

# **LAPORAN RESMI**

## **Praktikum 5 String**

Mata Kuliah: Praktek Pemrograman Berbasis Objek



Disusun oleh:

M. Ainur Ramadhan (3122500047)

2 D3 Teknik Informatika B

Dosen Pengampu: Yanuar Risah Prayogi S.Kom., M.Kom.

**PROGRAM STUDI D3 TEKNIK INFORMATIKA  
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

**2023/2024**

## A. TUGAS PENDAHULUAN

### 1. Apakah perbedaan class String, StringBuffer dan StringBuilder?

Jawab :

**String:** String memiliki metode untuk manipulasi individual karakter, perbandingan string, pencarian string, pengambilan substring, pembuatan salinan string, serta perubahan menjadi huruf kecil atau besar. String adalah immutable, artinya tidak dapat diubah setelah dibuat, dan tidak aman digunakan oleh beberapa thread.

**StringBuffer:** StringBuffer mirip dengan String dalam banyak hal, tetapi bersifat mutable. Ini berarti Anda dapat mengubahnya atau memodifikasinya menggunakan berbagai metode yang dimilikinya. StringBuffer aman digunakan oleh beberapa thread karena metodenya bersifat synchronized, sehingga operasi-operasi pada StringBuffer akan diselesaikan secara serial sesuai urutan pemanggilan. Operasi utamanya adalah append (menambahkan teks di akhir string) dan insert (menambahkan teks di posisi tertentu).

**StringBuilder:** StringBuilder juga mutable seperti StringBuffer, tetapi tidak mendukung sinkronisasi. Oleh karena itu, StringBuilder lebih efisien dalam konteks single-threading. Ini memiliki operasi yang serupa dengan StringBuffer, seperti append dan insert.

### 2. Apakah yang dimaksud dengan sifat mutable dan immutable? Beri contoh!

Jawab :

- **Mutable (Mutable):** Objek yang bersifat mutable dapat diubah atau dimodifikasi setelah dibuat. Ini berarti nilai-nilai dalam objek tersebut dapat diperbarui tanpa harus membuat objek baru.

Contoh :

```
StringBuilder sb = new StringBuilder("Hello");  
sb.append(" World"); // Mengubah nilai sb menjadi "Hello World"
```

- **Immutable (Immutable):** Objek yang bersifat immutable tidak dapat diubah setelah dibuat. Ini berarti setiap kali ingin mengubah nilainya, akan membuat objek baru yang berisi nilai yang diinginkan.

Contoh :

```
String str1 = "Hello";  
String str2 = str1 + " World"; // Membuat string baru "Hello World"
```

### 3. Jelaskan operasi utama append dan insert yang dimiliki oleh StringBuffer!

- **append:** Metode append digunakan untuk menambahkan teks atau nilai ke akhir string yang ada dalam objek StringBuffer. Ini adalah operasi yang paling umum digunakan untuk memperpanjang string yang sedang dibangun. Metode ini memiliki beberapa bentuk yang berbeda, termasuk yang dapat menerima tipe data primitif dan objek referensi.

Contoh :

```
StringBuffer sb = new StringBuffer("Hello");  
sb.append(" World"); // Menambahkan " World" ke akhir string  
System.out.println(sb.toString()); // Output: "Hello World"
```

- insert: Metode insert digunakan untuk menyisipkan teks atau nilai ke posisi tertentu dalam string yang ada dalam objek StringBuffer. Dengan insert, kita dapat menentukan posisi di mana teks harus disisipkan dalam string.

Contoh :

```
StringBuffer sb = new StringBuffer("Hello");
sb.insert(5, " World"); // Menyisipkan " World" setelah indeks 5
(sesudah 'o')
System.out.println(sb.toString()); // Output: "Hello World"
```

## B. LATIHAN

Latihan 1 → Apakah output program dibawah ini?

