# Tipe Data Bentukan

## Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Mengenal tipedatabentukan (objek)
2. Membedakan antara tipe data primitif dan reference/ bentukan
3. Mengetahui String dan sifat-sifatnya
4. Mengetahui perintah-perintah pada String

## Alat & Bahan

Alat & Bahan Yang digunakan adalah hardware perangkat PC beserta Kelengkapannya berjumlah 40 PC serta Software IntelliJ IDEA yang telah terinstall pada masing-masing PC

## Dasar Teori

Pada Java, tipe data bentukan disebut sebagai tipe data reference/ objek. Berbeda dengan bahasa C/ C++, Java tidak mengenal struct, semua objek merupakan bentukan dari suatu class. Karena itu, untuk tipe data selain tipe data primitif yang telah dipelajari pada pertemuan dua, merupakan objek yang berasal dari kelas tertentu (yang juga dapat berasal dari kelas yang dibentuk oleh programmer).

### String

String merupakan bentuk data yang sering digunakan dalam bahasa pemrograman untuk mengolah data teks atau kalimat. Jika pada bahasa C, string merupakan kumpulan karakter pada array (array of characters), maka pada bahasa Java, string merupakan suatu objek yang berasal dari class String. Deklarasi dan inisialisasi variabel dengan tipe String dapat dilakukan dengan beberapa cara, seperti berikut:

|  |
| --- |
| String nama = “Nina”;  String nama2 = new String (“Nina”);  char[] nama3 = new char[] {‘N’, ‘i’, ‘n’, ‘a’};  String nama4 = new String(nama3); |

Walaupun menyimpan data yang sama (Nina), variabel nama, nama2 dan nama4 merupakan objek yang berbeda. Jadi jika dilakukan operasi persamaan (‘==’) maka hasilnya akan ‘false’. String dapat menerima nilai dari suatu array karakter, seperti terlihat pada variabel nama4. Keyword new akan membuat sebuah objek String baru, sehingga dua objek String yang memiliki nilai sama tidak akan dianggap sama oleh Java. Perhatikan potongan program berikut.

|  |  |
| --- | --- |
| package praktikum8\_1;  class StringCompare {  public static void main (String args[]){  String nama = “Lilo”;  String nama2 = “Lilo”;  String nama3 = new String(“Lilo”);  String nama4 = new String(“Lilo”);  System.out.println(nama == nama2);  System.out.println(nama == nama3);  System.out.println(nama3 == nama4);  }  } | Apakah Outputnya?  ………………………… |

Program ‎8‑1 Perbandingan pada String

Dalam program 8-1, hasil dari nama == nama2 adalah true, karena kedua variabel merujuk pada objek yang sama (“Lilo”), namun tidak demikian halnya dengan dua perbandingan lainnya. Variabel nama3 dan nama4, walaupun memiliki nilai yang sama (“Lilo”) merupakan dua objek yang berbeda, karenanya hasil dari perbandingan adalah false. Untuk membandingkan isi dari variabel String, gunakan fungsi equals yang akan dibahas kemudian.

String termasuk tipe data yang paling banyak digunakan, dan kelas String memiliki method yang dapat digunakan untuk melakukan manipulasi String, namun tidak mengubahnya, seperti charAt, indexOf, substring, trim, replace dan length.

Tabel ‎8‑1 Method pada String

|  |  |
| --- | --- |
| Sintaks | Deskripsi |
| charAt(int index) | Mengambil satu karakter dari string sesuai indeks |
| indexOf() | Membandingkan apakah karakter atau string tertentu ada pada suatu string. Jika ada, akan mengembalikan indeks pertama yang ditemukan, jika tidak akan mengembalikan nilai -1. |
| substring() | Membuat sub-string dari suatu string. |
| trim() | Menghilangkan white space (baris baru, spasi atau tab) sebelum atau sesudah suatu string. White space di antara string tidak dihilangkan. |
| replace() | Mengembalikan string baru dengan mengganti suatu karakter (atau string) dengan karakter (atau string) lainnya. |
| length() | Mengembalikan nilai panjang suatu string (banyaknya huruf pada string tersebut) |

Seperti pada array, indeks suatu string dimulai dari 0. Yang harus diperhatikan dari operasi manipulasi string adalah, walaupun sekilas terlihat method yang ada dapat mengubah suatu string, pada kenyataannya string tersebut tidak berubah, tetap seperti sebelum dilakukan operasi pada string. Hal ini dikarenakan, dalam Java, String bersifat tidak dapat diubah (*immutable*), nilai suatu string akan tetap seperti pada saat inisialisasi. Untuk lebih jelasnya perhatikan contoh program berikut.

