

LAPORAN RESMI

Praktikum 5 String

Mata Kuliah: Praktek Pemrograman Berbasis Objek



Disusun oleh:

Bayu Hadi Leksana (3122500046)

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:

Di Java, terdapat tiga kelas yang umum digunakan untuk memanipulasi teks: String, StringBuffer, dan StringBuilder. Perbedaan utama antara ketiganya terletak pada sifat mutabilitas (apakah mereka dapat diubah setelah dibuat) dan kinerja (seberapa efisien mereka dalam mengubah teks).

a. String:

- Immutability (tidak dapat diubah): Objek String bersifat tidak dapat diubah setelah dibuat. Artinya, setiap kali kita melakukan perubahan pada String, sebenarnya kita membuat objek String baru.
- Keamanan (thread-safe): String adalah thread-safe, yang berarti bisa digunakan di lingkungan multithread tanpa risiko kesalahan.
- Kinerja rendah untuk pengubahan: Jika kita perlu melakukan operasi pengubahan teks secara berulang, penggunaan String dapat mengakibatkan alokasi memori yang berlebihan dan mempengaruhi kinerja aplikasi.

b. StringBuffer:

- Mutabilitas (dapat diubah): StringBuffer adalah kelas yang bersifat mutable. Kita dapat mengubah kontennya tanpa membuat objek baru setiap kali.
- Keamanan (thread-safe): StringBuffer dirancang untuk bekerja dengan aman dalam lingkungan multithread. Ini memiliki metode yang disinkronkan (synchronized) yang memungkinkan penggunaannya dalam thread yang berbeda.
- Kinerja yang lebih baik untuk pengubahan: StringBuffer lebih efisien daripada String ketika kita perlu melakukan banyak operasi pengubahan pada teks, karena ia menghindari alokasi memori berlebihan.

c. StringBuilder:

- Mutabilitas (dapat diubah): Seperti StringBuffer, StringBuilder juga bersifat mutable dan memungkinkan perubahan teks tanpa membuat objek baru setiap kali.
- Tidak aman untuk thread (non-thread-safe): StringBuilder tidak memiliki mekanisme sinkronisasi, sehingga lebih efisien dalam penggunaan single-threaded. Ini tidak aman untuk digunakan dalam thread yang berbeda tanpa perlindungan eksternal.
- Kinerja terbaik: StringBuilder adalah yang paling efisien dalam hal kinerja ketika kita tahu bahwa tidak akan ada akses konkuren ke objek tersebut.

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

Jawab:

Sifat mutable (dapat diubah) dan immutable (tidak dapat diubah) mengacu pada kemampuan objek untuk mengalami perubahan setelah objek tersebut dibuat. Ini adalah konsep yang penting dalam pemrograman dan memiliki implikasi besar terhadap

bagaimana objek digunakan dan diproses dalam sebuah program. Mari jelaskan kedua konsep ini beserta contohnya:

a. **Mutable (Dapat Diubah):**

- Objek yang bersifat mutable adalah objek yang dapat mengalami perubahan setelah objek tersebut dibuat.
- Setiap kali Anda mengubah objek mutable, Anda tidak perlu membuat objek baru. Nilai dalam objek ini dapat berubah langsung.
- Contoh umum objek mutable adalah `StringBuilder`, `StringBuffer`, daftar (seperti `ArrayList`), dan sebagian besar objek koleksi di Java.

b. **Immutable (Tidak Dapat Diubah):**

- Objek yang bersifat immutable adalah objek yang tidak dapat mengalami perubahan setelah objek tersebut dibuat. Jika Anda ingin mengubah nilainya, Anda harus membuat objek baru.
- Immutable seringkali lebih aman dan lebih mudah digunakan dalam lingkungan multithread karena tidak perlu memperhatikan perubahan yang mungkin terjadi oleh thread lain.
- Contoh umum objek immutable adalah `String` dan objek yang digunakan untuk representasi data yang tidak boleh berubah (seperti objek tanggal `java.util.Date` dalam Java lama, yang sekarang digantikan oleh `java.time.LocalDate` yang juga immutable).

c. **Jelaskan operasi utama append dan insert yang dimiliki oleh StringBuffer!**

Jawab:

Operasi utama class `StringBuffer` adalah method `append` dan `insert`. Method `append` selalu menambahkan teks diakhir string, sedang method `insert` menambahkan teks di posisi tertentu. Sebagai contoh:

```
StringBuffer sb1 = new StringBuffer("start");
```

```
sb1.append("le"); → maka isi sb1 adalah "startle"
```

```
StringBuffer sb2 = new StringBuffer("start");
```

```
sb2.insert(4,"le"); → maka isi sb1 adalah "starlet"
```

Secara umum, jika sb adalah obyek dari `StringBuffer`, maka `sb.append(x)` sama dengan operasi `sb.insert(sb.length(),x)`.

