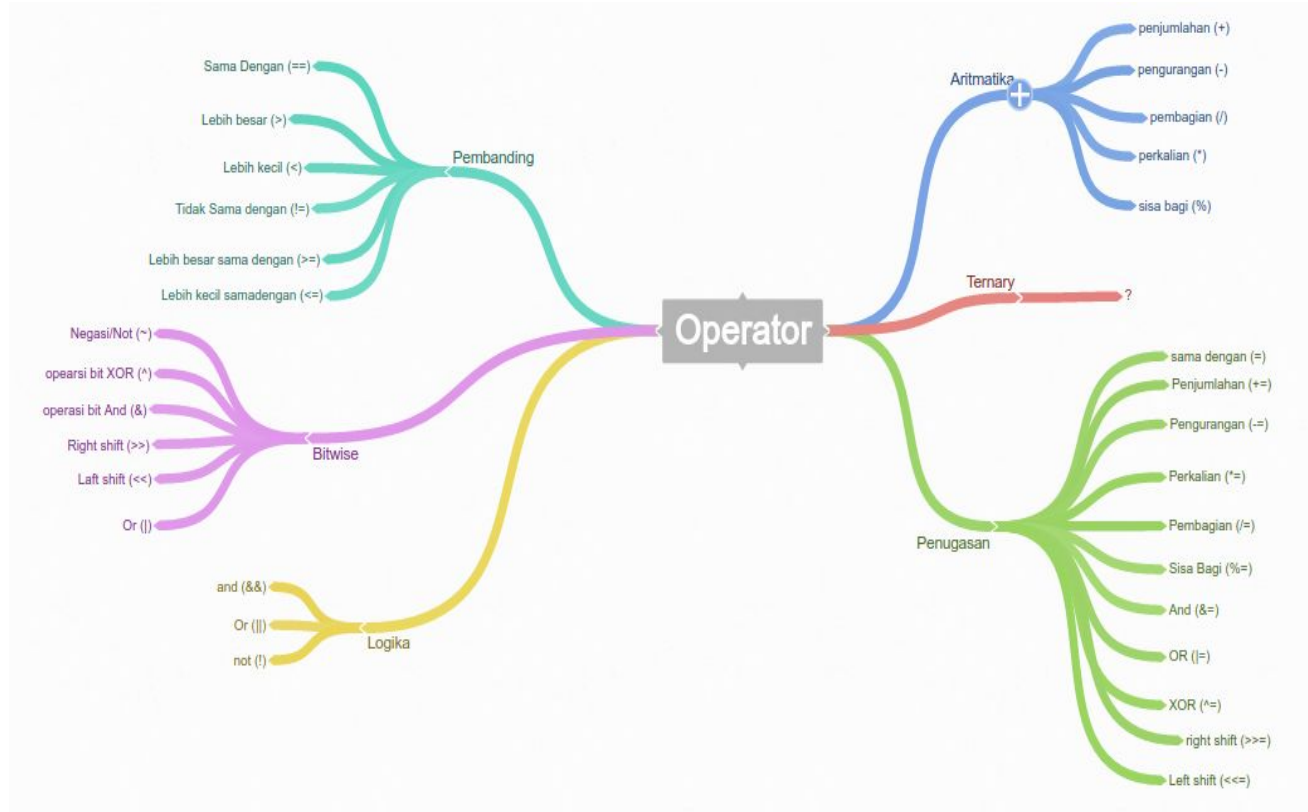
Sesi 2



Java Operator⁺ & Conditional Logic

Java Operator & Conditional Logic - Sesi 2

Java Operator



OPERATOR DALAM JAVA

Operator merupakan suatu simbol yang digunakan untuk memberikan perintah kepada komputer untuk melakukan aksi terhadap satu atau lebih operand.

Operand sendiri adalah sesuatu yang dioperasikan oleh operator.

Di dalam java ada beberapa tipe Operator yang wajib dipahami , yaitu :

- Operator Aritmatika
- Operator Penugasan
- Operator Pembandingan
- Operator Logika

1. Operator Aritmatika

Operator Aritmatika ini adalah operator yang sering kita gunakan untuk hitung menghitung seperti kali bagi tambah kurang dan lainnya .

