



Pertemuan 3

Variabel, Tipe Data, Operator

dan Input-Output

Tim Ajar Dasar Pemrograman 2024



Tujuan

- Mahasiswa memahami dan mampu menjelaskan tentang Tipe Data
- Mahasiswa menjabarkan dan mampu menjelaskan tentang Variabel
- Mahasiswa memahami dan mampu menguraikan tentang Operator (Penugasan Aritmatika, Penugasan Gabungan, Increment, Decrement, Relasional, Logika, Kondisional, Bitwise)

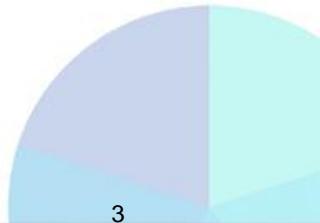


Variabel

- Variable digunakan dalam bahasa pemrograman untuk menyimpan nilai sementara dimana untuk digunakan kembali nantinya.
- Variabel memiliki tipe data dan nama.
- tipe data mengindikasikan tipe dari nilai pada variabel tersebut.

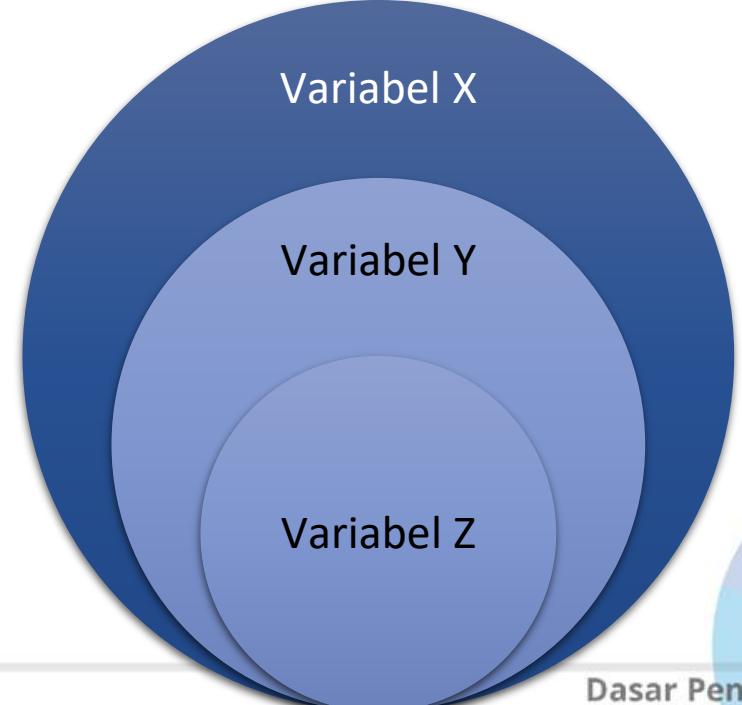


Apa yang anda bayang dengan gambar di samping?



Jenis Variabel

- Variable lokal adalah variable yang hanya bisa dikenali pada sub program
- Variabel global adalah variable yang dapat dikenali pada keseluruhan program





Aturan Penulisan Variabel

- Nama variable tidak boleh menggunakan kata kunci Java
- Nama variable boleh menggunakan huruf, angka(0-9), garis bawah(_), dan symbol dolar(\$), namun sebaiknya penggunaan
 - ✓ Nama variable menggunakan diawali huruf kecil
 - ✓ Apabila nama variable lebih dari satu kata maka kata yang setelahnya diawali huruf besar.

❖ Bentuk:

<tipe data> <nama> [=nilai awal]

nilai dalam tanda [] bersifat optional.

contoh:

```
int contVariabel;
```

```
int contohVar = 34;
```



Tipe Data

- Tipe data adalah jenis data yang ingin kita simpan di variabel.
- Tipe data dapat dikategorikan menjadi dua kelompok, yaitu
 1. *tipe data Primitif*
 2. *tipe data Referensi.*



Apa yang anda bayangkan dengan gambar di samping?



Bagimana dengan gambar yang ini?



