STRUKTUR DATA

Menggunakan JAVA

Agung BP

Buku Ajar & Latihan

Buku Ajar dan Latihan

STRUKTUR DATA

Agung Budi Prasetyo

2017

Kata Pengantar

Sudah dapat dipastikan bahwa sebagai calon informatikawan yang akan bergelut di dunia IT mahasiswa pasti akan dituntut banyak hal, tidak hanya dalam penguasaan hardware namun juga software, tidak hanya dalam pengoperasian program-program aplikasi mutakhir namun juga penguasaannya terhadap konsep pemrograman yang baik dan benar. Untuk menguasai itu semua salah satu bentuk pembelajaran yang harus ditempuh adalah dengan mempelajari konsep dan praktik.

Matakuliah Struktur Data adalah matakuliah yang akan membantu mahasiswa memahami konsep-konsep pengelolaan data secara algoritmis. Mahasiswa akan belajar segala hal yang mendasar yang berkaitan tentang data, mulai dari pengenalan data, struktur penyimpanan data, pengelolaan data di dalam memori, hingga pemecahan permasalahan tentang data, dan buku ini diciptakan untuk membantu mahasiswa mempelajarinya.

Dengan mengikuti materi dalam buku ini mahasiswa diharapkan mampu membuat penyelesaian kasus-kasus pengelolaan data dalam sebuah program berbahasa Java.

Buku ini masih akan terus disempurnakan seiring dengan berjalannya proses pembelajaran hingga mencapai kondisi ideal. Untuk itu kritik bagi perbaikan buku ini silakan kirimkan melalui email ***agung.b.prasetyo@gmail.com*.**

Akhirnya penulis mengucapkan selamat belajar, selamat berproses, dan jadilah yang terbaik.

Salam.

Penulis.

Agung Budi Prasetyo.

Daftar Isi

BAB 1
TIPE-TIPE DATA DALAM JAVA

Tujuan

Mahasiswa dapat menggunakan tipe data sesuai dengan kebutuhan dan dapat mengimplementaskannya dengan bahasa pemrograman Java.

1.1. Tipe-tipe Data dalam Java

Setiap bahasa pemrograman memiliki tipe data yang spesifik. Tipe data akan digunakan untuk mendeklarasikan variable yang digunakan. Tipe data digunakan untuk menentukan bentuk data yag dapat ditampung oleh sebuah variabel .

Dalam java terdapat dua jenis tipe data. Yang pertama adalah tipe data *primitive* yang merupakan tipe data bawaan dari compiler java. Tipe data ini akan anda pelajari pada bab 1 ini.

Sedangkan tipe data yang kedua adalah tipe data *buatan* yang baru akan anda pelajari pada bab 2.

Dalam bahasa Java, *tipe data primitive* dibedakan menjadi tiga bagian yaitu :

1. **Tipe Data Alphabetic**
    - Char
    - String
2. **Tipe Data Alphanumeric**
    a. Tipe data Bilangan Bulat
        - Byte
        - Short
        - Int
        - Long
    b. Tipe Data Bilangan Pecahan
        - Float
        - Double
3. **Tipe Data Boolean**

1.2. Tipe Data Alphabetic

1.2.1. Char (Karakter)

Tipe data char merupakan tipe yang digunakan untuk menyatakan sebuah karakter, bisa berupa huruf/ tandabaca/ simbol, yang didefinisikan dengan diawali dan diakhiri dengan tanda ‘ (petik tunggal).

Untuk merepresentasikan semua karakter yang ada bahasa Java menggunakan karakter *Unicode* yaitu sekumpulan karakter umum yang terdapat pada semua bahasa, seperti English, Latin, Arab, Yunani dan lain-lainnya.Karakter *Unicode* yang membutuhkan ukuran 16-bit dan memiliki 1680 jenis karakter.