Nama	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Sisa Bagi	%

Operator Aritmatika

```
public static void main(String[] args) {  
  
    int a = 2 ;  
    int b = 1 ;  
    int tambah, kurang , kali , bagi , sisaPembagian;  
  
    tambah = a + b;  
    kurang = a - b;  
    kali = a * b;  
    bagi = a / b;  
    sisaPembagian = a % b;  
}
```

Operator Aritmatika



```

1  import java.util.Scanner;
2
3  public class OperatorAritmatika {
4
5      public static void main(String[] args) {
6          int angka1;
7          int angka2;
8          int hasil;
9
10         Scanner keyboard = new Scanner(System.in);
11
12         System.out.print("Input angka-1: ");
13         angka1 = keyboard.nextInt();
14         System.out.print("Input angka-2: ");
15         angka2 = keyboard.nextInt();
16
17         // penjumlahan
18         hasil = angka1 + angka2;
19         System.out.println("Hasil = " + hasil);
20
21         System.out.print("Input angka-1: ");
22         angka1 = keyboard.nextInt();
23         System.out.print("Input angka-2: ");
24         angka2 = keyboard.nextInt();
25
26         // pengurangan
27         hasil = angka1 - angka2;
28         System.out.println("Hasil = " + hasil);
29
30         System.out.print("Input angka-1: ");
31         angka1 = keyboard.nextInt();
32         System.out.print("Input angka-2: ");
33         angka2 = keyboard.nextInt();
34
35         // perkalian
36         hasil = angka1 * angka2;
37         System.out.println("Hasil = " + hasil);
38

```

```

40         System.out.print("Input angka-1: ");
41         angka1 = keyboard.nextInt();
42         System.out.print("Input angka-2: ");
43         angka2 = keyboard.nextInt();
44
45         // Pembagian
46         hasil = angka1 / angka2;
47         System.out.println("Hasil = " + hasil);
48
49         System.out.print("Input angka-1: ");
50         angka1 = keyboard.nextInt();
51         System.out.print("Input angka-2: ");
52         angka2 = keyboard.nextInt();
53
54         // Sisa Bagi
55         hasil = angka1 % angka2;
56         System.out.println("Hasil = " + hasil);
57
58     }
59
60 }

```



Java Operator & Conditional Logic - Sesi 2

Notes

++ (Increment)	Meningkatkan nilai operan sebesar 1.	x++ akan menghasilkan 20
-- (Decrement)	Mengurangi nilai operan sebesar 1.	x-- akan menghasilkan 21

```
public class newExample {  
  
    public static void main(String args[]) {  
        int x = 20;  
        int y = 40;  
  
        System.out.println("x + y = " + (x + y) );  
        System.out.println("x - y = " + (x - y) );  
        System.out.println("x * y = " + (x * y) );  
        System.out.println("y / x = " + (y / x) );  
        System.out.println("y % x = " + (y % x) );  
        System.out.println("x++   = " + (x++) );  
        System.out.println("x--   = " + (x--) );  
  
        // perbedaan x++ dengan ++x  
        System.out.println("x++   = " + (x++) );  
        System.out.println("++x   = " + (++x) );  
    }  
}
```

Untuk lebih memahami operator ++ , -- => silahkan jalan sendiri program dan lihat hasil compiler java nya.



Java Operator & Conditional Logic - Sesi 2

2. Relational Penugasan

Assignment Operator digunakan untuk memberi tugas suatu variable untuk melakukan suatu proses. Operator ini sering digunakan dalam pemrograman untuk mengulangi suatu perintah , salah satunya adalah increment / decrement.

Nama	Simbol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pembagian	/=
Pengisian dan Sisa Bagi	%=

Operator Penugasan

```
public class Operator {  
  
    public static void main(String[] args) {  
  
        int a = 2 ;  
        int b = 1 ;  
        int c = 2 ;  
        int d = 1 ;  
        int e = 1 ;  
  
        if(a==2) {           //comment : jika A sama dengan 2 maka ...  
            a = 10 ;         // a sama dengan 10  
            b += 1 ;         // b ditambah 1  
            c -= 1 ;         // c dikurang 1  
            d /= 2 ;         // d dibagi 2  
            e %= 1;         // e dapat hasil dari sisa bagi antara a dan 1  
        }  
    }  
}
```