Tipe data primitif

Jenis Data	Deskripsi	Ukuran	Minimum	Maksimum
boolean	true / false	1-bit		
char	Karakter Unicode	16-bit		
byte	Bilangan bulat	8-bit	-127	128
short	Bilangan bulat	16-bit	-32768	32767
int	Bilangan bulat	32-bit	-2147483648	2147483647
long	Bilangan bulat	64-bit	- 9223372036854775807 5808	
float	Bilangan riil	32-bit	1.40129846432481 707e-45	3.40282346638528860e +38
double	Bilangan riil	64-bit	4.94065645841246 544e-324	1.79769313486231570e +308



DEKLARASI

-----Deklarasi-----

```
int nilai;  
double angka;  
float a, b, c;
```

-----Pemberian nilai-----

```
int nilai=75;  
double angka=2.5;
```



Mencetak Variabel

```
System.out.println(nilai);  
System.out.println(a);  
-----atau-----  
System.out.println("Nilai anda adalah" +nilai);  
System.out.println("angka adalah" +a);
```



Casting tipe data

Casting adalah ketika kita ingin memberikan nilai dari tipe data primitive ke tipe data primitive yang lain

- Widening casting (otomatis) – mengubah tipe data dari yang ukurannya lebih kecil ke tipe data yang lebih besar
byte -> short -> char -> int -> long -> float -> double



Ilustrasi widening casting

Casting tipe data(2)

- Narrowing casting (manual) – mengubah tipe data dari yang ukurannya lebih besar ke tipe data yang lebih kecil
 - double -> float -> long -> int -> char -> short -> byte



Ilustrasi narrowing casting



Contoh Casting tipe data

- Widening casting(otomatis)

```
byte umur = 9;
double myDouble = umur;
System.out.println(umur);      // Outputs 9
System.out.println(myDouble);   // Outputs 9.0
```

- Narrowing casting(manual)

```
double ipk = 3.78;
int myInt = (int) ipk;
System.out.println(ipk);    // Outputs 3.78
System.out.println(myInt);   // Outputs 3
```



Kegunaan ASCII ??

- ASCII adalah singkatan dari American Standard Code for Information Interchange.
- Sesuai dengan namanya, ASCII digunakan untuk pertukaran informasi dan komunikasi data.
- ASCII merupakan kode angka yang mewakili sebuah karakter.



USASCII code chart



Tipe data referensi

- Tipe data non-primitive dibuat berdasarkan kebutuhan programmer.
- Nilai bawaan non-primitive adalah null
- Pendeklarasian tipe data ini hampir sama dengan deklarasi pada tipe data primitif.
- Tipe data non-primitive diawali dengan huruf besar





Ciri khas tipe data referensi adalah kemampuannya menampung banyak nilai.

Pada tipe data primitif , nilai yang bisa ditampung Cuma 1 saja. Perhatikan contoh berikut ini:

Tipe Primitif :

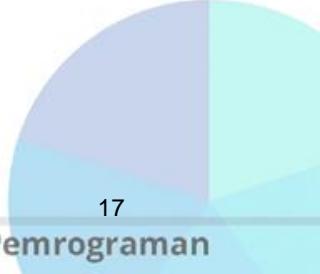
int x = 9; (ada 1 nilai saja, yaitu angka 9)

char hurufku = "h"; (ada 1 nilai saja, yaitu huruf h)

Tipe Referensi :

String tulisan = "Aku Belajar Java"; (ada 16 nilai, termasuk spasi)

int[] daftar = { 1, 4, 9, 16, 25, 36, 49 }; (ada 7 nilai bertipe integer)





Operator

- Operator merupakan simbol yang biasa digunakan dalam menulis suatu pernyataan (*statement*) dalam bahasa pemrograman apapun. Operator akan melakukan suatu operasi terhadap operand sesuai dengan fungsinya.
- Contoh operasi antara lain penjumlahan, pengurangan, pembagian dan sebagainya.

3 + 8 * 4

3 8 4 adalah operand

+ * adalah Operator



Jenis operator

1. Operator Aritmatika
2. Operator Increment dan Decrement
3. Operator Assignment
4. Operator Relasi
5. Operator Logika
6. Operator Bitwise



1. Operator Aritmatika

Arithmatic operator (operator aritmatika) adalah operator yang berfungsi untuk operasi aritmatika.

Arithmatic Operator	Description
+	plus
-	minus
*	point
/	divide
%	modulus



2. Operator Increment dan Decrement

Operator Increment dan Decrement digunakan untuk menaikan atau menurunkan suatu nilai integer (bilangan bulat) sebanyak satu satuan, dan hanya dapat digunakan pada variabel.