Berikut ini disajikan contoh program sederhana menggunakan tipe data char.

```
public class tipeData {
	public static void main(String[] args) {
		char   data1 = 'C';
		System.out.println("Nilai Char    : "+ data1);
	}
}
Hasil Eksekusi :
	Nilai Char    : C
	Press any key to continue . . .
```

Program 1.1 Contoh pemakaian `char`

Dalam Java, data bertipe char juga dapat diinputkan melalui keyboard untuk disimpan dalam sebuah variabel char. Namun java tidak menyediakan fungsi khusus untuk membaca masukan bertipe char sehingga perlu dibuat pembacaan karakter menggunakan pembacaan kode unicode menggunakan fungsi `System.in.read()` dan kemudian mengkonversinya menggunakan fungsi `(char)` untuk dapat membacanya. Berikut ini adalah program untuk membaca masukannya.

```
import java.util.Scanner;
public class inputViaKeyboard
{ public static void main(String[] args)
  {
	Scanner masukan = new Scanner(System.in);
	int   bacaTombol=0;
	char  huruf;
	System.out.print("Silakan masukkan sebuah huruf: ");
	try
	{   bacaTombol = System.in.read();	}
	catch(java.io.IOException e)
	{	}
	huruf = (char)bacaTombol;
	System.out.println("Huruf yang anda entri adalah : " + huruf);
  }
}


Hasil Eksekusi :
	Silakan masukkan sebuah huruf: Z ↵

	Huruf yang anda entri adalah : Z
	Press any key to continue . . .
```

Program 1.2 Contoh pemakaian `char` Via Keyboard

1.2.2. String

Tipe data String merupakan kumpulan dari tipe data char. Karena merupakan kumpulan char, maka tipe data String dapat digunakan untuk menyimpan kalimat.

Jika dilihat dari unsur pembentuknya, tipe data string bukan merupakan *tipe data primitif*, tetapi sudah merupakan sebuah *objek* yang berisi kumpulan tipe data char.

Berikut ini disajikan contoh program sederhana menggunakan tipe data string.

```
public class tipeData {
     public static void main(String[] args) {
          String data2 = "Namaku Agung Budi Prasetyo";
          System.out.println("Nilai String  : "+ data2);
     }
}
Hasil Eksekusi :
       Nilai String  : Namaku Agung Budi Prasetyo
       Press any key to continue . . .
```

Program 1.3 Contoh pemakaian `String`

Dalam bahasa java, data bertipe string juga dapat diinputkan melalui keyboard untuk disimpan dalam sebuah variabel string. Untuk keperluan tersebut java telah menyediakan sebuah fungsi untuk membaca masukan yaitu `next()` dan `nextline()`. Namun dari kedua fungsi di atas penulis lebih menyarankan untuk menggunakan fungsi `next()`.

Berikut ini adalah program untuk membaca masukannya.

```
import java.util.Scanner;
public class inputViaKeyboard
{
  public static void main(String[] args)
  {
       Scanner masukan = new Scanner(System.in);
       String kalimat;
       System.out.print("Silakan masukkan sebuah kalimat : ");
       kalimat = masukan.next();
       System.out.println("Kalimat yang anda entri adalah :" + kalimat);
  }
}

Hasil Eksekusi :
       Silakan masukkan sebuah kalimat: Namaku Agung Budi Prasetyo ↵

       Kalimat yang anda entri adalah : Namaku Agung Budi Prasetyo
       Press any key to continue . . .
```

Program 1.4 Contoh pemakaian `String` via keyboard

Untuk karakter-karakter yang tidak dapat diketikkan secara langsung melalui keyboard, java menyediakan beberapa *escape sequence* (pasangan karakter yang

dianggap sebagai karakter tunggal). *Escape sequence* tidak dianggap sebagai *String*, melainkan tetap sebagai tipe karakter khusus. Berikut ini adalah beberapa contoh escape sequence.