Operator Penugasan / Assignment Operator


```

1 public class OperatorPenugasan {
2
3     public static void main(String[] args) {
4         int a;
5         int b;
6
7         // Pengisian nilai
8         a = 5;
9         b = 10;
10
11        // penambahan
12        b += a;
13        // sekarang b = 15
14        System.out.println("Penambahan : " + b);
15
16        // pengurangan
17        b -= a;
18        // sekarang b = 10 (karena 15-5)
19        System.out.println("Pengurangan : " + b);
20
21        // perkalian
22        b *= a;
23        // sekarang b = 50 (karena 10*5)
24        System.out.println("Perkalian : " + b);
25
26        // Pembagian
27        b /= a;
28        // sekarang b=10
29        System.out.println("Pembagian : " + b);
30
31        // Sisa bagi
32        b %= a;
33        // sekarang b=0
34        System.out.println("Sisa Bagi: " + b);
35
36    }
37
38 }

```



Java Operator & Conditional Logic - Sesi 2

Notes

Operator	Deskripsi	Contoh
<code>==</code> (equal to)	Memeriksa apakah nilai dua operan sama atau tidak, jika ya maka kondisi menjadi benar. Memeriksa apakah nilai operan kiri lebih besar dari nilai operan kanan, jika ya maka kondisi menjadi benar.	<code>(x == y)</code> bernilai benar
<code>!=</code> (not equal to)	Memeriksa apakah nilai dua operan sama atau tidak, jika nilainya tidak sama maka kondisi menjadi benar.	<code>(x != y)</code> bernilai benar
<code>></code> (greater than)	Memeriksa apakah nilai operan kiri lebih besar dari nilai operan kanan, jika ya maka kondisi menjadi benar.	<code>(x > y)</code> bernilai tidak benar
<code><</code> (less than)	Memeriksa apakah nilai operan kiri lebih kecil dari nilai operan kanan, jika ya maka kondisi menjadi benar.	<code>(x < y)</code> bernilai benar
<code>>=</code> (greater than or equal to)	Memeriksa apakah nilai operan kiri lebih besar atau sama dengan nilai operan kanan, jika ya maka kondisi menjadi benar.	<code>(x >= y)</code> bernilai tidak benar
<code><=</code> (less than or equal to)	Memeriksa apakah nilai operan kiri kurang dari atau sama dengan nilai operan kanan, jika ya maka kondisi menjadi benar.	<code>(x <= y)</code> bernilai benar



Java Operator & Conditional Logic - Sesi 2

Notes

Silahkan coba code berikut dan jalankan program java anda, apa yang anda temui ?

```
public class newExample {  
  
    public static void main(String args[]) {  
        int x = 20;  
        int y = 40;  
  
        System.out.println("x == y = " + (x == y) );  
        System.out.println("x != y = " + (x != y) );  
        System.out.println("x > y = " + (x > y) );  
        System.out.println("x < y = " + (x < y) );  
        System.out.println("y >= x = " + (y >= x) );  
        System.out.println("y <= x = " + (y <= x) );  
    }  
}
```



3. Operator Pembandingan

Operator Pembandingan merupakan suatu operator yang digunakan untuk membandingkan antara dua buah nilai . Hasil perbandingan dari operator ini adalah TRUE dan FALSE (Tipe data BOOLEAN)

Nama	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	==
Tidak Sama Dengan	!=
Lebih Besar Sama Dengan	>=
Lebih Kecil Sama Dengan	<=

Operator Pembandingan

```
public static void main(String[] args) {
```

```
    int a = 2 ;
```

```
    int b = 2 ;
```

```
    boolean c = a == b ;
```

```
    boolean d = a != b ;
```

```
    boolean e = a > b ;
```

```
    boolean f = a < b ;
```

```
    boolean g = a >= b ;
```

```
    boolean h = a <= b ;
```

Operator Pembandingan



```

1  public class OperatorPembanding {
2
3      public static void main(String[] args) {
4          int nilaiA = 12;
5          int nilaiB = 4;
6          boolean hasil;
7
8          // apakah A lebih besar dari B?
9          hasil = nilaiA > nilaiB;
10         System.out.println(hasil);
11
12         // apakah A lebih kecil dari B?
13         hasil = nilaiA < nilaiB;
14         System.out.println(hasil);
15
16         // apakah A lebih besar samadengan B?
17         hasil = nilaiA >= nilaiB;
18         System.out.println(hasil);
19
20         // apakah A lebih kecil samadengan B?
21         hasil = nilaiA <= nilaiB;
22         System.out.println(hasil);
23
24         // apakah nilai A sama dengan B?
25         hasil = nilaiA == nilaiB;
26         System.out.println(hasil);
27
28         // apakah nilai A tidak samadengan B?
29         hasil = nilaiA != nilaiB;
30         System.out.println(hasil);
31
32     }
33
34 }