Source Code

```
class Str1 {
    public static void main(String[] args) {
        String s = new String("Bicycle");
        int iBegin = 1;
        char iEnd = 3;
        System.out.println(s.substring(iBegin, iEnd));
    }
}
```

Analisa

- Kelas Str1 dideklarasikan dengan kata kunci class. Kelas ini merupakan titik masuk program karena mengandung method main.
- Method main dideklarasikan sebagai public dan static, yang berarti dapat diakses tanpa membuat instance dari kelas Str1.
- Method main menerima argumen array String dengan nama args.
- Di dalam method main, sebuah variabel String dengan nama s dideklarasikan dan diinisialisasi dengan nilai "Bicycle".
- Sebuah variabel int dengan nama iBegin dideklarasikan dan diisi dengan nilai 1.
- Sebuah variabel char dengan nama iEnd dideklarasikan dan diisi dengan nilai 3.
- Method substring dipanggil pada variabel s dengan iBegin dan iEnd sebagai parameter. Method substring mengembalikan String baru yang merupakan substring dari string s asli, dimulai dari indeks iBegin dan berakhir pada indeks iEnd (tidak termasuk indeks iEnd).
- Hasil dari method substring dicetak ke konsol menggunakan method println dari objek System.out.

Output dari program akan bergantung pada nilai iBegin dan iEnd. Dalam kasus ini, iBegin adalah 1 dan iEnd adalah 3. Karena method substring digunakan dengan nilai-nilai tersebut, maka akan mengambil karakter pada indeks 1 dan 2 (tidak termasuk indeks iEnd) dari string s, yang dalam hal ini adalah "Bicycle". Oleh karena itu, output dari program akan menjadi "ic".

Latihan 2 → Apa yang terjadi bila kode berikut ini dikompilasi dan dijalankan?

Jelaskan!

Source Code

```
public class Conv {
    public static void main(String argv[]) {
        Conv c = new Conv();
        String s = new String("ello");
        c.amethod(s);
    }

    public void amethod(String s) {
        char c = 'H';
        c += s;
        System.out.println(c);
    }
}
```

Analisa code

- Kelas Conv dideklarasikan dengan kata kunci public class.
- Method main dideklarasikan sebagai public dan static, yang berarti dapat diakses tanpa membuat instance dari kelas Conv. Method ini memiliki parameter String[] argv.
- Di dalam method main, sebuah variabel Conv dengan nama c dideklarasikan dan diinisialisasi dengan instance baru dari kelas Conv.
- Sebuah variabel String dengan nama s dideklarasikan dan diinisialisasi dengan nilai "ello".
- Method amethod dipanggil pada objek c dengan argumen s.
- Method amethod dideklarasikan sebagai public dan tidak static. Method ini memiliki parameter String s.
- Di dalam method amethod, sebuah variabel char dengan nama c dideklarasikan dan diinisialisasi dengan karakter 'H'.
- Nilai dari variabel c ditambah dengan nilai s.
- Hasil dari penjumlahan tersebut dicetak ke konsol menggunakan method println dari objek System.out.

Output code

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 2> javac *.java
Conv.java:10: error: incompatible types: String cannot be converted to char
        c += s;
          ^
1 error
```

Pada Output terdapat error yang disebabkan karena karakter tidak dapat langsung dijumlahkan dengan string. Untuk itu berikut kode yang benar agar bisa menampilkan output yang benar.

```
public class Conv {
    public static void main(String argv[]) {
```

```

    Conv c = new Conv();
    String s = new String("ello");
    c.amethod(s);
}

public void amethod(String s) {
    char c = 'H';
    String result = Character.toString(c) + s;
    System.out.println(result);
}
}

```

Dalam kode tersebut, karakter 'H' diubah menjadi string menggunakan method `Character.toString(c)`, dan kemudian digabungkan dengan string `s`. Output dari program ini akan menjadi "ello".

Latihan 3 → Apa yang terjadi bila kode berikut ini dikompile dan dijalankan?  
Jelaskan!

Source Code

```

public class EqTest3 {
    public static void main(String argv[]) {
        EqTest e = new EqTest();
    }

    EqTest(){
        String s1 = "Java";
        String s2 = "Java";
        String s3 = "java";
        if(s1 == s2) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
        if(s1 == s3) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
    }
}

```

Output Code