Kode	Keterangan
\b	Backspace
\t	Tab
\n	Linefeed
\r	Carriage return
\f	Formfeed
\’	Petik tunggal
\”	Petik ganda
\ddd	Octal (dd= 0 s/d 377)
\xdd	Heksadesimal (dd=0 s/d FF atau ff)

Contoh penggunaan escape sequence

```
public class tipeData {
     public static void main(String[] args) {
          String kalimat = "Halo \"Agung Budi Prasetyo\" apa kabar.." ;
          System.out.println("Nilai String  : "+ kalimat);
     }
}
Hasil Eksekusi :
       Nilai String  : Halo "Agung Budi Prasetyo" apa kabar..
       Press any key to continue . . .
```

Program 1.5 Contoh pemakaian *escape sequence* petik (“)

1.3. Tipe Data Alphanumeric

1.3.1. Tipe Data Bilangan Bulat

Ada empat macam tipe bilangan bulat, dimana masing-masing memiliki jangkauan nilai yang berbeda yaitu byte, short, int dan long.

Tipe	Ukuran	Jangkauan Nilai
byte	8 bit	-128 s/d 127
short	16 bit	-32.768 s/d 32.767
int	32 bit	-2.147.483.648 s/d 2.147.483.647
long	64 bit	-9.223.372.036.854.775.808 s/d 9.223.372.036.854.775.807

1.3.1.1. Byte

Type data *byte* umumnya digunakan pada saat kita bekerja dengan sebuah data *stream* dari suatu file, memory, maupun jaringan koputer yaitu untuk kepeluan proses membaca/menulis berkas. Selain itu, tipe ini juga digunakan saat bekerja dengan data biner. Namun tidak jarang pula tipe byte digunakan untuk menyimpan bilangan yang tidak terlalu besar (bilangan di bawah bilangan 127, contohnya umur pegawai).

1.3.1.2. Short

Tipa data short memiliki ukuran yang sedikit lebih besar dibandingkan byte. Pada umumnya tipe data short digunakan pada komputer-komputer 16-bit yang memang memiliki ukuran bilangan yang terbatas. Tipe short digunakan pula pada aplikasi khusus yang memperhatikan penggunaan memori

1.3.1.3. Int

Tipe data int merupakan tipe yang paling banyak dipakai dalam merepresentasikan angka dalam Java. Hal ini karena tipe data int dianggap paling efisien dibandingkan dengan tipe-tipe integer lainnya. Tipe *Int* banyak digunakan untuk indeks dalam struktur pengulangan maupun dalam konstruksi sebuah *array.*Selain itu, secara teori setiap ekspresi yang melibatkan tipe integer *byte, short, int, long* semuanya itu akan dipromosikan ke *int* terlebih dahulu sebelum dilakukan proses perhitungan

1.3.1.4. Long

Tipe ini digunakan untuk kasus-kasus tertentu yang nilainya berada di luar rentang tipe int, karna tipe ini punya range paling tinggi dibanding Integer lainnya. Dengan kata lain, tipe long terpaksa digunakan jika data memiliki range diluar range int.

Semua bilangan bulat dalam Java secara default dianggap sebagai tipe `int`. Sedangkan bilangan yang ingin dikategorikan sebagai long harus diakhiri dengan huruf `L`. Misalnya : `18102006L`. Berikut ini disajikan contoh program sederhana menggunakan tipe data byte, short, int, dan long.

```
public class tipeData {
  public static void main(String[] args) {
         // Tipe data primitif
         byte   data3 = 34;
         short  data4 = 714;
         int    data5 = 2235641;
         long   data6 = 546767226531L;
         System.out.println("Nilai Byte    : "+ data3);
         System.out.println("Nilai Short   : "+ data4);
         System.out.println("Nilai Int     : "+ data5);
         System.out.println("Nilai Long    : "+ data6);
  }
}
Hasil Eksekusi :
  Nilai Byte    : 34
  Nilai Short   : 714
  Nilai Int     : 2235641
  Nilai Long    : 546767226531
  Press any key to continue . . .
```