```



Java Operator & Conditional Logic - Sesi 2

Notes

Operator	Description	Example
& (bitwise and)	Biner AND Operator menyalin sedikit ke hasil jika ada di kedua operan.	(x & y) ia akan menghasilkan 10 yang mana merupakan 1010
(bitwise or)	Biner ATAU Operator menyalin sedikit jika ada di salah satu operan.	(x y) ia akan menghasilkan 15 yang mana merupakan 1111
^ (bitwise XOR)	Operator XOR Biner menyalin bit jika diatur dalam satu operan tetapi tidak keduanya.	(x ^ y) ia akan menghasilkan 5 yang mana 101
~ (bitwise compliment)	Operator Pelengkap Binary Ones adalah unary dan memiliki efek 'membalik' bit.	(~x) ia akan menghasilkan -11 yang mana bernilai 0101 dalam bentuk komplemen 2 karena bilangan biner bertanda.
<< (left shift)	Operator Shift Kiri Biner. Nilai operan kiri dipindahkan ke kiri dengan jumlah bit yang ditentukan oleh operan kanan.	x << 2 ia akan menghasilkan 40 yang bernilai 101000
>> (right shift)	Operator Shift Kanan Biner. Nilai operan kiri dipindahkan ke kanan dengan jumlah bit yang ditentukan oleh operan kanan.	x >> 2 ia akan menghasilkan 2 yang bernilai 10
>>> (zero fill right shift)	Geser operator pengisian nol ke kanan. Nilai operan kiri dipindahkan ke kanan dengan jumlah bit yang ditentukan oleh operan kanan dan nilai yang digeser diisi dengan nol.	x >>> 2 ia akan menghasilkan 2 yang bernilai 10



Java Operator & Conditional Logic - Sesi 2

Notes

Silahkan coba code berikut dan jalankan program java anda, apa yang anda temui ?

```
public class newExample {  
  
    public static void main(String args[]) {  
        int x = 10;  
        int y = 15;  
        int hasil = 0;  
  
        hasil = x & y;  
        System.out.println("x & y = " + hasil );  
  
        hasil = x | y;  
        System.out.println("x | y = " + hasil );  
  
        hasil = x ^ y;  
        System.out.println("x ^ y = " + hasil );  
  
        hasil = ~x;  
        System.out.println("~x = " + hasil );  
  
        hasil = x << 2;  
        System.out.println("x << 2 = " + hasil );  
  
        hasil = x >> 2;  
        System.out.println("x >> 2 = " + hasil );  
  
        hasil = x >>> 2;  
        System.out.println("x >>> 2 = " + hasil );  
    }  
}
```



Java Operator & Conditional Logic - Sesi 2

Java Operator

Operator logika adalah suatu operator yang membandingkan dua bukan KONDISI , mirip seperti dengan operator pembandingan. Operator Logika ini menghasilkan nilai bertipe BOOLEAN , yaitu TRUE dan FALSE.

Operator	Deskripsi	Contoh
&& (logical and)	Disebut operator logika AND. Jika kedua operan bukan nol, maka kondisinya menjadi benar.	(x && y) bernilai false
(logical or)	Disebut Logis ATAU Operator. Jika salah satu dari dua operan bukan nol, maka kondisinya menjadi benar.	(x y) bernilai true
! (logical not)	Disebut Operator BUKAN Logis. Gunakan untuk membalikkan keadaan logis operandnya. Jika suatu kondisi benar maka operator Logical NOT akan membuat false.	!(x && y) bernilai true



Java Operator & Conditional Logic - Sesi 2

Notes

Silahkan coba code berikut dan jalankan program java anda, apa yang anda temui ?

```
public class newExample {  
  
    public static void main(String args[]) {  
        boolean x = true;  
        boolean y = false;  
  
        System.out.println("x && y = " + (x&&y));  
        System.out.println("x || y = " + (x||y) );  
        System.out.println("!(x && y) = " + !(x && y));  
    }  
}
```