```

C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 3> javac *.java
EqTest3.java:6: error: invalid method declaration; return type required
    EqTest(){
    ^
EqTest3.java:10: error: illegal start of expression
        if(s1 == s2)) {
            ^
EqTest3.java:12: error: 'else' without 'if'
        }else{
        ^
EqTest3.java:15: error: illegal start of expression
        if(s1 == s3)) {
            ^
EqTest3.java:17: error: 'else' without 'if'
        }else{
        ^
5 errors

```

Pada output tersebut terdapat beberapa error, yang mana terdapat pemanggilan metode yang kurang benar pada EqTest() serta tanda kurung yang berganda setelah pernyataan if. Maka untuk itu perlu perbaikan agar kode dapat berjalan dengan benar. Berikut perbaikan kode tersebut.

```

public class EqTest3 {
    public static void main(String argv[]) {
        EqTest();
    }

    public static void EqTest() {
        String s1 = "Java";
        String s2 = "Java";
        String s3 = "java";
        if (s1 == s2) {
            System.out.println("Equal");
        } else {
            System.out.println("Not equal");
        }
        if (s1 == s3) {
            System.out.println("Equal");
        } else {
            System.out.println("Not equal");
        }
    }
}

```

Analisa :

Pada kode program tersebut yang dibenahi adalah pada main utama memanggil method EqTest() yang telah dibuat di bagian bawah main utama dengan menggunakan public static void EqTest serta menghapus tanda kurung tutup yang ganda setelah pernyataan if. Maka setelah perbaikan kode, berikut output yang ditampilkan pada kode program tersebut.

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 3>java EqTest3
Equal
Not equal
```

Latihan 4 → Apa yang terjadi bila kode berikut ini dikompilasi dan dijalankan?  
Jelaskan!

Source code

```
public class EqTest4 {
    public static void main(String argv[]) {
        EqTest e = new EqTest();
    }

    EqTest(){
        String s1 = new String("Java");
        String s2 = new String("Java");
        String s3 = new String("java");
        if(s1 == s2)) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
        if(s1 == s3)) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
    }
}
```

Output code

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 4>javac *.java
EqTest4.java:6: error: invalid method declaration; return type required
    EqTest(){
    ^
EqTest4.java:10: error: illegal start of expression
        if(s1 == s2)) {
        ^
EqTest4.java:12: error: 'else' without 'if'
        }else{
        ^
EqTest4.java:15: error: illegal start of expression
        if(s1 == s3)) {
        ^
EqTest4.java:17: error: 'else' without 'if'
        }else{
        ^
5 errors
```

Pada kode tersebut terdapat kesalahan yang sama seperti kode program pada Latihan 3. Berikut perbaikan pada kode tersebut.

```
public class EqTest4 {
    public static void main(String argv[]) {
        EqTest();
    }

    public static void EqTest(){
        String s1 = new String("Java");
        String s2 = new String("Java");
        String s3 = new String("java");
        if(s1 == s2) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
        if(s1 == s3) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
    }
}
```

Perbaikan sama seperti sebelumnya, dan berikut output yang ditampilkan pada kode program tersebut.

Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 4>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 4>java EqTest4
Not equal
Not equal
```

Analisa

Kelas EqTest4 dideklarasikan sebagai public. Kelas ini berisi method main, yang merupakan titik masuk program.

Method main menerima sebuah array string (String argv[]) sebagai parameter.

Di dalam method main, method EqTest dipanggil. Method ini dideklarasikan sebagai static, yang berarti dapat dipanggil tanpa membuat objek dari kelas tersebut.



Method EqTest dimulai dengan membuat tiga objek string: s1, s2, dan s3. Setiap objek ini dibuat menggunakan kata kunci new, yang membuat instance baru dari kelas String.

String-string "Java", "Java", dan "java" diberikan kepada s1, s2, dan s3, secara berturut-turut.

Program kemudian membandingkan s1 dan s2 menggunakan operator ==. Operator ini memeriksa apakah dua variabel merujuk ke lokasi memori yang sama. Dalam kasus ini, karena s1 dan s2 adalah objek terpisah yang dibuat menggunakan kata kunci new, mereka tidak merujuk ke lokasi memori yang sama. Oleh karena itu, kondisi s1 == s2 bernilai false, dan program mencetak "Not equal" (tidak sama).