Program 1.6 Contoh pemakaian `byte`, `short`, `int` dan `long`

Dalam bahasa java, data bertipe byte, short, int dan long juga dapat diinputkan melalui keyboard untuk disimpan dalam sebuah variabel dengan tipe yang sama. Untuk keperluan tersebut java telah menyediakan sebuah fungsi untuk membaca masukan yaitu

`nextByte()` untuk tipe data byte, `nextShort()` untuk tipe data short, `nextInt()` untuk tipe data int, dan `nextLong()` untuk tipe data long.

Berikut ini adalah program untuk membaca masukannya.

```
import java.util.Scanner;
public class inputViaKeyboard
{
  public static void main(String[] args)
  {
    Scanner masukan = new Scanner(System.in);
    byte   bilanganByte;
    short  bilanganShort;
    int    bilanganInt;
    long   bilanganLong;

    System.out.print("Silakan masukkan bilangan bertipe byte: ");
    bilanganByte = masukan.nextByte();
    System.out.print("Silakan masukkan bilangan bertipe short: ");
    bilanganShort = masukan.nextShort();
    System.out.print("Silakan masukkan bilangan bertipe int: ");
    bilanganInt = masukan.nextInt();
    System.out.print("Silakan masukkan bilangan bertipe long: ");
    bilanganLong = masukan.nextLong();

    System.out.println("bilangan byte  yang anda entri = " + bilanganByte);
    System.out.println("bilangan short yang anda entri = " + bilanganShort);
    System.out.println("bilangan int   yang anda entri = " + bilanganInt);
    System.out.println("bilangan long  yang anda entri = " + bilanganLong);
  }
}
Hasil Eksekusi :
    Silakan masukkan bilangan bertipe byte: 34 ↵
    Silakan masukkan bilangan bertipe short: 714 ↵
    Silakan masukkan bilangan bertipe int: 2235641 ↵
    Silakan masukkan bilangan bertipe long: 546767226531 ↵

    bilangan byte  yang anda entri adalah = 34
    bilangan short yang anda entri adalah = 714
    bilangan int   yang anda entri adalah = 2235641
    bilangan long  yang anda entri adalah = 546767226531
Press any key to continue . . .
```

Program 1.7 Contoh pemakaian `float` dan `double`

1.3.2. Tipe Data Bilangan Pecahan

1.3.2.1. Float dan Double

Ada dua tipe data yang berkaitan dengan bilangan pecahan yang sering juga diistilahkan dengan sebutan bilangan titik mengambang.

Tipe	Ukuran	Jangkauan Nilai
`float`	32 bit, presisi 6-7 digit	-3.4E38 s/d +3.4E38
`double`	64 bit, presisis 14-15 bit	-1.7E308 s/d +1.7E308

Semua bilangan pecahan atau desimal dalam Java tanpa diakhiri huruf **f** akan dianggap sebagai *double*. Sedangkan bilangan yang ingin dikategorikan sebagai *float* harus diakhiri dengan huruf **F**. Misalnya : `4.22F` atau `2.314f`.

```
public class tipeData {
   public static void main(String[] args) {
        // Tipe data primitif
        float  data7 = 1.733F; // tipe data pecahan
        double data8 = 4.967; // tipe data pecahan
        System.out.println("Nilai Float   : "+ data5);
        System.out.println("Nilai Double  : "+ data6);
   }
}
Hasil Eksekusi :
       Nilai Float   : 4.967
       Nilai Double  : 1.733
       Press any key to continue . . .
```