--- Conditional + Logic

Java Operator & Conditional Logic - Sesi 2

Conditional Logic

Dalam bahasa pemrograman Java, untuk menganalisa dan mengevaluasi kondisi digunakan variabel boolean. Dalam hal ini, nilai boolean true dan false dikembalikan pada saat melakukan perbandingan ekspresi atau dievaluasi.

```
int x = 5;
boolean y = x == 5;

if (y) {
    System.out.println("True");
}
```

Dalam kondisi normal, memang pada umumnya Anda tidak perlu menetapkan ekspresi kondisional ke dalam boolean. Pada kondisi normal, Anda dapat hanya menggunakan versi pendek seperti sintaks di bawah ini

```
int x = 5;

if (x == 5) {
    System.out.println("x bernilai 5!");
}
```



Berikut contoh Java conditional pada operator Boolean:

```
int x = 5;
int y = 10;
boolean hasil;
hasil = x < y; // true
hasil = x > y; // false
hasil = x <= 5; // x lebih kecil atau sama dengan 5 - true
hasil = y >= 11; // y lebih besar atau sama dengan 11 - false
hasil = x == y; // x sama dengan y - false
hasil = x != y; // x tidak sama dengan y - true
hasil = x > y || x < y; // Logika or - true
hasil = 3 < x && x < 6; // Logika and - true
hasil = !hasil; // Logika not - false
```



Java Operator & Conditional Logic - Sesi 2

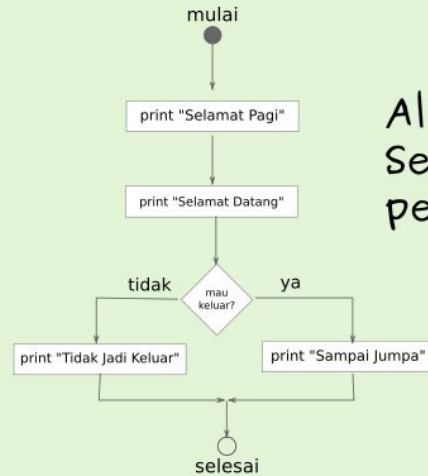
If - Else and Between

Java conditional berikutnya adalah *if* – *else* dan *between*. Untuk pernyataan *if* dan *else* dalam bahasa pemrograman Java ini cukup sederhana.

```
if (x == y) {  
    // x dan y bernilai sama, maka ....  
}
```

Selain itu, dalam kondisi tertentu Anda dapat menambahkan pernyataan lain (*else*) setelah *if*. Ini dapat Anda lakukan ketika Anda ingin melakukan sesuatu jika kondisinya tidak benar

```
if (x == y) {  
    // x dan y bernilai sama, maka ....  
} else {  
    // x dan y tidak bernilai sama, maka ....  
}
```



Alur Program Sederhana dengan percabangan

Lalu bagaimana cara handle-conditional-structures di Java ?

Gunakan kata kunci **if**, **else**, **switch**, **and case**. *ternary operator

Percabangan - IF

Percabangan yang hanya memiliki 1 pilihan.

```

16  import java.util.Scanner;
17
18  public class Hadiah {
19
20      public static void main(String[] args) {
21
22          // membuat variabel belanja dan scanner
23          int belanja = 0;
24          Scanner scan = new Scanner(System.in);
25
26          // mengambil input
27          System.out.print("Total Belanjaan: Rp ");
28          belanja = scan.nextInt();
29
30          // cek apakah dia belanja di atas 100000
31          if ( belanja > 100000 ) {
32              System.out.println("Selamat, anda mendapatkan hadiah!");
33          }
34
35          System.out.println("Terima kasih...");
36
37      }
38
39  }

```

Contoh Program Hadiah.