B. PERCOBAAN

Percobaan 1. Karakter escape

Di dalam String kita dapat memasukkan sekuen dari karakter escape yang terdiri dari satu karakter back slash (`\`) diikuti oleh karakter escape tersebut.

Beberapa escape character yang sering digunakan adalah :

- `\n` untuk baris baru
- `\t` untuk tab
- `\\` untuk karakter back slash (`\`) itu sendiri

Source Code:

```
class Str1{
    public static void main(String [] args){
        String str1 = "PENS";
        String str2 = "pens";
        System.out.println(str1 + "\n" + str2);
    }
}
```

Output:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\percobaan\src>java Str1
PENS
pens
```

Percobaan 2. Membandingkan String

Membandingkan String di java tidak bisa menggunakan operator equals (==) tetapi kita harus menggunakan method dari String yaitu equals dan equalsIgnoreCase. Sesuai dengan nama methodnya, equals digunakan untuk membandingkan objek String secara case-sensitive (huruf kecil dan besar dibedakan) dan sebaliknya equalsIgnoreCase digunakan untuk membandingkan String secara case-insensitive (huruf besar dan kecil tidak dibedakan).

Source Code:

```
class Str2{
    public static void main(String [] args){
        String str1 = "PENS";
        String str2 = "pens";
        System.out.println(str1.equals(str2));
        System.out.println(str1.equalsIgnoreCase(str2));
    }
}
```

Output:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\percobaan\src>java Str2.java
false
true
```

Percobaan 3. Menggabungkan String (Concatenation)

Untuk menggabungkan string kita bisa menggunakan beberapa cara yaitu dengan menggunakan method concat maupun dengan operator + (plus) dan += (plus sama dengan).

Source Code:

```
class Str3{
    public static void main(String [] args){
        String str1 = "Monas";
        System.out.println(str1);
        String str2 = "monas";
        System.out.println(str2);
        String str3 = str1 + str2;
```

```

        System.out.println(str3);
        String str4 = str1.concat(str2);
        System.out.println(str4);
        str1 += str2;
        System.out.println(str1);
    }
}

```

Output:

```

C:\Users\bayuh\IdeaProjects\pbo\praktikum5\percobaan\src>java Str3.java
Monas
monas
Monasmonas
Monasmonas
Monasmonas

```

Percobaan 4. Konversi Otomatis (Automatic Conversion)

Pada saat kita menggunakan concatenation (penggabungan) antara objek string dengan suatu nilai atau variable yang bertipe primitive (int, char, float, dsbnya) maka secara otomatis data primitive tersebut akan dikonversi menjadi string.

Source Code:

```

class Str4{
    public static void main(String [] args){
        int nilai = 100;
        nilai = nilai * 2;
        System.out.println("Angka " + 1 + " adalah angka pertama
bilangan bulat positif");
        System.out.println("100 x 2 = " + nilai);
    }
}

```

Output:

```

C:\Users\bayuh\IdeaProjects\pbo\praktikum5\percobaan\src>java Str4.java
Angka 1 adalah angka pertama bilangan bulat positif
100 x 2 = 200

```

Percobaan 5. Mengganti Nilai String Sederhana

Untuk mengganti nilai text dari String cukup menggunakan method replace dari String.