|  |  |
| --- | --- |
| package praktikum8\_2;  class StringManipulation {  public static void main (String args[]){  String nama = “Telkom University ”;  System.out.println(nama.charAt(2));  System.out.println(nama.indexOf(‘k’));  System.out.println(nama.indexOf(“Telkom”));  System.out.println(nama.indexOf(“koma”));  String sub1 = nama.substring(3);  String sub2 = nama.substring(7,10);  System.out.println(nama.length());  System.out.println(nama.trim());  System.out.println(nama.length());  System.out.println(nama.replace(‘r’, ‘R’));  System.out.println(nama.repplace(“ty”,”tas”));  System.out.println(nama);  }  } | Apakah Outputnya?  ………………………… |

Program ‎8‑2 Manipulasi String

Perintah indexOf dapat digunakan untuk mencari baik karakter maupun kata (string). Nilai yang dimunculkan adalah indeks dari karakter pertama yang ditemukan. Perintah substring dapat digunakan dengan menuliskan indeks dari karakter pertama yang akan diambil, atau dari karakter pertama dan karakter akhir. Pada Program 8-2, perintah substring(3) akan membuat substring baru mulai dari huruf ‘k’ sampai akhir string, sementara perintah substring(7,10) akan mengambil huruf dari indeks 7-9 (indeks ke-10 tidak diambil).

Perintah length akan mengembalikan panjang suatu string, *whitespace* dan tanda baca dihitung. Perintah trim akan menghapus *whitespace* sebelum atau sesudah string, namun tidak menghapus *whitespace* diantara string. Replace dapat mengganti karakter maupun string. Namun, semua manipulasi yang dilakukan tidak akan mengubah string awal, jadi isi dari variabel nama tetap “Telkom University”.

#### Penggabungan String

Dua string atau lebih dapat digabungkan dengan operator penggabungan (*concatenation*) + atau +=. Untuk lebih jelasnya dapat dilihat pada Program 8-3.

|  |  |
| --- | --- |
| package praktikum8\_3;  import java.util.Scanner;  class StringConcatenation {  public static void main (String args[]){  String gabung = “Telkom” + “” + “University”;  System.out.println(gabung);  gabung += “ Bandung”;  System.out.println(gabung);  Scanner in = new Scanner(System.in);  System.out.println(“Masukkan NIM Anda:”);  int NIM = in.nextInt();  System.out.println(“Masukkan Kelas Anda (1-4):”);  int kelas = in.nextInt();  System.out.println(“Masukkan nama Anda:”);  String nama = in.next();  System.out.println(“NIM, kelas dan nama Anda:”);  System.out.println(NIM + kelas + nama);  System.out.println(“Bandingkan dengan:”);  System.out.println(“”+ NIM + kelas + nama);  }  } | Apakah Outputnya?  ………………………… |

Program ‎8‑3 Penggabungan String

Operator + dan += dapat menggabungkan baik String maupun integer. Namun, harus diperhatikan, jika menggabungkan literal string dengan integer, maka nilai integer akan diperlakukan sebagai suatu integer (jadi akan dilakukan operasi terhadap nilai integer tersebut, dalam hal ini penjumlahan). Untuk mencegah hal tersebut, awali proses dengan suatu string (dalam Program 8-3 diwakili dengan “”).

#### Perbandingan String

Sebuah string dapat dibandingkan dengan string lainnya untuk melihat apakah kedua string tersebut sama. Perbandingan string dilakukan dengan menggunakan perintah equals. Sebaiknya tidak menggunakan operator “==” dalam perbandingan string, karena dapat menimbulkan hasil yang tidak sesuai. Untuk lebih jelasnya, perhatikan Program 8-4.

|  |  |
| --- | --- |
| package praktikum8\_4;  class StringCompare {  public static void main (String args[]){  String banding = “Telkom”;  String banding2 = “Telkom”;  System.out.println(banding == banding2);  System.out.println(banding.equals(banding2));  String banding3 = new String(“Telkom”);  String banding4 = new String(“Telkom”);  System.out.println(banding3 == banding4);  System.out.println(banding3.equals(banding4));  System.out.println(banding == banding4);  System.out.println(banding.equals(banding3));  }  } | Apakah Outputnya?  ………………………… |

Program ‎8‑4 Perbandingan String

Hasil perbandingan antara variabel banding dan banding2 adalah true baik untuk operator “==” maupun perintah equals karena kedua variabel merujuk pada objek yang sama di Java. Sebaliknya, keyword new akan selalu membuat objek baru. Hasil dari perbandingan banding3 == banding4 adalah false karena kedua variabel merujuk pada objek yang berbeda (operator “==” membandingkan apakah kedua objek sama). Namun ketika digunakan equals, hasilnya akan menjadi true karena equals membandingkan isi string.