Beri Nilai Belanja 150.000 dan lihat apa yang terjadi!



```

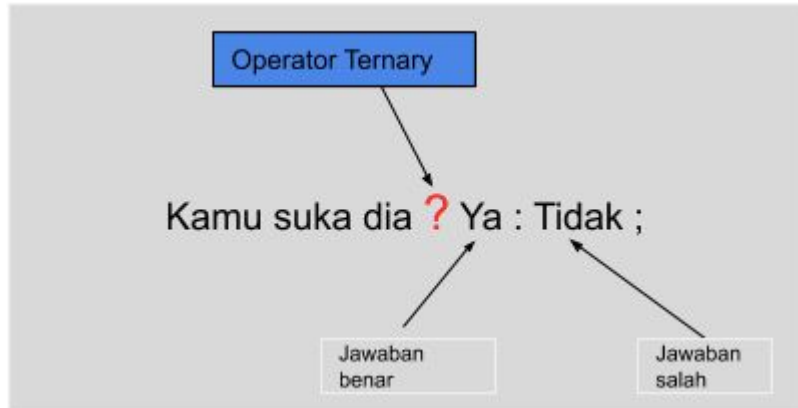
41 import java.util.Scanner;
42
43 public class CekGraduation {
44
45     public static void main(String[] args) {
46
47         // membuat variabel dan Scanner
48         int nilai;
49         String nama;
50         Scanner scan = new Scanner(System.in);
51
52         // mengambil input
53         System.out.print("Nama: ");
54         nama = scan.nextLine();
55         System.out.print("Nilai: ");
56         nilai = scan.nextInt();
57
58         // cek apakah dia lulus atau tidak
59         if( nilai >= 70 ) {
60             System.out.println("Selamat " + nama + ", anda lulus!");
61         } else {
62             System.out.println("Maaf " + nama + ", anda gagal");
63         }
64     }
65 }
66
67 }

```

Percabangan IF/ELSE dengan
operator Ternary

*Seperti yang pernah dibahas pada
Operator Ternary yang juga
menggunakan IF/ELSE.





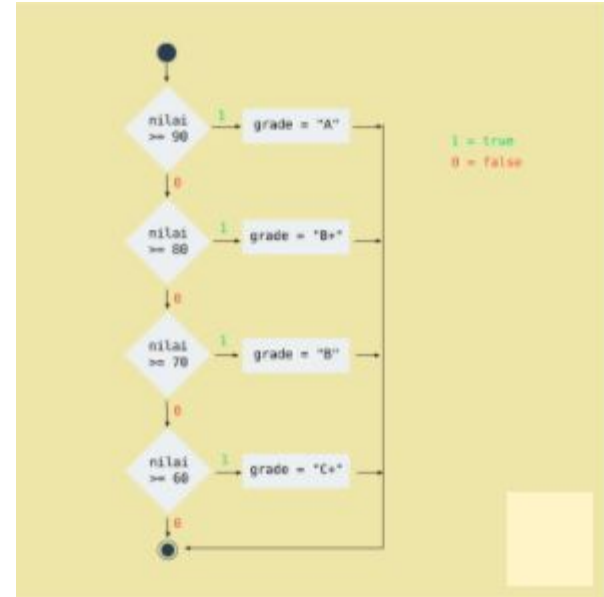
```
1 public class OperatorTernary {
2     public static void main(String[] args) {
3
4         boolean suka = true;
5         String jawaban;
6
7         // menggunakan operator ternary
8         jawaban = suka ? "iya" : "tidak";
9
10        // menampilkan jawaban
11        System.out.println(jawaban);
12    }
13 }
14 }
```



```

72  if (suatu kondisi) {
73      // maka kerjakan ini
74      // kerjakan perintah ini juga
75      // ...
76  } else if (kondisi lain) {
77      // kerjakan ini
78      // kerjakan ini juga
79      // ...
80  } else if (kondisi yang lain lagi) {
81      // kerjakan perintah ini
82      // kerjakan ini juga
83      // ...
84  } else {
85      // kerjakan ini kalau
86      // semua kondisi di atas
87      // tidak ada yang benar
88      // ...
89  }

```



Jika :

nilainya lebih dari 90, maka grade-nya “A” ,

nilai lebih besar dari 80, maka “B+”,

nilai lebih besar dari 70, maka “B”.



HACKTIV8

```

92  import java.util.Scanner;
93
94  public class HitungGrade {
95      public static void main(String[] args) {
96
97          // membuat variabel dan scanner
98          int nilai;
99          String grade;
100         Scanner scan = new Scanner(System.in);
101
102         // mengambil input
103         System.out.print("Inputkan nilai: ");
104         nilai = scan.nextInt();
105
106         // hitung gradenya
107         if ( nilai >= 90 ) {
108             grade = "A";
109         } else if ( nilai >= 80 ){
110             grade = "B+";
111         } else if ( nilai >= 70 ){
112             grade = "B";
113         } else if ( nilai >= 60 ){
114             grade = "C+";
115         } else if ( nilai >= 50 ){
116             grade = "C";
117         } else if ( nilai >= 40 ){
118             grade = "D";
119         } else {
120             grade = "E";
121         }
122
123         // cetak hasilnya
124         System.out.println("Grade: " + grade);
125     }
126 }