Operator	Use	Description
++	a++	Menaikan/menambah 1 nilai setelah operasi dilakukan
	++a	Menaikan/menambah 1 nilai sebelum operasi dilakukan
--	a--	Penurunan/mengurangi 1 nilai setelah operasi dilakukan
	--a	Penurunan/mengurangi 1 nilai sebelum operasi dilakukan



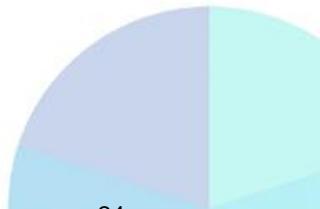


3. Operator Assignment

Operator assignment dalam Java digunakan untuk memberikan sebuah nilai ke sebuah variabel. Operator assignment hanya berupa '=' , shortcut assignment operator yang penting, yang digambarkan dalam

tabel berikut :

Operator	Penggunaan	Ekuivalen Dengan
<code>+=</code>	<code>Op1 += Op2</code>	<code>Op1 = Op1 + Op2</code>
<code>-=</code>	<code>Op1 -= Op2</code>	<code>Op1 = Op1 - Op2</code>
<code>*=</code>	<code>Op1 *= Op2</code>	<code>Op1 = Op1 * Op2</code>
<code>/=</code>	<code>Op1 /= Op2</code>	<code>Op1 = Op1 / Op2</code>
<code>%=</code>	<code>Op1 %= Op2</code>	<code>Op1 = Op1 % Op2</code>
<code>&=</code>	<code>Op1 &= Op2</code>	<code>Op1 = Op1 & Op2</code>
<code> =</code>	<code>Op1 = Op2</code>	<code>Op1 = Op1 Op2</code>
<code>^=</code>	<code>Op1 ^= Op2</code>	<code>Op1 = Op1 ^ Op2</code>
<code><<=</code>	<code>Op1 <<= Op2</code>	<code>Op1 = Op1 << Op2</code>
<code>>>=</code>	<code>Op1 >>= Op2</code>	<code>Op1 = Op1 >> Op2</code>
<code>>>>=</code>	<code>Op1 >>>= Op2</code>	<code>Op1 = Op1 >>> Op2</code>





Contoh

a = a+5 bisa dipersingkat menjadi a += 5

b = b-5 bisa dipersingkat menjadi b -= 5

c = c*5 bisa dipersingkat menjadi c *= 5

d = d/5 bisa dipersingkat menjadi d /= 5

e = e%5 bisa dipersingkat menjadi e %= 5



4. Operator Relasi

Operator relasi dalam Java digunakan untuk menghasilkan nilai boolean yang sering digunakan untuk mengatur alur jalannya sebuah program.

Operator	Penggunaan	Deskripsi
>	$Op1 > Op2$	Menghasilkan true jika Op1 lebih besar dari Op2
<	$Op1 < Op2$	Menghasilkan true jika Op1 lebih kecil dari Op2
\geq	$Op1 \geq Op2$	Menghasilkan true jika Op1 lebih besar atau sama dengan Op2
\leq	$Op1 \leq Op2$	Menghasilkan true jika Op1 lebih kecil atau sama dengan Op2
\equiv	$Op1 \equiv Op2$	Menghasilkan true jika Op1 sama dengan Op2
\neq	$Op1 \neq Op2$	Menghasilkan true jika Op1 tidak sama dengan Op2

5. Operator Logika

Operator ini digunakan untuk ekspresi logik yang menghasilkan nilai boolean.

Operator-operator yang digunakan adalah AND (`&&`), OR (`||`) dan NOT (`!`).

Operator	Deskripsi	Contoh
<code>&&</code>	and	<code>x=6</code> <code>y=3</code> <code>(x < 10 && y > 1)</code> hasil true
<code> </code>	or	<code>x=6</code> <code>y=3</code> <code>(x==5 y==5)</code> hasil false
<code>!</code>	not	<code>x=6</code> <code>y=3</code> <code>!(x==y)</code> hasil true



6. Operator Bitwise

Operator ini digunakan untuk melakukan manipulasi bit dari sebuah bilangan

- Bitwise OR(|)

Hasil bit bernilai 1 ketika salah satu bit-bit bernilai 1, selain itu bernilai 0.

Contoh:

```
int a = 5;//0101
int b = 7;//0111
System.out.println(a|b);//output 7
//0101
//0111
//_____
//0111 -> 7
```



6. Operator Bitwise(2)

- Bitwise AND(&)

Hasil bit bernilai 1 ketika semua bit-bit bernilai 1, selain itu bernilai 0.

Contoh:

```
int a = 5;//0101
int b = 7;//0111
System.out.println(a&b);//output 5
//0101
//0111
//_____
//0101 -> 5
```



6. Operator Bitwise(3)

- Bitwise XOR(^)

Nilai bit bernilai 1 ketika ada bit bernilai 1 dan 0, selain itu bernilai 0.

