

OPERASI DENGAN STRING

A. String pada Java

String pada Java berbentuk objek. Objek ini dibentuk dengan menggunakan kelas `String`. Sebagai contoh, untuk membentuk objek string bernama `namaBunga`, pernyataan yang diperlukan berupa:

```
String namaBunga;
```

Pada deklarasi tersebut, `namaBunga` tidak berisi string apapun.

Setelah objek string terbentuk, penugasan string ke objek tersebut dapat dilakukan menggunakan bentuk seperti berikut:

```
objekString = suatuString;
```

Contoh berikut mengisikan nama string “Mawar” ke `namaBunga`;

```
namaBunga = "Mawar";
```

B. Inisialisasi String

String dapat diberikan saat objek string dibuat. Bentuk inisialisasi string adalah seperti berikut:

1. `String objekString = new(ekspresiString);`
2. `String objekString = ekspresiString;`

Inisialisasi string juga dapat dilakukan melalui larik karakter.

C. Cacah Karakter dalam String

Untuk mengetahui cacah karakter yang terkandung pada string, menggunakan metode `length()`.

D. Pengaksesan Karakter dalam String

Salah satu cara untuk mengakses karakter dalam suatu objek string adalah menggunakan notasi seperti berikut :

```
larikKarakter[indeks]
```

Indeks 0 menyatakan karakter pertama pada string. Cara ini digunakan dengan cara mengkonversi string ke larikKarakter terlebih dahulu.

`toCharArray()` merupakan metode pada string yang digunakan untuk konversi ke larik. Cara lain untuk mengakses satu karakter pada string adalah menggunakan metode pada string bernama `charAt()`.

E. Operator + untuk Penggabungan String

Operator `+` dapat digunakan untuk menggabungkan dua string. Sebagai contoh, penulisan seperti berikut :

```
String kalimat = "Dian" + " " + "pergi";
```

Pernyataan di atas membentuk objek string bernama `kalimat` dan mengisi "Dian pergi" ke objek tersebut.

```
String teks = "Jumlah =" + 5;
```

Pernyataan seperti ini juga diperbolehkan. Operan pertama berupa string dan operan kedua berupa angka. Dengan sendirinya, angka tersebut akan dikonversikan ke string sehingga hasilnya berupa "Jumlah = 5".

F. Larik String

Variabel larik string dideklarasikan dengan bentuk seperti berikut :

1. `String namaLarik[];`
2. `String[] namaLarik;`

Adapun penentuan *jumlahElemen* dinyatakan dengan bentuk seperti berikut :

```
namaLarik = new String[jumlahElemen];
```

Deklarasi dan inisialisasi jumlah elemen larik string dapat disatukan dengan bentuk seperti berikut :

1. `String namaLarik[] = new String[jumlahElemen]`
2. `String[] namaLarik = new String[jumlahElemen]`

Contoh :

```
String[] namaHewan = new String[3];
```

Maka `namaHewan` dapat menampung hingga 3 string. Sebagai contoh untuk pengisian elemen pertama dapat berupa seperti berikut :

```
namaHewan [0] = "Garuda";
```

Namun inisialisasi pada larik string juga dapat dilakukan seperti contoh :

```
String [] namaHewan = { "Garuda", " Kuda", "Kangguru" };
```

G. Pembandingan String

Pembandingan dua string dilakukan melalui metode bernama `compareTo()`. Penggunaannya seperti berikut :

```
objekString.compareTo(string)
```

Nilai baliknya bertipe `int`, dengan kemungkinan seperti berikut :

- Nilai negatif sekiranya string pada *objekString* bernilai lebih kecil daripada string argumen;
- Nilai 0 sekiranya string pada *objekString* bernilai sama dengan string argumen;
- Nilai positif sekiranya string pada *objekString* bernilai lebih besar daripada string argumen;

Nilai balik yang selain nol menyatakan perbedaan urutan karakter. Sebagai contoh, jika `st` berisi "B", `st.compareTo("A")` menghasilkan 1 dan `st.compareTo("C")` menghasilkan -1.