Program kemudian membandingkan s1 dan s3 menggunakan operator ==. Sekali lagi, karena s1 dan s3 adalah objek terpisah, kondisi s1 == s3 bernilai false, dan program mencetak "Not equal" kembali.

Latihan 5 → Apa yang terjadi bila kode berikut ini dikompilasi dan dijalankan?  
Jelaskan!

Source code

```
public class EqTest5 {
    public static void main(String argv[]) {
        EqTest e = new EqTest();
    }

    EqTest(){
        String s = "Java";
        String s2 = "java";
        if(s.equalsIgnoreCase(s2)) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
    }
}
```

Output code

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 5> javac *.java
EqTest5.java:6: error: invalid method declaration; return type required
    EqTest(){
    ^
1 error
```

Terdapat error pada kode tersebut, yakni kesalahan dalam pendefinisian fungsi Eqtest(), seharusnya menggunakan public static void sebelum EqTest() sama seperti sebelumnya.

Berikut perbaikan pada kode tersebut.

```
public class EqTest5 {
    public static void main(String argv[]) {
        EqTest();
    }

    public static void EqTest(){
        String s = "Java";
        String s2 = "java";
        if(s.equalsIgnoreCase(s2)) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
    }
}
```

Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 5>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 5>java EqTest5
Equal
```

Analisa

Kelas EqTest5 dideklarasikan sebagai public. Kelas ini berisi method main, yang merupakan titik masuk program.

Method main menerima sebuah array string (String argv[]) sebagai parameter.

Di dalam method main, sebuah objek EqTest dibuat dengan menggunakan konstruktor EqTest().

Di dalam konstruktor EqTest(), terdapat dua variabel string yaitu s dan s2. Variabel s memiliki nilai "Java" dan variabel s2 memiliki nilai "java".

Program kemudian membandingkan nilai dari s dan s2 menggunakan method equalsIgnoreCase(). Method ini membandingkan dua string dengan mengabaikan perbedaan huruf besar dan kecil. Jika kedua string memiliki nilai yang sama, meskipun perbedaan huruf besar dan kecil, maka kondisi s.equalsIgnoreCase(s2) akan bernilai true, dan program akan mencetak "Equal" (sama).

Jika kedua string memiliki nilai yang berbeda, termasuk perbedaan huruf besar dan kecil, maka kondisi s.equalsIgnoreCase(s2) akan bernilai false, dan program akan mencetak "Not equal" (tidak sama).

Dalam kasus ini, karena nilai dari s ("Java") dan s2 ("java") berbeda dalam hal huruf besar dan kecil, maka kondisi `s.equalsIgnoreCase(s2)` akan bernilai false, dan program akan mencetak "Not equal" (tidak sama).

Dengan menggunakan method `equalsIgnoreCase()`, perbandingan string dapat dilakukan tanpa memperhatikan perbedaan huruf besar dan kecil.

Latihan 6 → Apa yang terjadi bila kode berikut ini dikompilasi dan dijalankan?

Jelaskan!

Source code

```
public class EqTest6 {
    public static void main(String argv[]) {
        String str = "Java";
        StringBuffer buffer = new StringBuffer(str);
        if (str.equals(buffer)) {
            System.out.println("Both are equal");
        } else {
            System.out.println("Both are not equal");
        }
    }
}
```

Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 6>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 6>java EqTest6
Both are not equal
```

Analisa

Di dalam method main, sebuah variabel String str dideklarasikan dan diberi nilai "Java".

Selanjutnya, sebuah objek StringBuffer buffer dibuat dengan menggunakan constructor StringBuffer yang menerima objek String str sebagai argumen.

Kemudian, dilakukan perbandingan antara str dan buffer menggunakan method `equals()`. Method ini membandingkan konten dari dua objek. Jika konten kedua objek tersebut sama, maka kondisi `str.equals(buffer)` akan bernilai true.

Jika kondisi `str.equals(buffer)` bernilai true, maka program akan mencetak "Both are equal" (keduanya sama). Jika kondisi tersebut bernilai false, maka program akan mencetak "Both are not equal" (keduanya tidak sama).

Namun, perlu diketahui bahwa objek String dan StringBuffer memiliki struktur internal yang berbeda. Objek String bersifat immutable, artinya tidak dapat diubah setelah dibuat. Sementara itu, StringBuffer bersifat mutable, yang berarti dapat diubah.

Dalam kode ini, karena objek String dan StringBuffer memiliki struktur internal yang berbeda, kondisi `str.equals(buffer)` akan selalu bernilai false. Oleh karena itu, program akan mencetak "Both are not equal" (keduanya tidak sama).

Latihan 7 → Apa yang terjadi bila kode berikut ini dikompilasi dan dijalankan? Jelaskan!

Source code

```
public class Test {
    public static void main(String args[]) {
        String str1 = "abc";
        String str2 = "def";
        String str3 = str1.concat(str2);
        str1.concat(str2);
        System.out.println(str1);
    }
}
```

Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 7>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Latihan\Latihan 7>java Test
abc
```

Analisa

Dua variabel String, yaitu `str1` dan `str2`, dideklarasikan dengan nilai "abc" dan "def" secara berturut-turut.

Metode `concat` dipanggil pada `str1` dengan `str2` sebagai argumen, dan hasilnya disimpan dalam variabel String baru yaitu `str3`. Hal ini menggabungkan `str2` di akhir `str1` dan mengassign hasilnya ke `str3`.

Metode `concat` dipanggil lagi pada `str1` dengan `str2` sebagai argumen. Namun, nilai yang dikembalikan dari pemanggilan metode ini tidak di-assign ke variabel manapun, sehingga tidak memiliki efek pada nilai `str1`.

Metode `println` dipanggil dengan `str1` sebagai argumen untuk mencetak nilainya.

Output yang akan dihasilkan adalah "abc" karena `str1` tidak pernah di-assign dengan hasil pemanggilan metode `concat` yang kedua. Metode `concat` mengembalikan objek String baru yang mewakili penggabungan dua string, tetapi tidak mengubah string asli.

Secara keseluruhan, kode tersebut menggabungkan str2 di akhir str1 dan menyimpan hasilnya di str3. Namun, nilai asli dari str1 tetap tidak berubah.

### C. TUGAS

#### Tugas 1 →

Buatlah sebuah program yang menampilkan indeks-indeks karakter pada suatu kalimat.

Contoh:

Input : Kalimat : "Politeknik Elektronika Negeri Surabaya"

Karakter : 'o'

Output: Karakter 'o' terdapat pada indeks 1, 17

#### Source Code

```
public class FindCharacter {
    public static void main(String[] args) {
        String kalimat = "Politeknik Elektronika Negeri Surabaya";
        char karakter = 'o';

        System.out.print("Karakter '" + karakter + "' terdapat pada
indeks: ");
        int count = 0; // Untuk melacak jumlah karakter yang telah
ditemukan

        for (int i = 0; i < kalimat.length(); i++) {
            if (kalimat.charAt(i) == karakter) {
                count++;
                if (count > 1) {
                    System.out.print(", ");
                }
                System.out.print(i);
            }
        }
    }
}
```

#### Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 1>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 1>java FindCharacter
Karakter 'o' terdapat pada indeks: 1, 17
```

#### Analisa

Variabel kalimat menyimpan string "Politeknik Elektronika Negeri Surabaya" yang akan dicari karakter 'o'-nya.

Variabel karakter menyimpan karakter 'o', yang akan dicari dalam string kalimat.

Pesan awal dicetak menggunakan System.out.print, yang berisi karakter yang akan dicari, misalnya "Karakter 'o' terdapat pada indeks: ".

Variabel count digunakan untuk menghitung berapa kali karakter 'o' ditemukan dalam string kalimat. Ini berguna untuk memutuskan apakah kita perlu mencetak koma dan spasi setelah indeks pertama yang ditemukan.

Program menggunakan loop for untuk mengiterasi melalui setiap karakter dalam string kalimat. Di dalam loop, jika karakter pada indeks ke-i sama dengan karakter yang dicari ('o'), maka count akan ditambah satu.

Jika count lebih dari 1, program mencetak koma dan spasi (, ) sebagai pemisah sebelum mencetak indeks karakter yang ditemukan.

Hasil keluaran program adalah indeks-indeks di mana karakter 'o' ditemukan dalam string kalimat.

## Tugas 2 →

Desain dan implementasikan program Java yang mampu melakukan beberapa operasi terhadap string "Selamat Datang di PENS" :

- a. Konversikan semua karakter menjadi huruf kapital dan tampilkan ke layar
- b. Konversikan semua karakter menjadi huruf kapital dan tampilkan ke layar
- c. Tampilkan panjang string
- d. Tampilkan indeks kata "PENS"

## Source Code