Contoh:

```
int a = 5;//0101
int b = 7;//0111
System.out.println(a^b);//output 2
//0101
//0111
//_____
//0010 -> 2
```



6. Operator Bitwise(4)

- Bitwise Complement(~)

Nilai bit yang berkebalikan, ketika nilai bit bernilai 1 maka menghasilkan 0 sedangkan bernilai 0 menghasilkan 1.

Contoh:

```
int a = 5;//0101
System.out.println(~a);//output .10
//0101
//_____
//1010 -> 10
```

$$\begin{aligned}\sim n &= -(n+1) \\ \sim(-n) &= (n-1)\end{aligned}$$



Input, Output dan Sequence

Input → Masukan nilai dari luar (program)

Output → Keluaran hasil program

Sequence → Runut, Program berjalan sesuai urutan langkah-Langkah algoritma.

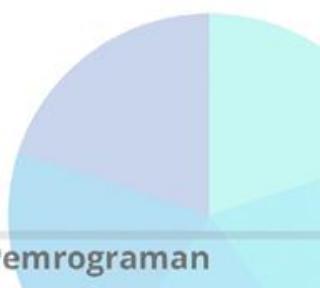
contoh :

Diberikan nilai awal :

$A = 8$

$B = 10$

Bagaimana susunan algoritma sehingga A menyimpan nilai B awal dan B menyimpan nilai A awal





Penggunaan Inputan Pada Java

- Untuk membaca input dari keyboard yaitu dengan menggunakan library Scanner yang di import kedalam program java.
- Caranya adalah dengan menuliskan perintah **import java.util.Scanner** di baris paling atas dari kode program yang akan dibuat.
- Selanjutnya tuliskan perintah deklarasi scanner berikut ini didalam fungsi main() :

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

- Selanjutnya, tergantung dari jenis input yang akan dimasukkan, berupa bilangan bulat (int), bilangan koma (float/double), atau karakter (String).
 1. Jika input berupa bilangan bulat, maka perintahnya adalah: **nextInt();**
 2. Jika input berupa bilangan koma, maka perintahnya adalah: **nextFloat();**
 3. Jika input berupa teks, maka perintahnya adalah: **nextLine();**



Menampilkan Output Pada Java

Untuk menampilkan output ke layar, ada beberapa cara:

1. **System.out.print("Hello world");** Perintah ini akan menampilkan kata Hello world ke layar, atau apapun yang kita tuliskan didalam tanda petik.
2. **System.out.println("Hello world");** Perintah ini akan menampilkan kata Hello world ke layar, atau apapun yang kita tuliskan didalam tanda petik, sekaligus memberi perintah ganti baris di akhir kata/kalimat.
3. **System.out.println(panjang);** Perintah ini akan menampilkan isi variabel panjang ke layar. Perhatikan bahwa untuk menampilkan isi variabel, tidak perlu menggunakan tanda petik (").
4. **System.out.println("Panjang segi empat: " + panjang);** Perintah ini akan menampilkan kalimat "Panjang segi empat: " kemudian disambung dengan isi variabel panjang ke layar. Perhatikan untuk menyambung kalimat dengan isi variabel, digunakan tanda plus (+).



Contoh Studi Kasus

Pak Adi mempunyai sebuah kebun berbentuk Persegi Panjang. Pak Adi ingin membuatkan pagar kayu untuk mengelilingi kebun tersebut. Sebelum membuat program untuk membantu pak adi menghitung keliling kebunnya, maka bantulah pak adi untuk mengidentifikasi variable dan tipe data beserta algoritmany!



Contoh Studi Kasus

1. Menentukan Algoritma Keliling Bangun Persegi

Input: panjang, lebar

Output: keliling

Proses:

1. input panjang, lebar
2. keliling = $2 \times (\text{panjang} + \text{lebar})$
3. Output keliling

2. Mengidentifikasi variable dan jenis tipe data berdasarkan algoritma

Variabel	Tipe data
panjang	int
lebar	int
keliling	int



Latihan

- Ubah soal pada studi kasus di atas untuk bentuk kebun lingkaran



Tugas

1. Susun langkah-langkah algoritma menentukan sebuah bilangan termasuk bilangan negatif atau positif, tentukan nama variabel dan tipe data yang digunakan.
2. Susun langkah-langkah algoritma menentukan sebuah bilangan termasuk bilangan ganjil atau genap, tentukan nama variabel dan tipe data yang digunakan.
3. Susun langkah-langkah algoritma menghitung rata-rata nilai mahasiswa dari 3 nilai mata kuliah dan status kelulusannya dengan syarat kelulusan adalah tidak ada nilai yang di bawah 60 dan rata-rata di atas 70, tentukan nama variabel dan tipe data yang digunakan.