Selain `compareTo()`, tersedia pula metode bernama `equals()` dan `equalsIgnoreCase()`. Penjelasan kedua metode ini adalah seperti berikut:

- `equals()` berguna untuk membandingkan string objek dengan string argument. Hasilnya berupa nilai `true` sekiranya kedua string sama atau `false` jika tidak sama.
- `equalsIgnoreCase()` berguna untuk membandingkan string objek dengan string argumen tanpa membedakan huruf kapital dan huruf kecil. Hasilnya berupa nilai `true` sekiranya kedua string sama atau `false` jika tidak sama.

H. Pembacaan String dari Papan Ketik

Pembacaan data dari papan ketik dan kemudian diberikan objek string dilakukan dengan menggunakan pernyataan berbentuk seperti berikut :

```
objekKeyboard.nextLine()
```

I. Metode-Metode Pemanipulasi String

Objek berkelas `String` mempunyai sejumlah metode yang dapat digunakan untuk manipulasi string, seperti menambahkan suatu string di akhir string, mencari karakter-karakter tertentu pada string, dan mengganti bagian string dengan suatu string. Beberapa metode penting dapat dilihat pada tabel berikut ini :

Metode	Keterangan
<code>substring(<i>indeks</i>)</code>	Memberikan nilai balik berupa substring pada string objek dimulai dari posisi <i>indeks</i> .
<code>substring(<i>awal,akhir</i>)</code>	Memberikan nilai balik berupa substring pada string objek dimulai dari indeks <i>awal</i> hingga indeks akhir -1.
<code>contains(<i>st</i>)</code>	Mengembalikan <code>true</code> sekiranya deretan karakter <i>st</i> terkandung pada string objek atau <code>false</code> untuk keadaan sebaliknya
<code>join(<i>pembatas, st, ...</i>)</code>	Mengembalikan string yang merupakan penggabungan deretan karakter argument kedua dan seterusnya dengan antar argument akan dipisahkan oleh deretan karakter <i>pembatas</i>
<code>isEmpty()</code>	Mengembalikan <code>true</code> kalau string objek berupa string kosong atau <code>false</code> untuk keadaan sebaliknya
<code>concat(<i>st</i>)</code>	Mengembalikan string yang merupakan penggabungan string objek dengan string argument
<code>replace(<i>karLama, karBaru</i>)</code>	Mengembalikan string berdasarkan string objek dengan semua karakter <i>karLama</i> diganti dengan karakter <i>karBaru</i>
<code>replace(<i>stringLama, stringBaru</i>)</code>	Mengembalikan string berdasarkan string objek dengan semua string <i>stringLama</i> diganti dengan deretan string <i>stringBaru</i>
<code>split(<i>regex</i>)</code>	Mengembalikan larik String yang merupakan hasil pemisahan string objek berdasarkan ekspresi regular <i>regex</i>
<code>indexOf(<i>kar</i>)</code>	Mengembalikan indeks pertama tempat karakter <i>kar</i> ditemukan di string objek
<code>indexOf(<i>kar, awal</i>)</code>	Mengembalikan indeks pertama tempat karakter <i>kar</i> ditemukan di string objek dengan pencarian dimulai dari indeks <i>awal</i>

<code>indexOf(st)</code>	Mengembalikan indeks pertama tempat string <i>st</i> ditemukan di string objek dengan pencarian dimulai dari dari indeks <i>awal</i>
<code>indexOf(st, awal)</code>	Mengembalikan indeks pertama tempat string <i>st</i> ditemukan di string objek dengan pencarian dimulai dari dari indeks <i>awal</i>
<code>toLowerCase()</code>	Mengembalikan string berdasarkan string objek dengan semua huruf kapital dikonversi ke huruf kecil
<code>toUpperCase()</code>	Mengembalikan string berdasarkan string objek dengan semua huruf kecil dikonversi ke huruf kapital
<code>valueOf(bilanganBulat)</code> <code>valueOf(karakter)</code>	Memberikan nilai balik berupa string yang didasarkan pada nilai argument
<code>trim()</code>	Mengembalikan string berdasarkan string objek dengan menghapus spasi-putih di awal dan akhir string objek
<code>format(stringFormat, objek, ...)</code>	Mengembalikan string yang merupakan hasil performatan objek-objek argumen berdasarkan string pemformat <code>stringFormat</code> . String penformat serupa dengan yang digunakan pada <code>System.out.printf()</code> .
<code>endsWith(nilai)</code>	Mengembalikan nilai <code>true</code> kalau string objek berakhiran dengan string <i>nilai</i>
<code>endsWith(nilai, awal)</code>	Mengembalikan <code>true</code> kalau substring pada objek dimulai pada indeks <i>awal</i> berakhiran dengan string <i>nilai</i>
<code>endsWith(nilai, awal, akhir)</code>	Nilai balik berupa <code>true</code> kalau substring pada objek dimulai pada indeks <i>awal</i> hingga indeks <i>akhir</i> berakhiran dengan string <i>nilai</i>
<code>startsWith(nilai)</code>	Mengembalikan nilai <code>true</code> kalau string objek berawalan dengan string <i>nilai</i>
<code>startsWith(nilai, awal)</code>	Mengembalikan <code>true</code> kalau substring pada objek dimulai pada indeks <i>awal</i> berawalan dengan string <i>nilai</i>
<code>startsWith(nilai, awal, akhir)</code>	Nilai balik berupa <code>true</code> kalau substring pada objek dimulai pada indeks <i>awal</i> hingga indeks <i>akhir</i> berawalan dengan string <i>nilai</i>