Program 1.8 Contoh pemakaian `float` dan `double`

1.4. Tipe Data Boolean

Dalam Java dikenal tipe data boolean yang terdiri dari dua nilai saja, yaitu `true` dan `false`. Boolean sangat penting dalam mengevaluasi suatu kondisi, dan sering digunakan untuk menentukan alur program.

```
public class tipeData {
     public static void main(String[] args) {
          // Tipe data primitif
          boolean data9 = true;
          boolean data10 = false;
          System.out.println("Nilai data9  : "+ data9);
          System.out.println("Nilai data10 : "+ data10);
     }
}
Hasil Eksekusi :
  Nilai data9  : true
  Nilai data10 : false
  Press any key to continue . . .
```

Program 1.9 Contoh pemakaian `boolean`

1.5. Tipe Data Array/ Larik

Selain ketiga tipe data di atas, dalam java terdapat juga tipe data Array/ Larik. Tipe data Array adalah tipe data untuk membuat variabel secara bersusun.

Dengan adanya tipe data array/ larik kita dapat membuat variabel-wariabel kembar yang bertipe data sama. Sebagai contoh dapat kita lihat program berikut ini.

```
import java.util.Scanner;
public class tipeDataArray
{
  public static void main(String[] args)
  {
       String hobi[] = new String[3];
```

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

        System.out.println("Silakan masukkan hobi (maks 3) : ");
        System.out.print("hobi ke-0 : "); hobi[0] = masukan.next();
        System.out.print("hobi ke-1 : "); hobi[1] = masukan.next();
        System.out.print("hobi ke-2 : "); hobi[2] = masukan.next();

        System.out.println("Hobi ke-0 anda adalah " + hobi[0]);
        System.out.println("Hobi ke-1 anda adalah " + hobi[1]);
        System.out.println("Hobi ke-2 anda adalah " + hobi[2]);
    }
}
```

Hasil Eksekusi :

```
    Silakan masukkan hobi (maks 3) :
    hobi ke-0 : musik ↵
    hobi ke-1 : mancing ↵
    hobi ke-2 : touring ↵

    Hobi ke-0 anda adalah musik
    Hobi ke-1 anda adalah mancing
    Hobi ke-2 anda adalah touring
Press any key to continue . . .
```

Program 1.10 Contoh pemakaian Array/Larik

Pada contoh di atas terdapat tiga buah variabel hobi. Sebagaimana kita ketahui sangat jarang seseorang yang hanya memiliki satu hobi saja. Kebanyakan orang memiliki lebih dari satu hobi. Jika kita menggunakan variabel tunggal untuk menyimpan data hobi maka kita tidak dapat menyimpan hobi yang lebih dari satu.

Di sisi lain jika kita menggunakan variabel dengan penamaan yang berbeda untuk hobi—hobi yang ada (misal: `hobi_1`, `hobi_2`, `hobi_3`, dst) maka kita akan kesulitan sendiri pada saat kita akan mengelola data-data tersebut. Oleh karena itu tipe data array/ larik akan menjadi solusi yang tepat untuk keperluan tersebut.

1.6. Latihan

Latihan 1 : *(percobaan tentang tipe data integer)*

Tulislah dan eksekusilah program dibawah ini.

```
public class pembagian{
     public static void main(String[] args){
          int banyaknyaApel = 5;
          int jumlahAnak = 2;
          int perolehan;
          perolehan = banyaknyaApel / jumlahAnak;
          System.out.println("Masing2 mendapat = " + perolehan);
     }
}
```

Program di atas digunakan untuk menghitung 5 dibagi 2 yang menghasilkan nilai 2.5. Sekarang eksekusilah program di atas. Berapa hasil yang diperoleh sewaktu program tersebut dieksekusi? Apakah hasilnya `2.5`? Mengapa bisa demikian?

Latihan 2 : *(percobaan tentang tipe data Long)*

Tulislah dan eksekusilah program dibawah ini.