```
public class StringOperation {
    public static void main(String[] args) {
        String kalimat = "Selamat Datang di PENS";

        // a. Konversi ke huruf kapital
        String kapital = kalimat.toUpperCase();
        System.out.println("a. Huruf kapital: " + kapital);

        // b. Konversi ke huruf kecil
        String kecil = kalimat.toLowerCase();
        System.out.println("b. Huruf kecil: " + kecil);
    }
}
```

```

// c. Panjang string
int panjang = kalimat.length();
System.out.println("c. Panjang string: " + panjang);

// d. Indeks kata "PENS"
int indeksPENS = kalimat.indexOf("PENS");
System.out.println("d. Indeks kata \"PENS\": " + indeksPENS);
}
}

```

## Output

```

C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 2>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 2>java StringOperation
a. Huruf kapital: SELAMAT DATANG DI PENS
b. Huruf kecil: selamat datang di pens
c. Panjang string: 22
d. Indeks kata "PENS": 18

```

## Analisa

Variabel kalimat menyimpan string "Selamat Datang di PENS".

- Konversi ke huruf kapital: Menggunakan method `toUpperCase()` pada string kalimat untuk mengonversi semua huruf dalam string menjadi huruf kapital (besar).
- Konversi ke huruf kecil: Menggunakan method `toLowerCase()` pada string kalimat untuk mengonversi semua huruf dalam string menjadi huruf kecil.
- Panjang string: Menggunakan method `length()` untuk mendapatkan panjang (jumlah karakter) dari string kalimat.
- Indeks kata "PENS": Menggunakan method `indexOf()` untuk mencari indeks pertama dari kata "PENS" dalam string kalimat.

## Tugas 3 →

Buatlah sebuah program yang mengubah huruf pertama suatu string dengan huruf terakhir string tersebut dan sebaliknya!

Contoh:       String masukan       : Amanda Ace

              String hasil           : Emende Eca

Program harus bisa berjalan untuk semua inputan string.

## Source Code

```

import java.util.Scanner;

public class SwapCharacter {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan string: ");
        String string = input.nextLine();
        String hasil = ubahHuruf(string);
        System.out.println("Hasil: " + hasil);
    }

    public static String ubahHuruf(String string) {
        if (string.length() <= 1) {
            return string;
        } else {
            StringBuilder sb = new StringBuilder(string);
            char firstChar = sb.charAt(0);
            char lastChar = sb.charAt(string.length() - 1);
            sb.setCharAt(0, lastChar);
            sb.setCharAt(string.length() - 1, firstChar);
            return sb.toString();
        }
    }
}

```

## Output

```

C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 3>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 3>java SwapCharacter
Masukkan string: emanda eci
Hasil: imanda ece

```

## Analisa

Program mengimpor kelas Scanner untuk membaca masukan dari pengguna.

Program membuat objek Scanner yang disebut input untuk membaca masukan dari pengguna.

Program mencetak pesan "Masukkan string: " untuk meminta pengguna memasukkan sebuah string.

Program menggunakan input.nextLine() untuk membaca string yang dimasukkan oleh pengguna dan menyimpannya dalam variabel string.

Program memanggil method ubahHuruf(string) untuk menukar karakter pertama dengan karakter terakhir dalam string yang dimasukkan.





## Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 4>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 4>java SwapNames
Masukkan nama pertama: katy Perry
Masukkan nama kedua: Taylor Swift
String nama pertama: katy Swift
String nama kedua: Taylor Perry
```

## Analisa

Program mengimpor kelas Scanner untuk membaca masukan dari pengguna.