J. Konversi String ke Bilangan

Untuk mengkonversi string ke bilangan bulat, metode `Integer.valueOf()` atau `Integer.parseInt()` bisa digunakan.

PRAKTIKUM

1. String pada Java

Contoh berikut menunjukkan pembuatan objek string dan pengisian string ke objek, dan sekaligus menampilkan string :

```
//Objek String

import java.util.Scanner;

public class ObjekString {
    public static void main(String[] args) {
        String namaBunga;
        namaBunga = "Mawar";

        System.out.println("namaBunga: " + namaBunga);
    }
}
```

2. Inisialisasi String

```
//Inisialisasi string
import java.util.Scanner;
public class inisialisasiString {
    public static void main(String[] args) {
        String namaHewan = new String ("Kelinci");
        String namaBunga = "Mawar";

        System.out.println("Nama Hewan : " + namaHewan);
        System.out.println("Nama Bunga : " + namaBunga);
    }
}
```

3. Inisialisasi String dengan Larik

```
//inisialisasi string dari larik karakter
import java.util.Scanner;
public class InisialisasiDariLarik {
    public static void main (String[] args) {
        char[] daftarkarakter = {'M','e','r','a','h'};
        String namaWarna = new String(daftarkarakter);
        System.out.println("NamaWarna : " + namaWarna);
    }
}
```

4. Cacah Karakter dengan String

```
// Cara untuk mendapatkan cacah karakter pada string

public class PanjangString {
    public static void main(String[] args) {
        String stringA = "";
        String stringB = "B";
        String stringC = "Dua kata";

        System.out.println ("Jumlah karakter pada stringA: " + stringA.length());
        System.out.println ("Jumlah karakter pada stringB: " + stringB.length());
        System.out.println ("Jumlah karakter pada stringC: " + stringC.length());
    }
}
```

5. Pengaksesan Karakter

Program berikut menunjukkan contoh untuk mendapatkan setiap karakter pada string dan menampilkan pada baris tersendiri :

```
//Pengaksesan karakter dalam string

public class KarakterString {
    public static void main(String[] args) {
        String kata = "Bravo";

        //Konversi string ke larik karakter
        char[] larikkarakter = kata.toCharArray();

        //Pengaksesan karakter
        for (int indeks = 0; indeks < kata.length() ; indeks++)

            System.out.println(larikkarakter[indeks]);
    }
}
```

```
// Pengaksesan karakter dalam string Versi 2

public class KarakterString2 {
    public static void main (String[] args){
        String kata = "Bravo";

        //Pengaksesan karakter
        for (int indeks = 0 ; indeks < kata.length(); indeks++)
            System.out.println(kata.charAt(indeks));
    }
}
```


6. Larik String

```
//Contoh larik String

import java.util.Scanner;
public class LarikString {
    public static void main(String[]args){
        System.out.println("Larik string");
        System.out.println("-----");

        String[] namaHewan = {"Garuda","Kuda","Kanguru","Barakuda","Burung Dara",
            "Unta", "Kedasih","Badak","Harimau","Panda"};

        //Penampilan isi larik
        for (int i =0; i < namaHewan.length; i++)
            System.out.println(namaHewan[i]);
    }
}
```

