

Bab I

Tipe Data, Variabel, dan Operator

Tujuan

1. Memahami konsep tipe data pada pemrograman komputer
2. Memahami konsep variabel pada pemrograman komputer
3. Melakukan manipulasi data menggunakan operator pemrograman
4. Menggunakan standard input-output pada bahasa pemrograman java

1.1 Variabel

Variabel merupakan representasi dari alamat memori yang digunakan untuk menyimpan nilai dari data. Java adalah bahasa pemrograman yang *Strongly Typed* dan *Statically Typed*, artinya untuk menggunakan variabel, haruslah terlebih dahulu dideklarasikan tipe data dan nama variabelnya. Dalam pendeklarasian variabel, terdapat beberapa hal yang perlu diperhatikan, seperti aturan penamaan variabel, beberapa aturan penamaan variabel dalam bahasa pemrograman java adalah :

- Nama variabel tidak boleh diawali dengan angka dan simbol khusus selain simbol *underscore* (`_`) dan *dollar sign* (`$`).
- Nama variabel tidak boleh mengandung spasi, berdasarkan *Java Naming Convention*, jika variabel terdiri dari beberapa kata, maka penulisannya mengikuti aturan *lowerCamelCase*.
- Nama variabel tidak boleh menggunakan reserved word pada java seperti *int*, *String*, *double*.

1.2 Tipe Data

Tipe Data merupakan konsep pemrograman yang merepresentasikan besarnya ruang serta jenis data yang dapat ditampung di dalam sebuah variabel. Tipe data juga mampu menentukan operasi apa saja yang dapat dilakukan terhadap data tersebut, dengan kata lain, tipe data lah yang menentukan karakteristik dari sebuah variabel. Terdapat tipe data yang mampu menyimpan nilai berupa angka sehingga dapat dilakukan operasi matematika, terdapat pula tipe data yang mampu menyimpan teks namun tidak dapat dilakukan operasi matematika.

Terdapat 2 (dua) jenis tipe data pada Java, yaitu tipe data primitif dan tipe data *reference*.

1.2.1 Tipe Data Primitif

Tipe data primitif adalah tipe data dasar yang telah ada di dalam Java dan diberi nama menggunakan *reserve keyword*, sehingga tidak perlu mengimport library secara eksplisit dan tidak dapat diganggu gugat penamaannya.

Terdapat 8 tipe data primitif pada java yang masing-masing memiliki batas maksimum dan minimum nilai yang dapat disimpan.

Keyword	Nilai Minimum	Nilai Maksimum
boolean	<i>true</i> atau <i>false</i>	
byte	-128	127
short	-32768	32767
int	-2147483648	2147483647
long	-9223372036854775808	9223372036854775807
float	1.4E-45	3.4028235E38
double	4.9E-324	1.7976931348623157E308
char	'\u0000' atau 0	'\uffff' atau 65535

Table 1: Daftar Tipe Data dan Range-nya

Dari tabel di atas, terdapat 6 tipe data yang menyimpan nilai berupa angka, yaitu *byte*, *short*, *int*, dan *long* yang digunakan untuk menyimpan bilangan bulat (*Integer*) serta *float* dan *double* untuk menyimpan bilangan *floating point*. Perbedaan tipe data untuk angka ini adalah *range* serta jumlah memori yang digunakan, sehingga *programmer* dapat lebih efisien dalam penggunaan memori dengan menggunakan tipe data yang tepat sesuai kebutuhan.

Berikut contoh penggunaan tipe data dan variabel pada java:

a) Integer

```
byte umur = 19;  
short tahunLahir = 2000;  
int umurDalamHari = 6935;  
long umurDalamJam = 166440;
```

Drawing 1: Deklarasi dan Inisialisasi Tipe Data Integer

b) Floating Point

```
float piInFloat = 3.1415927f;  
double piInDouble = 3.141592653589793;
```

Drawing 2: Deklarasi dan Inisialisasi Tipe Data Floating Point

c) Boolean

Boolean adalah tipe data yang dapat menampung 2 nilai, yaitu *true* dan *false*, operasi dengan *Logical Operator* juga akan menghasilkan nilai boolean, sehingga tipe data ini biasa digunakan dalam penyeleksian kondisi dan perulangan.

```
boolean sedangHujan = false;  
boolean sedangBerjalan = true;
```

Drawing 3: Deklarasi dan Inisialisasi Tipe Data Boolean

d) Char

Char adalah tipe data yang digunakan dalam java untuk merepresentasikan *Unicode Character*, tipe data ini ditandai dengan petik satu (*Single Quote*) (') atau dengan angka yang juga akan direpresentasikan kembali menjadi *Unicode Character* sesuai dengan tabel ASCII.

```
char inisial = 'f';  
char charInNumber = 102;
```

Drawing 4: Deklarasi dan Inisialisasi Tipe Data Char

1.2.2 Tipe Data Reference

Tipe Data Reference adalah tipe data yang merujuk pada *object* atau *instance class* sehingga variabel yang menggunakan tipe data reference pada dasarnya hanya menyimpan reference/address dari object. Tipe Data Reference juga dapat dibuat sendiri oleh programmer dengan menggunakan konsep Class dan Object, pembahasan lebih mendalam tentang tipe data reference akan dibahas pada mata kuliah Object Oriented Programming, namun terdapat tipe data reference yang bisa langsung digunakan layaknya tipe data primitif, yaitu tipe data String dari class `java.lang.String` yang akan dibahas pada bab 6.