```



Untuk Anda ketahui, pernyataan *if – else* tidak harus Anda tuliskan dalam beberapa baris dengan menggunakan tanda {}, *if* dapat Anda gunakan dalam satu baris, atau tanpa {}, untuk satu baris pernyataan.

```
if (x == y)
    System.out.println("Kalimat utama");
else
    System.out.println("Kalimat pendukung");
```

Ada cara lain untuk menulis pernyataan *if – else* dalam satu baris, namun ini disebut sebagai sisi buruk dari *if*. Berikut contoh cara yang dapat digunakan:

```
int x = 5;
int hasil = x == 5 ? 10 : 20;

// maka hasilnya adalah 10
// Ini sama dengan

int hasil;

if (x == 5) {
    hasil = 10;
} else {
    hasil = 20;
}
```



Java Operator & Conditional Logic - Sesi 2

== and Equals

Operator Java conditional berikutnya ialah = dan equals. Untuk operator = sendiri, memiliki cara kerja yang sedikit berbeda pada objek daripada variabel tipe primitif. Ketika Anda ingin menggunakan objek dan ingin memastikan dengan cara memeriksa apakah mereka sama, operator = akan mengatakan jika mereka sama.

Apabila Anda ingin memeriksa apakah keduanya memang sama secara logis, maka cara yang harus Anda tempuh ialah dengan menggunakan equals metode pada objek.

```
String x = new String("Test");  
String y = new String("Test");  
String sameX = x;  
  
boolean hasil1 = x == y;           // Ini false, karena x dan y bukanlah objek yang sama  
boolean hasil2 = x.equals(y);      // Ini true, karena x dan y secara logis sama  
boolean hasil3 = x == sameX;       // Ini true, karena x dan sameX adalah objek yang sama  
a
```

Java Operator & Conditional Logic - Sesi 2

Switch/Case

Contoh Format SWITCH CASE

```
1  switch(variabel){  
2      case 1:  
3          // kerjakan kode ini  
4          // kode ini juga  
5          break;  
6      case 2:  
7          // kerjakan kode ini  
8          // kode ini juga  
9          break;  
10     case 3:  
11         // kerjakan kode ini  
12         // kode ini juga  
13         break;  
14     default:  
15         // kerjakan kode ini  
16         // kode ini juga  
17         break;
```

Perhatikan **case 1** artinya nilai variabel yang akan dibandingkan apakah nilai nya sama dengan 1 atau tidak. Kalau iya, maka kerjakan kode yang ada di dalam **case 1**.

```
1  switch (variabel) {  
2      case 'A':  
3          // lakukan sesuatu  
4          break;  
5      case 'B':  
6          // lakukan ini  
7          break;  
8      default:  
9          // lakukan ini  
10 }
```

Perlu diperhatikan juga, ada **break**; dan **default**:
Break; artinya berhenti, ini berarti system memerintahkan untuk berhenti menghentikan pengecekan **case** yang lain/selanjutnya.

Default: artinya jika nilai variabel tidak ada yang sama dengan pilihan case di atas, maka kerjakan kode yang ada didalam **default**.

Pilihan **default** bisa juga tidak memiliki **break**, karena dia adalah pilihan terakhir atau pengecekan akan berakhir disitu.

```

1  import java.util.Scanner;
2
3  public class LampuLalulintas {
4      public static void main(String[] args) {
5
6          // membuat variabel dan Scanner
7          String lampu;
8          Scanner scan = new Scanner(System.in);
9
10         // mengambil input
11         System.out.print("Inputkan nama warna: ");
12         lampu = scan.nextLine();
13
14         switch(lampu){
15             case "merah":
16                 System.out.println("Lampu merah, berhenti!");
17                 break;
18             case "kuning":
19                 System.out.println("Lampu kuning, harap hati-hati!");
20                 break;
21             case "hijau":
22                 System.out.println("Lampu hijau, silahkan jalan!");
23                 break;
24             default:
25                 System.out.println("Warna lampu salah!");
26         }
27     }
28 }

```

Must try :

Cobalah untuk menghilangkan break di salah satu case dan perhatikanlah hasilnya.