### StringBuilder

Karena String bersifat tidak dapat diubah, bagaimana jika suatu ketika isi string tersebut harus diubah? Java menyediakan kelas StringBuilder untuk kebutuhan tersebut. Suatu StringBuilder dapat dideklarasikan dalam beberapa cara, seperti terlihat pada potongan program berikut.

|  |
| --- |
| StringBuilder nama = new StringBuilder();  StringBuilder nama2 = new StringBuilder(nama);  StringBuilder nama3 = new StringBuilder(30);  StringBuilder nama4 = new StringBuilder(“Crystal Clear”); |

Variabel nama membuat objek StringBuilder baru dengan kapasitas awal 16 karakter. Variabel nama2 menerima variabel nama sebagai nilai awalnya. Sementara itu, variabel nama3 memiliki kapasitas awal 30, sesuai dengan masukan nilai awalnya. Kemudian, suatu StringBuilder juga dapat dibuat dengan langsung memasukkan string yang diinginkan (dalam potongan program diatas adalah Crystal Clear). Pada dasarnya, semua tipe data (termasuk objek) dapat dijadikan nilai awal bagi StringBuilder.

Seperti halnya String, StringBuilder juga memiliki perintah-perintah yang dapat digunakan untuk manipulasi isinya. Namun, karena sifat StringBuilder adalah dapat berubah (mutable), maka perubahan yang dilakukan akan mengubah isi dari StringBuilder tersebut. Tabel 8-2 menampilkan beberapa perintah pada StringBuilder.

Tabel ‎8‑2 Method pada StringBuilder

|  |  |
| --- | --- |
| Sintaks | Deskripsi |
| charAt(int index) | Mengambil satu karakter dari string sesuai indeks |
| indexOf() | Membandingkan apakah karakter atau string tertentu ada pada suatu string. Jika ada, akan mengembalikan indeks pertama yang ditemukan, jika tidak akan mengembalikan nilai -1. |
| substring() | Membuat sub-string dari suatu string. |
| append() | Menambahkan nilai diujung string. Nilai yang ditambahkan dapat berasal dari seluruh tipe data. |
| insert() | Memiliki fungsi sama seperti append, tapi nilai dapat ditambahkan pada posisi manapun. |
| delete() | Menghapus karakter pada jangkauan tertentu pada suatu string. |
| deleteCharAt() | Menghapus karakter pada satu posisi tertentu pada suatu string. |
| length() | Mengembalikan nilai panjang suatu string (banyaknya huruf pada string tersebut) |

Perintah charAt, indexOf, substring dan length pada StringBuilder bersifat sama persis dengan pada String, karena itu, contoh program berikut akan membahas perintah append, insert, delete dan deleteCharAt.

|  |  |
| --- | --- |
| package praktikum8\_5;  class StringBuilderDemo {  public static void main (String args[]){  StringBuildersb = new StringBuilder(“Tkm”);  char tambah[] = {‘T’,’e’,’l’,’k’,’o’,’m’};  sb.insert(1,tambah,1,2);  System.out.println(sb);  sb.insert(4,’o’);  System.out.println(sb);  sb.delete(1,3);  System.out.println(sb);  sb.deleteCharAt(3);  System.out.println(sb);  sb.append(10.5);  System.out.println(sb);  sb.append(‘r’);  System.out.println(sb);  }  } | Apakah Outputnya?  ………………………… |

Program ‎8‑5 StringBuilder

Pada Program 8-4, perintah insert menyisipkan isi dari indeks 1 – 2 dari array tambah pada variabel StringBuilder sb. Dengan demikian, huruf ‘e’ dan ‘l’ akan menempati indeks 1 dan 2 dari sb. Perintah insert tidak menimpa isi dari StringBuilder, dengan demikian, huruf ‘k’ yang awalnya berada di indeks ke-1 akan tergeser menjadi indeks ke-3.

Sementara itu, perintah delete menghapus karakter pada sb mulai dari indeks pertama sampai kedua (walaupun pada sintaks tertulis 3, namun indeks ketiga tidak ikut terhapus). Perintah deleteCharAt menghapus karakter sesuai indeks yang tertulis (dalam Program 8-4 adalah indeks ketiga). Perintah append akan menggabungkan nilai yang tertulis pada ujung StringBuilder.

### Tipe Data Reference

Seperti telah disebutkan sebelumnya, tipe data reference merupakan tipe data bentukan. Berbeda dengan tipe data primitif yang memiliki hanya satu nilai literal untuk satu variabel, variabel dengan tipe data reference dapat memiliki lebih dari satu nilai literal dengan tipe yang berbeda.

Pembahasan lebih jauh mengenai objek, kelas dan konsep pemrograman yang berhubungan dengan keduanya akan didapatkan pada mata kuliah Pemrograman Berbasis Objek. Untuk saat ini, kita akan membuat suatu tipe data bentukan dengan objek darisuatu inner class.