7. Pembandingan String

```
// Contoh pengurutan data berkelas string
import java.util.Scanner;
public class PengurutanString {
    public static void main(String[]args){
        System.out.println("Pengurutan data nama hewan");
        System.out.println("-----");

        String[] namaHewan = {"Garuda","Kuda","Kanguru","Barakuda","Burung Dara",
            "Unta", "Kedasih","Badak","Harimau","Panda"};

        //Pengurutan data
        int jumlahHewan = namaHewan.length;
        String sementara;

        for (int i=0; i < jumlahHewan-1; i++)
            for(int j = i+1; j < jumlahHewan; j++ )
                if (namaHewan[i].compareTo(namaHewan[j])>0){

                    //Penukaran dua data
                    sementara = namaHewan[i];
                    namaHewan[i] = namaHewan[j];
                    namaHewan[j]= sementara;
                }

        //Penampilan hasil pengurutan
        for (int i=0; i<jumlahHewan;i++)
            System.out.println(namaHewan[i]);
    }
}
```

```
// Contoh penggunaan equals() dan equalsIgnoreCase()
public class PembandinganString {
    public static void main(String[]args){
        String stA = "Bravo";
        String stB = "BRAVO";

        System.out.println("Pembandingan dengan equals():");
        System.out.println(stA + " sama dengan " + stB + " --> " + stA.equals(stB));

        System.out.println();
        System.out.println("Pembandingan dengan equalsIgnoreCase():");
        System.out.println(stA + " sama dengan " + stB + " --> " + stA.equalsIgnoreCase(stB));
    }
}
```

8. Contoh pembacaan data string dari papan ketik

```
// Pemasukan data string dari papan ketik
// dan diberikan ke objek berkelas String
import java.util.Scanner;
public class PembacaanNama {
    public static void main(String[] args) {
        System.out.println("Pemasukan string dari papan ketik");
        System.out.println("-----");

        String nama;

        Scanner kbd = new Scanner(System.in);
        System.out.print("Nama Anda: ");
        nama = kbd.nextLine();

        System.out.println("Hai, " + nama + ". Terima kasih");
    }
}
```

9. Pemeriksaan string kosong dengan metode isEmpty()

```
//Pemeriksaan string kosong

public class StringKosong {
    public static void main(String[] args) {
        System.out.println("Pemeriksaan String Kosong");
        System.out.println("-----");

        String teks = "";
        System.out.println("Kosongkah? " + teks.isEmpty());

        teks = "ABC";
        System.out.println("Kosongkah? " + teks.isEmpty());

        teks = "";
        System.out.println("Kosongkah? " + teks.isEmpty());
    }
}
```

10. Penghapusan karakter pada string dengan replace()

```
//Penghapusan substring dalam string

public class PenghapusanString {
    public static void main(String[] args) {
        String teks = " Tika tidak pergi";

        System.out.println("Isi semula : " + teks);

        teks = teks.replace("tidak ", "");
        System.out.println("Isi sekarang: " + teks);
    }
}
```

11. Pencarian string dengan metode contains()

```
// Contoh pencarian string yang mengandung "da"
// Menggunakan contains()

public class PencarianString {
    public static void main(String[] args) {
        System.out.println("Nama-nama hewan yang mengandung 'da'");
        System.out.println("-----");

        String[] namaHewan = {"Garuda", "Kuda", "Kanguru", "Barakuda", "Burung Dara",
                               "Unta", "Kedasi", "Badak", "Harimau", "Panda"};

        //Seleksi yang mengandung 'da'
        for (int i = 0; i < namaHewan.length; i++)
            if (namaHewan[i].contains("da"))
                System.out.println(namaHewan[i]);
    }
}
```

12. Penggantian string dengan metode replace()

```
// Contoh penggunaan replace ()
// Untuk mengganti bagian string

public class PenggantianString {
    public static void main (String[] args) {
        System.out.println("Metode Replace");
        System.out.println("-----");

        String teks = "ratatata taratara";

        System.out.println("Isi teks semula : " + teks);

        //Penggantian "ra" dengan "di"
        teks = teks.replace("ra", "di");
        System.out.println("ra diganti di : " + teks);
    }
}
```