Java Operator & Conditional Logic - Sesi 2

Nested IF

Misalnya ada model bisnis 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

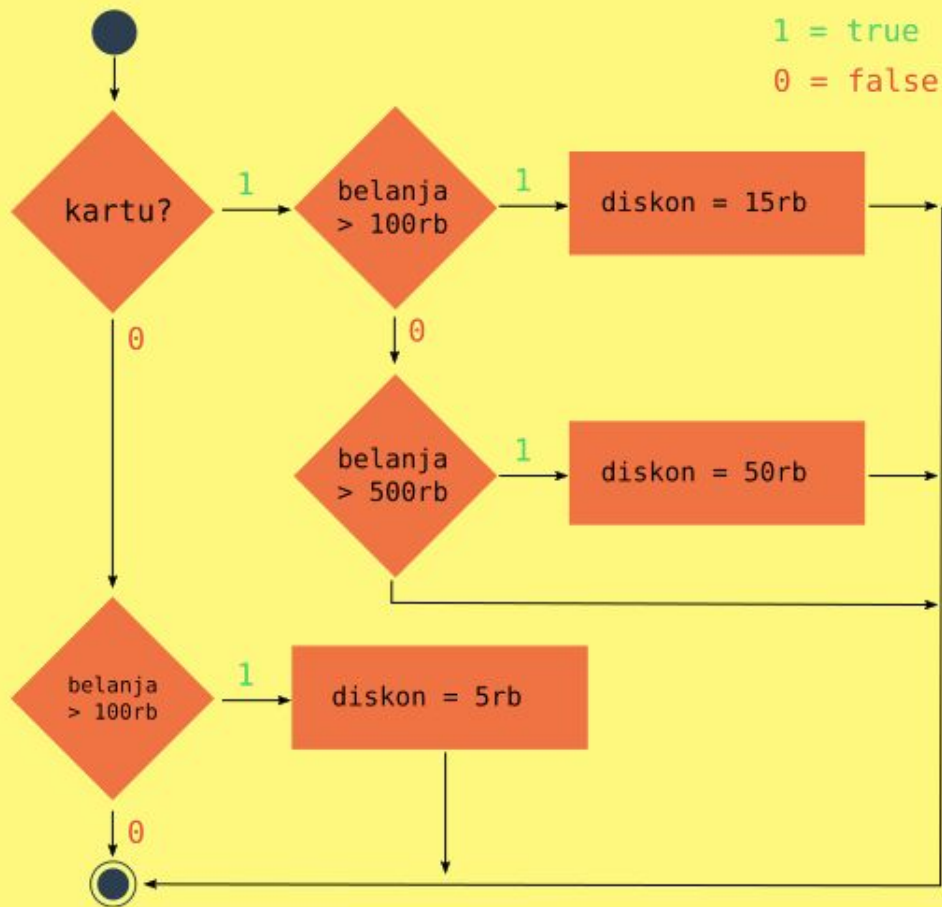
- tidak

- * Apakah belanjaan anda lebih dari 100rb?

- # ya : mendapatkan diskon 10rb

- # tidak: tidak mendapatkan diskon





```

1  import java.util.Scanner;
2
3  public class Kasir {
4      public static void main(String[] args) {
5          // deklarasi variabel dan Scanner
6          int belanjaan, diskon, bayar;
7          String kartu;
8          Scanner scan = new Scanner(System.in);
9
10         // mengambil input
11         System.out.print("Apakah ada kartu member: ");
12         kartu = scan.nextLine();
13         System.out.print("Total belanjaan: ");
14         belanjaan = scan.nextInt();
15
16         // proses
17         if (kartu.equalsIgnoreCase("ya")) {
18             if (belanjaan > 500000) {
19                 diskon = 50000;
20             } else if (belanjaan > 100000) {
21                 diskon = 15000;
22             } else {
23                 diskon = 0;
24             }
25
26         } else {
27             if (belanjaan > 100000) {
28                 diskon = 5000;
29             } else {
30                 diskon = 0;
31             }
32         }
33
34         // total yang harus dibayar
35         bayar = belanjaan - diskon;
36
37         // output
38         System.out.println("Total Bayar: Rp " + bayar);
39     }
40 }

```



Perlu di perhatikan :

A. Fungsi `equalsIgnoreCase("ya")` digunakan untuk membandingkan string tanpa memperhatikan huruf besar dan huruf kecilnya.

B. Ada juga Fungsi `equals()`, fungsinya sama. Tapi `equals` lebih memperhatikan case hurufnya.