```
public class cobaLong{
	public static void main(String[] args){
		long coba = 1234567890123;
		System.out.println(coba);
	}
}
```

Apa yang terjadi sewaktu program di atas dieksekusi? Mengapa bisa demikian? Sekarang tambahkan "`L`" pada akhir angka pada baris 3. Apa yang terjadi? Mengapa bisa demikian?

Latihan 3 : *(percobaan tentang tipe data String dan Char)*

Tulislah dan eksekusilah program dibawah ini.

```
public class cobaKalimat{
	public static void main(String[] args){
		char coba="HAI";
		System.out.println(coba);
	}
}
```

Apa yang terjadi sewaktu program di atas dieksekusi? Mengapa bisa demikian? Sekarang gantilah `char` pada baris 3 dengan `String`. Apa yang terjadi? Mengapa bisa demikian?

Latihan 4 : *(percobaan tentang lingkup variabel)*

Tulislah dan eksekusilah program dibawah ini.

```
1. public class Variabel {
2.   static int a;
3.   public static void main(String[] args) {
4.        int x;
5.        x = 10;
6.        a = 2;
7.        System.out.println("Nilai a : " + a);
8.        {    int y;
9.             y = 5;
10.            System.out.println("Nilai x : " + x);
11.            System.out.println("Nilai a : " + a);
12.            {      int z;
13.                   z = 20;
14.                   System.out.println("Nilai x + y + z + a :"
```

```
15.                                 + (x + y + z + a));
16.             }
17.             System.out.println("Nilai Z : " + Z);
18.             System.out.println("Nilai y : " + y);
19.         }
20.         System.out.println("Nilai Z : " + Z);
21.         System.out.println("Nilai y : " + y);
22.         System.out.println("Nilai x : " + x);
23.   }
24.}
```

Apakah yang terjadi sewaktu program tersebut dieksekusi? Mengapa bisa demikian?

Coba sekarang hapuslah instruksi pada baris 17, 20 dan 21 kemudian eksekusi kembali program tersebut. Apa yang terjadi? Mengapa bisa demikian?

Latihan 5 : *(percobaan tentang menerima masukan keyboard)*

Tulislah dan eksekusilah program dibawah ini.

```
import java.util.Scanner;
public class inputDataViaKeyboard
{
  public static void main(String[] args)
  {    String nama;
       String alamat;
       int    umur;
       char   jekel; //jenis kelamin
       String hobi[] = new String[3];
       float  ipk;

       Scanner masukan = new Scanner(System.in);
       int bacaTombol=0;

       System.out.print("Silakan masukkan nama anda : ");
       nama = masukan.next();

       System.out.print("Silakan masukkan alamat anda : ");
       alamat = masukan.next();

       System.out.print("Silakan masukkan umur anda : ");
       umur = masukan.nextInt();

       System.out.print("Silakan masukkan Jenis Kelamin anda : ");
       try
       {   bacaTombol = System.in.read();
       }
       catch(java.io.IOException e)
       {
       }
       jekel = (char)bacaTombol;

       System.out.println("Silakan masukkan hobi (maks 3) : ");
       System.out.print("hobi ke-0 : ");hobi[0] = masukan.next();
       System.out.print("hobi ke-1 : ");hobi[1] = masukan.next();
       System.out.print("hobi ke-2 : ");hobi[2] = masukan.next();

       System.out.print("Silakan masukkan IPK anda : ");
```

```
        ipk = masukan.nextFloat();

        System.out.println("Nama anda adalah " + nama);
        System.out.println("Nama alamat adalah " + alamat);
        System.out.println("Umur anda adalah " + umur);
        System.out.println("Jenis Kelamin anda adalah " + jekel);
        System.out.println("Hobi ke-0 anda adalah " + hobi[0]);
        System.out.println("Hobi ke-1 anda adalah " + hobi[1]);
        System.out.println("Hobi ke-2 anda adalah " + hobi[2]);
        System.out.println("IPK anda adalah " + ipk);
    }
}
```

Apakah yang terjadi jika program di atas di *run* ?