Source Code:

```

class Str5{
    public static void main(String [] args){
        String nama = "Dora";
        nama = nama.replace("Dor", "Ti");
        System.out.println(nama);
    }
}

```

Output:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\percobaan\src>java Str5  
Tia
```

C. 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));  
    }  
}
```

Output:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>java Str1  
ic
```

Latihan 2. Apa yang terjadi bila kode berikut ini dikompile 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);  
    }  
}
```

Output:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>javac Conv.java  
Conv.java:9: error: incompatible types: String cannot be converted to char  
        c += s;  
          ^  
1 error
```

Penjelasan:

Operator += yang digunakan dalam baris `c += s;` tidak dapat digunakan untuk menggabungkan karakter (char) dengan string (String). Operator += biasanya digunakan untuk menggabungkan nilai numerik. Jika kita ingin menggabungkan karakter dengan string, kita dapat menggunakan operator + untuk menggabungkannya.

Kode di atas mencoba memasukkan karakter c ke dalam string s, yang tidak memungkinkan karena s adalah objek String. Operasi ini akan menyebabkan kesalahan kompilasi. Untuk menggabungkan karakter 'H' dengan string "ello" dalam bahasa Java, kita dapat menggunakan operator +. Berikut adalah kode 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 = c + s;  
        System.out.println(result);  
    }  
}
```

Maka outputnya:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>java Conv  
Hello
```

**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:

```

C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>javac EqTest3.java
EqTest3.java:5: error: invalid method declaration; return type required
    EqTest(){
    ^
EqTest3.java:9: error: illegal start of expression
        if(s1 == s2)) {
        ^
EqTest3.java:11: error: 'else' without 'if'
        }else{
        ^
EqTest3.java:14: error: illegal start of expression
        if(s1 == s3)) {
        ^
EqTest3.java:16: error: 'else' without 'if'
        }else{
        ^
5 errors

```

Penjelasan:

Ada beberapa kesalahan dalam kode di atas:

- Kesalahan sintaksis dalam kondisi if:
Kode di atas menggunakan dua tanda kurung tutup berturut-turut di dalam kondisi if, yang tidak valid. Perlu menghapus tanda kurung tutup ekstra di dalam kondisi if.
- Kode di atas memanggil EqTest di dalam metode main, tetapi konstruktor EqTest tidak didefinisikan dalam kelas EqTest. Kode tersebut seharusnya tidak dapat dikompilasi karena kelas EqTest tidak memiliki konstruktor yang sesuai dengan pemanggilan yang dilakukan di dalam metode main.

Berikut adalah kode yang diperbaiki:

```

public class EqTest3 {
    public static void main(String argv[]) {
        EqTest3 e = new EqTest3(); // Perbaiki nama kelas
    }

    EqTest3() { // Perbaiki nama kelas
        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");
        }
    }
}

```

Maka outputnya adalah:

```

C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>java EqTest3
Equal
Not equal

```


Latihan 4. Apa yang terjadi bila kode berikut ini dikompile dan dijalankan?

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:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>javac EqTest4.java
EqTest4.java:5: error: invalid method declaration; return type required
    EqTest(){
    ^
EqTest4.java:9: error: illegal start of expression
        if(s1 == s2)) {
        ^
EqTest4.java:11: error: 'else' without 'if'
        }else{
        ^
EqTest4.java:14: error: illegal start of expression
        if(s1 == s3)) {
        ^
EqTest4.java:16: error: 'else' without 'if'
        }else{
        ^
5 errors
```

Penjelasan:

Kode di atas terdapat beberapa kesalahan sintaksis:

a. Kesalahan dalam kondisi if:

Terdapat dua tanda kurung tutup) berturut-turut di dalam kondisi if, yang tidak valid. Perlu menghapus salah satu tanda kurung tutup ekstra di dalam kondisi if.

b. Kode memanggil EqTest di dalam metode main, tetapi konstruktor EqTest tidak didefinisikan dalam kelas EqTest. Kode tersebut seharusnya tidak dapat dikompilasi karena kelas EqTest tidak memiliki konstruktor yang sesuai dengan pemanggilan yang dilakukan di dalam metode main.

- c. Kode membuat objek String baru menggunakan `new String("Java")`. Ini akan membuat string baru di dalam heap memory, yang berbeda dengan menggunakan literal string seperti `"Java"` yang akan mengakses string di dalam string pool. Ini akan memengaruhi hasil perbandingan string dengan operator `==`.