Inner class merupakan suatu nested class, kelas yang berada di dalam kelas lain. Kelas yang berada di dalam kelas lain ini memilliki akses – jadi dapat menggunakan – variabel dari outer class (kelas yang tempat inner class bernaung). Namun, pada pembahasan kali ini inner class yang dibuat tidak akan menggunakan variabel dari outer class. Untuk lebih jelasnya, perhatikan Program 8-6 berikut.

|  |  |
| --- | --- |
| package praktikum8\_6;  import java.util.Scanner;  class PanggilMhs {  class Mahasiswa{  String NIM;  String nama;  int IPK;  }  public static void main (String args[]){  Scanner sc= new Scanner(System.in);  Mahasiswa mhs = new Mahasiswa();  System.out.println(“Masukkan NIM mahasiswa:”);  mhs.NIM = sc.next();  System.out.println(“Masukkan nama mahasiswa:”);  mhs.nama = sc.next();  System.out.println(“Masukkan IPK mahasiswa:”);  mhs.IPK = sc.nextInt();  System.out.println(“Data mahasiswa adalah:”);  System.out.println(“NIM: ” + mhs.NIM  + “ Nama: ” + mhs.nama + “ IPK: ” + mhs.IPK);  }  } | Apakah Outputnya?  ………………………… |

Program ‎8‑6 Tipe data bentukan

Kelas Mahasiswa merupakan inner class dari kelas PanggilMhs. Kelas Mahasiswa memiliki tiga variabel dengan tipe data yang berbeda. Untuk saat ini, kelas Mahasiswa dapat dipandang sebagai suatu struktur pembentuk tipe data baru yang terdiri atas tiga unsur: NIM, nama dan IPK. Dengan demikian, tipe data yang terbentuk bukanlah String atau integer (yang merupakan tipe data dari masing-masing unsur di dalam Mahasiswa), namun tipe data baru yang terdiri dari ketiga tipe data masing-masing unsur.

Untuk menggunakan tipe data baru ini, pada kelas PanggilMhs dibuatlah sebuah objek dari Mahasiswa, yaitu mhs. Karena Mahasiswa memiliki tiga unsur data, maka variabel mhs pun memiliki tiga unsur; NIM, nama dan IPK. Untuk mengakses masing-masing unsur, variabel mhs harus memanggil unsur yang dibutuhkan dengan menggunakan notasi titik (.). Jadi, baris mhs.NIM merupakan baris perintah untuk memakai unsur NIM yang ada pada variabel mhs.

Gambar 8-1 mengilustrasikan perbedaan antara tipe data bentukan (variabel mhs) dengan variabel biasa (misal, int data). Variabel mhs terdiri atas unsur-unsur lain dalam satu kesatuan, sementara data hanya dapat menyimpan satu nilai saja.

NIM

nama

IP

String

String

int

int

mhs

data

Gambar ‎8‑1 Tipe data bentukan dan primitif

## Latihan

1. Di suatu perguruan tinggi ada 2 mahasiswa yang berimprovisasi ingin membuat program untuk menulis data diri kedua mahasiswa tersebut serta menghitung Nilai Total yang dia dapat dalam 1 semester dengan spesifikasi:

* Input data Mahasiswa (Nama dan NIM) beserta Nilai UTS, UAS, dan Tugas.
* Rumus Nilai Total = 40% x Nilai UTS + 40% x Nilai UAS + 20% x Nilai Tugas.
* Setelah data nilai total dihitung, kemudian nilai total setiap mahasiswa dioutputkan ke layar.

Buatlah program dengan spesifikasi tersebut menggunakan tipe bentukan.

1. Di suatu perusahaan dibutuhkan aplikasi untuk mencatat data karyawan serta mencatat lamanya seorang karyawan bekerja, buatlah sebuah aplikasi tersebut dengan spesifikasi:

* Input data karyawan, jam masuk, dan jam keluar.
* Lama kerja = jam keluar - jam masuk.
* Kemudian lama kerja di-*output*kan ke layar.

1. Buatlah sebuah program yang menerima input nomor telepon seluler dari *user* dan mengubahnya dalam format +62 XXX-XXXX-XXXX

Contoh:

Inputan *user* : 085634784956

Keluaran : +62 856-3478-4956

Catatan:

Hati-hati untuk beberapa kasus, misal: user tidak memasukkan no telp seluler yang valid (angka kurang atau bukan no telp Indonesia). Untuk kasus seperti itu, tampilkan pesan peringatan.

Contoh:

Inputan *user* : 085634784

Keluaran : Anda tidak memasukkan nomor yang valid

Inputan *user* : 075874784956

Keluaran : Anda tidak memasukkan nomor yang valid