Program membuat objek Scanner yang disebut scanner untuk membaca masukan dari pengguna.

Program mencetak pesan "Masukkan nama pertama: " untuk meminta pengguna memasukkan nama pertama dan menyimpannya dalam variabel name1.

Program mencetak pesan "Masukkan nama kedua: " untuk meminta pengguna memasukkan nama kedua dan menyimpannya dalam variabel name2.

Program menggunakan method split(" ") untuk memecah setiap nama lengkap menjadi dua bagian, yaitu nama depan (first name) dan nama belakang (last name), dan menyimpannya dalam array name1Parts dan name2Parts.

Program menukar nama belakang antara dua nama dengan menggabungkan nama depan dari name1 dengan nama belakang dari name2 untuk mendapatkan newName1, dan sebaliknya, menggabungkan nama depan dari name2 dengan nama belakang dari name1 untuk mendapatkan newName2.

Hasil nama-nama yang baru setelah pertukaran nama belakang dicetak ke layar dengan pesan "String nama pertama: " dan "String nama kedua: ".

Program menutup objek Scanner setelah selesai membaca masukan.

## Tugas 5 →

Buatlah sebuah program yang mampu mengganti kata tertentu suatu string.

Contoh:

Input : String kalimat utama : Praktikum di laboratorium Database  
String yang diganti : Database  
String pengganti : Sistem Informasi  
Output : Kalimat utama menjadi : Praktikum di laboratorium Sistem Informasi  
Program harus bisa berjalan untuk semua inputan string.

Source Code

```
import java.util.Scanner;

public class ReplaceWord {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkan kalimat: ");
        String sentence = scanner.nextLine();

        System.out.print("Masukkan kata yang diganti: ");
        String oldWord = scanner.nextLine();

        System.out.print("Masukkan kata pengganti: ");
        String newWord = scanner.nextLine();

        // Replace the old word with the new word
        String newSentence = sentence.replace(oldWord, newWord);

        // Print the new sentence
        System.out.println("Kalimat baru: " + newSentence);

        scanner.close();
    }
}
```

Output

```
C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 5>javac *.java
C:\Users\USER\OOP\Praktikum\Praktikum 5\Tugas\Tugas 5>java ReplaceWord
Masukkan kalimat: Praktikum Basis Data
Masukkan kata yang diganti: Basis Data
Masukkan kata pengganti: Pemrograman Berbasis Obyek
Kalimat baru: Praktikum Pemrograman Berbasis Obyek
```

## Analisa

Program mengimpor kelas Scanner untuk membaca masukan dari pengguna.

Program membuat objek Scanner yang disebut scanner untuk membaca masukan dari pengguna.

Program mencetak pesan "Masukkan kalimat: " untuk meminta pengguna memasukkan sebuah kalimat dan menyimpannya dalam variabel sentence.

Program mencetak pesan "Masukkan kata yang diganti: " untuk meminta pengguna memasukkan kata yang ingin diganti dan menyimpannya dalam variabel oldWord.

Program mencetak pesan "Masukkan kata pengganti: " untuk meminta pengguna memasukkan kata pengganti yang baru dan menyimpannya dalam variabel newWord.

Program menggunakan method replace(oldWord, newWord) pada string sentence untuk menggantikan setiap kemunculan kata oldWord dengan newWord dalam kalimat.

Hasil kalimat yang baru dengan kata-kata yang telah diganti dicetak ke layar dengan pesan "Kalimat baru: ".

Program menutup objek Scanner setelah selesai membaca masukan.

## **KESIMPULAN**

Penggunaan string dalam bahasa pemrograman Java sangat penting karena teks adalah salah satu jenis data yang paling umum digunakan dalam pengembangan perangkat lunak. Memahami operasi-operasi yang dapat dilakukan pada string dan bagaimana mengelolanya dengan benar adalah kunci untuk mengembangkan aplikasi Java yang efisien dan kuat.