1.3 Operator

Operator pada bahasa pemrograman adalah simbol yang digunakan untuk mengolah dan memanipulasi nilai dari data. Pada bahasa java, terdapat beberapa kategori operator berdasarkan fungsinya, seperti:

- Assignment Operator
- Arithmetic Operator
- Unary Operator
- Relational Operator
- Logical Operator

Berikut Fungsi dan penggunaan dari operator-operator di atas:

1.3.1 Assignment Operator

Assignment Operator adalah operator yang digunakan untuk mengatur alamat dan nilai dari variabel, operator ini ditandai dengan simbol sama dengan ‘=’, simbol ini akan memberikan nilai variabel di sebelah kiri dengan nilai di sebelah kanan.

```
int a = 10;  
int b = a;  
int c = b;
```

Drawing 5: Penggunaan Assignment Operator pada Variabel bertipe int

1.3.2 Arithmetic Operator

Arithmetic Operator adalah operator yang digunakan untuk melakukan proses perhitungan aritmatika. Arithmetic Operator pada java antara lain :

Simbol	Fungsi
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulo

Table 2: Operator Aritmatika

```
int a = 10;
int b = 20;
int c = a + b;
b = c;
float d = 22 / 7f;
long e = Integer.MAX_VALUE + 1L;
c = c % a;
```

Drawing 6: Operasi Aritmatika pada data Numerik

1.3.3 Unary Operator

Unary Operator tidak lain adalah operator aritmatika yang hanya menggunakan satu operand, ditambah dengan fungsi untuk membalikkan nilai boolean. Operator ini biasanya digunakan untuk mempersingkat penulisan kode.

Operator	Contoh	Fungsi
+	A += B atau A += B	Penjumlahan
-	A -= B atau A -= B	Pengurangan
++	A++ atau ++A	Increment, atau peningkatan nilai sebesar 1 poin
--	B-- atau --B	Decremen, atau pengurangan nilai sebesar 1 poin
!	!A	Negasi, membalik nilai boolean

Table 3: Operator Unary

```
int a = 10;
a++
int b += --a;
boolean isFailure = !true;
```

Drawing 7: Operasi Aritmatika dan Negasi menggunakan Unary Operator

1.3.4 Relational Operator

Relational Operator adalah operator yang digunakan untuk menentukan dan membandingkan antara dua variabel, dan hasil dari perbandingannya adalah *true* atau *false*. Berikut Relational Operator pada bahasa Java:

Operator	Fungsi
==	Membandingkan kedua nilai sama
!=	Membandingkan kedua nilai tidak sama
>=	Membandingkan salah satu nilai lebih besar atau sama dengan nilai yang lainnya
<=	Membandingkan salah satu nilai lebih kecil atau sama dengan nilai yang lainnya
>	Membandingkan salah satu nilai lebih besar dari nilai yang lainnya
<	Membandingkan salah satu nilai lebih kecil dari nilai yang lainnya

Table 4: Operator Relasional

```
int a = 10;
int b = 20;
int c = 10;
boolean isEqual = a == b;
boolean isGreaterThan = a > c;
boolean isGreaterOrEquals = a >= c;
```

Drawing 8: Operasi Relasional pada data integer yang disimpan di dalam data boolean

1.3.5 Logical Operator

Logical Operator adalah operator yang digunakan untuk menguji beberapa kondisi menggunakan logika **AND** dan **OR** yang menghasilkan nilai berupa boolean, operator ini biasa juga dikenal sebagai *Conditional Operator*.

Operator	Logika	Fungsi
&&	AND	Mengembalikan nilai <i>true</i> jika dan hanya jika semua kondisi bernilai <i>true</i>
	OR	Mengembalikan nilai <i>true</i> jika semua atau salah satu dari kondisi bernilai <i>true</i>

Table 5: Operator Logika

Selain 2 operator logika di atas, terdapat pula operator yang fungsinya mirip dengan operator logika, yaitu *Bitwise Operator*, namun operator ini melakukan operasinya pada level biner, beberapa Bitwise Operator pada java adalah:

Operator	Nama	Fungsi
&	AND	Nilai 1 jika kedua nilai adalah 1 dan 0 jika tidak
	OR	Nilai 1 jika keduanya atau salah satu nilai adalah 1 dan 0 jika tidak
^	XOR	Nilai 1 jika dan hanya jika salah satu nilai adalah 1 dan 0 jika tidak
~	Compliment	Nilai 1 menjadi 0 dan sebaliknya
<<	Left Shift	Menggeser nilai 1 kekiri
>>	Right Shift	Menggeser nilai 1 kekanan
<<<	Zero Fill Left Shift	Menggeser nilai 1 kekiri dan menjadikan nilai paling kiri 0
>>>	Zero Fill Right Shift	Menggeser nilai 1 kekanan dan menjadikan nilai paling kekanan 0

Table 6: Operator Bitwise

```

boolean utsIsPassed = true;
boolean uasIsPassed = false;
boolean isPassed = utsIsPassed && uasIsPassed;
int a = 7; // 111
int b = 5; // 101
int c = a & b; // 101
int d = a ^ b; // 010

```

Drawing 9: Operasi Logical dan Bitwise