13. Konversi ke huruf Kapital dan sebaliknya

```
// Contoh konversi string dari huruf kapital
// ke huruf kecil dan sebaliknya

public class KonversiHuruf {
    public static void main(String[] args) {
        System.out.println("Konversi huruf dalam string");
        System.out.println("-----");

        String teks = "Tes...tes...tes...123";
        System.out.println("Keadaan awal : " + teks);

        //Konversi huruf kecil ke huruf kapital
        teks = teks.toUpperCase();
        System.out.println("Konversi ke huruf Besar: " + teks);

        //Konversi huruf kapital ke huruf kecil
        teks = teks.toLowerCase();
        System.out.println("Konversi ke huruf Kecil: " + teks);
    }
}
```

14. Penggabungan string dengan metode join()

```
// Contoh penggabungan string dengan join()

public class PenggabunganString {
    public static void main(String[] args) {
        System.out.println("Penggabungan String");
        System.out.println("-----");

        String kotaA = "Jakarta";
        String kotaB = "Kuala Lumpur";
        String kotaC = "Bangkok";
        String kotaD = "Amsterdam";

        String hasil = kotaA.join("-", kotaB, kotaC, kotaD);
        System.out.println("Hasil join() : " + hasil);
    }
}
```

15. Penggabungan string dengan metode concat()

```
// Contoh Penggabungan string dengan concat()

public class PenggabunganString2 {
    public static void main(String[] args) {
        System.out.println("Penggabungan String");
        System.out.println("-----");

        String kataA = "Bulan";
        String kataB = "Bintang";

        String hasil = kataA.concat(kataB);

        System.out.println("Hasil concat(): " + hasil);
    }
}
```

16. Penguraian string dengan metode split()

```
// Contoh pemisahan string dengan split()

public class PemisahanString {
    public static void main(String[] args) {
        System.out.println("Pemisahan String");
        System.out.println("-----");

        String teks = "Jakarta Bandung Denpasar Ambon";

        String[] hasil = teks.split(" ");

        for (int j = 0; j < hasil.length; j++)
            System.out.println(hasil[j]);
    }
}
```

17. Konversi string ke bilangan

```
// Konversi string ke bilangan bulat
import java.util.Scanner;

public class StringkeBil {
    public static void main(String[] args) {
        String st;
        int bilBulat;

        //Tes pertama
        st = "123";
        bilBulat = Integer.valueOf(st);
        System.out.println(bilBulat);

        //Tes Kedua
        st = "123";
        bilBulat = Integer.parseInt(st);
        System.out.println(bilBulat);
    }
}
```

TUGAS

Buatlah beberapa program Java sesuai ketentuan berikut:

1. Program 1 – Panjang dan Karakter String

- Input: nama lengkap pengguna.
- Output:
 - Menampilkan jumlah total karakter dengan `length()`.
 - Menampilkan huruf pertama dan terakhir dengan `charAt()`.
- Contoh keluaran:
Masukkan nama lengkap: Madiun
Panjang karakter: 6
Huruf pertama: M
Huruf terakhir: n

2. Program 2 – Pembandingan Dua String

- Input: dua buah string dari pengguna.
- Lakukan pembandingan dengan:
 - `equals()`
 - `equalsIgnoreCase()`
 - `compareTo()`
- Contoh keluaran:
- Masukkan string pertama: UNS Madiun
Masukkan string kedua: uns madiun
Hasil `equals()`: false
Hasil `equalsIgnoreCase()`: true
Hasil `compareTo()`: -32

3. Program 3 – Penggabungan String

Tampilkan tiga contoh penggabungan string dengan cara berbeda:

- Menggunakan operator `+`
- Menggunakan metode `concat()`
- Menggunakan metode `join()`

Contoh keluaran:

- Contoh 1 (`+`) : Dian pergi ke pasar
- Contoh 2 (`concat`) : Belajar Java itu menyenangkan
- Contoh 3 (`join`) : Pemrograman - Dasar – Java

Ketentuan pengumpulan :

1. Screenshot hasil praktikum (sourcecode dan hasil outputnya)
2. Kumpulkan Tugas berupa Screenshot sourcecode dan Hasil output minimal dua contoh input