Berikut adalah kode yang diperbaiki:

```
public class EqTest4 {
    public static void main(String argv[]) {
        EqTest4 e = new EqTest4(); // Perbaiki nama kelas
    }

    EqTest4() { // Perbaiki nama kelas
        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");
        }
    }
}
```

Maka outputnya adalah:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>java EqTest4
Not equal
Not equal
```

**Latihan 5. Apa yang terjadi bila kode berikut ini dikompile 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:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>javac EqTest5.java
EqTest5.java:5: error: invalid method declaration; return type required
    EqTest(){
        ^
1 error
```

Penjelasan:

Ada beberapa masalah pada kode di atas:

- Seharusnya pada pembuatan konstruktor, harusnya menggunakan EqTest5() karena nama class nya adalah EqTest5.
- Lalu pada fungsi main, pembuatan object seharusnya menggunakan EqTest5.

Perbaikan kode nya adalah:

```
public class EqTest5{
    public static void main(String argv[]){
        EqTest5 e = new EqTest5();
    }
    EqTest5(){
        String s = "Java";
        String s2 = "java";
        if(s.equalsIgnoreCase(s2)) {
            System.out.println("Equal");
        }else{
            System.out.println("Not equal");
        }
    }
}
```

Maka outputnya adalah:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>java EqTest5
Equal
```

**Latihan 6. Apa yang terjadi bila kode berikut ini dikompile 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\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>java EqTest6
Both are not equal
```

Penjelasan:

Kode di atas mencoba membandingkan dua jenis objek yang berbeda, yaitu objek String dan objek StringBuffer, menggunakan metode equals. Inilah penjelasan dari kode tersebut:

- a. Kode mendefinisikan sebuah string str dengan nilai "Java".
- b. Kemudian, kode membuat objek StringBuffer bernama buffer dan menginisialisasinya dengan nilai yang sama, yaitu "Java".
- c. Selanjutnya, kode mencoba membandingkan kedua objek menggunakan metode equals:
Jika menggunakan metode equals pada dua objek yang berbeda jenis (dalam hal ini String dan StringBuffer), hasilnya akan selalu "Both are not equal". Ini karena objek equals memeriksa kesamaan objek berdasarkan identitasnya, dan objek String dan StringBuffer memiliki identitas yang berbeda.
- d. Jika ingin membandingkan nilai (konten) dari kedua objek, kita perlu mengonversi StringBuffer ke String terlebih dahulu, kemudian membandingkannya. Kita dapat melakukan ini dengan menggunakan metode toString() pada objek StringBuffer:

```
if (str.equals(buffer.toString())) {  
    System.out.println("Both are equal");  
} else {  
    System.out.println("Both are not equal");  
}
```

Dengan perubahan ini, kode akan membandingkan nilai konten dari str (string) dan nilai yang diambil dari objek buffer (setelah dikonversi ke string), dan jika nilai kontennya sama, maka hasilnya akan menjadi "Both are equal".

Latihan 7. Apa yang terjadi bila kode berikut ini dikompile 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\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>java Test  
abc
```

Penjelasan:

Kode di atas berikan adalah program Java sederhana yang mencoba untuk menggabungkan dua string (str1 dan str2) menggunakan metode concat dari kelas String. Namun, perlu diingat bahwa String adalah tipe data yang bersifat immutable (tidak dapat diubah). Ini adalah penjelasan dari kode tersebut:

- Kode mendefinisikan tiga variabel string, yaitu `str1`, `str2`, dan `str3`, dengan nilai berturut-turut "abc", "def", dan hasil penggabungan "abcdef" yang dihasilkan dari `str1.concat(str2)`.
- Kemudian memanggil `str1.concat(str2)` lagi, tetapi hasil penggabungan ini tidak disimpan atau digunakan dalam kode. Ini karena pada saat ini, kode hanya memanggil metode `concat`, tetapi kode tidak menyimpan hasil penggabungan ke dalam variabel atau melakukan apa pun dengan hasilnya.
- Terakhir, kode mencoba mencetak nilai `str1` ke layar. Kode ini akan mencetak "abc" ke layar.

Namun, perlu diingat bahwa meskipun kode telah memanggil `str1.concat(str2)` dua kali, nilai dari `str1` tetap tidak berubah. Ini karena string dalam Java bersifat immutable, yang berarti setiap kali kita melakukan operasi penggabungan atau perubahan pada string, kita sebenarnya membuat string baru daripada mengubah yang ada. Jadi, jika Anda ingin menggabungkan `str1` dan `str2` dan menyimpan hasilnya dalam `str1`, kita perlu melakukan sesuatu seperti ini:

```
str1 = str1.concat(str2);
```

Dengan cara ini, kita akan menggabungkan `str1` dan `str2`, dan hasil penggabungannya akan disimpan kembali dalam `str1`.

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\latihan\src>java Test
abcdef
```

D. 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

Jawab:

Source Code:

```
import javax.swing.JOptionPane;
public class IndexKarakter {
    public static void main(String[] args) {
        String kalimat = JOptionPane.showInputDialog("Masukkan kalimat:");
        String str = JOptionPane.showInputDialog("Masukkan karakter yang dicari:");
        char karakter = str.charAt(0);
        System.out.print("Karakter " + karakter + " terdapat pada index");
        for (int i = 0; i < kalimat.length(); i++) {
            char elemenStr = kalimat.charAt(i);
            if (Character.toLowerCase(elemenStr) == Character.toLowerCase(karakter)) {
                System.out.print(i + ", ");
            }
        }
    }
}
```

```
}
}
```

Input:

Output:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\tugas\src>java IndexKarakter
Karakter o terdapat pada index 1, 17,
```

Penjelasan:

1. Pertama, saya membuat inputan menggunakan JOptionPane. Inputan pertama digunakan untuk menginputkan kalimat yang hasil inputannya saya masukkan ke variable kalimat bertipe data string. Inputan kedua digunakan untuk menginputkan kata yang hasil inputannya saya ubah menjadi char dan saya masukkan ke variable karakter bertipe data char.
2. Lalu, untuk mencari karakter pada kalimat, saya lakukan looping mulai dari 0 hingga kurang dari Panjang string kalimat.
3. Pada setiap iterasi, saya buat variable baru Bernama elemenStr bertipe data char yang digunakan untuk menyimpan elemen string kalimat pada index ke-i. Kemudian saya lakukan pengecekan jika isi variabel elemenStr sama dengan variabel karakter maka program mencetak I yang merupakan index dari karakter yang dicari. Perbandingan char ini dilakukan secara non case sensitive karena saya menggunakan Character.toLowerCase().

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 index kata "PENS"

Jawab:

Source Code:

```
public class OperasiString {
    public static void main(String[] args) {
        String kalimat = "Selamat Datang di PENS";
        System.out.println(kalimat.toUpperCase());
        System.out.println(kalimat.length());
        int indeks = cariIndeksKata(kalimat, "PENS");
        if (indeks != -1) {
            System.out.println("Indeks kata PENS dalam kalimat: " +
indeks);
        } else {
            System.out.println("Kata PENS tidak ditemukan dalam
```

```

kalimat.");
    }
}

public static int cariIndeksKata(String kalimat, String kata) {
    String[] kataKunci = kalimat.split(" ");
    for (int i = 0; i < kataKunci.length; i++) {
        if (kataKunci[i].equals(kata)) {
            return i;
        }
    }
    return -1;
}
}

```

Output:

```

C:\Users\bayuh\IdeaProjects\pbo\praktikum5\tugas\src>java OperasiString
SELAMAT DATANG DI PENS
22
Indeks kata PENS dalam kalimat: 3

```

Penjelasan:

1. Untuk mengubah string menjadi huruf besar, kita bisa menggunakan method `toUpperCase()`.
2. Untuk menghitung Panjang string, kita bisa menggunakan method `length()`.
3. Untuk mendapatkan index kata PENS, saya membuat fungsi baru untuk mencari indeks kata yang bernama `cariIndeksKata()`.
4. Pada fungsi itu, pertama saya membuat array `KataKunci` yang berisi setiap kata dari string `kalimat`. Untuk mengubah string menjadi array per kata, bisa menggunakan method `split`. Lalu saya lakukan looping array `KataKunci` yang didalamnya terdapat pengecekan jika `KataKunci[i]` sama dengan kata yang dicari, maka `return i` yang menunjukkan indeks kata nya. Untuk mengecek kesamaan, saya menggunakan method `equals()`. Lalu jika tidak memenuhi pengecekan maka di akhir fungsi akan mengembalikan nilai `-1`.
5. Kemudian pada fungsi main, saya membuat variabel indeks yang diisi dengan fungsi `cariIndeksKata` dengan memasukkan parameter `PENS`. Lalu dicek, jika indeks `!= -1` maka akan print indeks kata, namun jika indeks `= -1` maka akan print kalimat bahwa kata `PENS` tidak ditemukan.

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.

Jawab:

Source Code:

```

import javax.swing.JOptionPane;
public class StringHurufUbah {
    public static void main(String[] args) {
        String str = JOptionPane.showInputDialog("String masukan: ");
    }
}

```

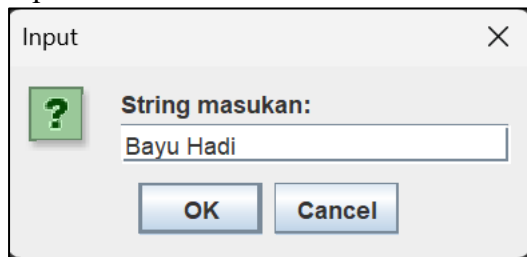
```

StringBuffer kalimat = new StringBuffer(str.toLowerCase());
char hurufPertama = kalimat.charAt(0);
char hurufTerakhir = kalimat.charAt(str.length()-1);
for (int i = 0; i < str.length(); i++) {
    if (kalimat.charAt(i) == hurufPertama) {
        kalimat.setCharAt(i, hurufTerakhir);
    } else if (kalimat.charAt(i) == hurufTerakhir) {
        kalimat.setCharAt(i, hurufPertama);
    }

    if (i==0 || kalimat.charAt(i-1)==' '){
        kalimat.setCharAt(i,
Character.toUpperCase(kalimat.charAt(i)));
    }
}
System.out.print("String Hasil: " + kalimat);
}
}

```

Input:



Output:

```

C:\Users\bayuh\IdeaProjects\pbo\praktikum5\tugas\src>java StringHurufUbah
String Hasil: Iayu Hadb

```

Penjelasan:

Untuk mengubah huruf pertama menjadi huruf terakhir dan sebaliknya, Langkah-langkah yang saya lakukan adalah sebagai berikut:

1. Pertama, saya buat inputan untuk menginputkan kalimat.
2. Lalu saya buat StringBuffer kalimat yang isinya sama dengan hasil inputan yang saya ubah jadi lowercase. StringBuffer digunakan karena bisa dimodifikasi.
3. Proses pengubahannya adalah dengan looping dan if else. Dimana huruf pertama dari string kalimat diubah menjadi huruf terakhir, dan huruf terakhir diubah menjadi huruf pertama.
4. Kemudian huruf pertama string kalimat yang baru diubah ke Uppercase.
5. Terakhir, kalimat yang baru itu diprint.

Tugas 4.

Buatlah sebuah program yang menukar family name dari dua buah string.

Contoh: Input : String nama pertama : Katy Perry

String nama kedua : Taylor Swift

Output : String nama pertama : Katy Swift

String nama kedua : Taylor Perry

Program harus bisa berjalan untuk semua inputan string.

Jawab:

Source Code:

```
import java.util.Scanner;
public class ReversedWord {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("String Nama Pertama: ");
        String inputString1 = input.nextLine();
        System.out.print("String Nama Kedua: ");
        String inputString2 = input.nextLine();
        String hasilString1 = "";
        String hasilString2 = "";
        for(int i = 0; i < inputString1.indexOf(' '); i++){
            hasilString1 += inputString1.charAt(i);
        }
        for(int i = inputString2.indexOf(' '); i <
inputString2.length(); i++){
            hasilString1 += inputString2.charAt(i);
        }
        for(int i = 0; i < inputString2.indexOf(' '); i++){
            hasilString2 += inputString2.charAt(i);
        }
        for(int i = inputString1.indexOf(' '); i <
inputString1.length(); i++){
            hasilString2 += inputString1.charAt(i);
        }
        System.out.println("Hasil 1 : " + hasilString1);
        System.out.println("Hasil 2 : " + hasilString2);
        input.close();
    }
}
```

Hasil:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\tugas\src>java ReversedWord
String Nama Pertama: Bayu Hadi
String Nama Kedua: Ainur Rama
Hasil 1 : Bayu Rama
Hasil 2 : Ainur Hadi
```

Penjelasan:

1. Pertama saya buat inputan untuk nama pertama dan nama kedua. Kali ini saya menggunakan java.util.scanner untuk membuat inputannya.
2. Lalu dilakukan looping pertama yang digunakan untuk mengisi hasilString1 dengan kata pertama dari inputan nama yang pertama.
3. Kemudian looping kedua digunakan untuk menambahkan hasilString1 dengan kata kedua dari nama kedua.
4. Selanjutnya looping ketiga untuk mengisi hasilString2 dengan kata pertama dari nama kedua.
5. Lalu looping keempat untuk menambahkan hasilString2 dengan kata kedua dari nama pertama.
6. Terakhir print hasilString1 dan hasilString2.

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.

Jawab:

Source Code:

```
import java.util.Scanner;
public class ChangedWord {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("String Kalimat Utama: ");
        String inputString1 = input.nextLine();
        System.out.print("String yang diganti: ");
        String inputString2 = input.nextLine();
        System.out.print("String pengganti: ");
        String inputString3 = input.nextLine();
        String hasil;

        if(inputString1.indexOf(inputString2) == 0){
            hasil = inputString3 +
inputString1.substring(inputString2.length(), inputString1.length());
        }
        else{
            hasil = inputString1.substring(0,
inputString1.indexOf(inputString2)) + inputString3 +
inputString1.substring(inputString1.indexOf(inputString2) +
inputString2.length(), inputString1.length());
        }

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

        input.close();
    }
}
```

Hasil:

```
C:\Users\bayuh\IdeaProjects\pbo\praktikum5\tugas\src>java ChangedWord
String Kalimat Utama: Saya Makan Nasi
String yang diganti: Nasi
String pengganti: Sate Ayam
Hasil : Saya Makan Sate Ayam
```

Penjelasan:

Kode di atas adalah sebuah program Java yang meminta input dari user untuk sebuah kalimat utama, sebuah string yang ingin diganti, dan sebuah string pengganti. Program akan melakukan penggantian string yang dimaksud dan menghasilkan output berupa kalimat terbaru setelah penggantian.

1. Pertama, program menampilkan string "String Kalimat Utama: " dan menunggu input dari user dalam bentuk String menggunakan Scanner. Input tersebut kemudian disimpan dalam variable `inputString1`.
2. Selanjutnya, program menampilkan string "String yang diganti: " dan mengambil input String dari user untuk string yang ingin diganti dan menyimpannya pada variable `inputString2`.
3. Program kemudian menampilkan string "String pengganti: " dan mengambil input string pengganti dari user, menyimpannya pada variable `inputString3`.
4. Setelah itu, program melakukan pengecekan dengan menggunakan fungsi `indexOf` untuk melihat apakah string yang ingin diganti ada pada kalimat utama di bagian awal. Jika ya, maka program akan menambahkan string pengganti di awal dan memotong string yang ingin diganti sehingga dihasilkan sebuah string yang baru. Jika tidak, maka program akan memotong string kalimat utama sesuai dengan string yang ingin diganti dan menempatkan string pengganti di tengah. Hasil akhir dari operasi penggantian string disimpan pada variable `hasil`.
5. Akhirnya, program menampilkan string "Hasil : " dan mencetak nilai dari variable `hasil` ke layar. Terakhir, program menutup Scanner menggunakan fungsi `close()`.

E. KESIMPULAN

Class String berisi string yang tetap (immutable string). Artinya sekali instance String dibuat maka isinya tidak bisa diubah. Kelas String memiliki 13 konstruktor yang memungkinkan kita membuat obyek String dan menginisialisasi nilainya dengan menggunakan berbagai macam sumber data yang berbeda. Class String memiliki method untuk menangani individual karakter dari suatu string, membandingkan string, mencari string, mendapatkan substring, membuat kopi isi string dan dijadikan lowercase atau uppercase. Java menyediakan operator spesial untuk menggabungkan dua string yaitu operator plus (+).

Class StringBuffer mirip dengan String tetapi bersifat mutable, atau dapat diubah atau dimodifikasi dengan menggunakan beberapa method yang dimilikinya. String buffer aman digunakan oleh beberapa thread. Method-methodnya bersifat synchronized sehingga beberapa operasi yang terjadi pada suatu obyek string buffer akan diselesaikan secara serial sesuai urutan pemanggilan.

StringBuilder adalah string yang bersifat mutable. Operasi yang dimiliki class StringBuilder mirip dengan class StringBuffer. Perbedaannya adalah StringBuilder tidak mendukung sinkronisasi.