Bahasa Pemrograman

(Pemrograman Visual dengan Java)

Modul Praktikum

# Topik

String, Array & FileReader

# Tujuan

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Membuat program dengan menggunakan object String untuk proses manipulasi terhadap object String.
2. Membuat program dengan menggunakan object array untuk proses manipulasi terhadap object array.
3. Membuat program yang dapat membaca file teks.

# Alat dan Bahan

* Komputer dengan **Java Development Kit (JDK)** dan aplikasi BlueJ sudah terinstall
* Informasi download dan instalasi JDK dapat diperoleh di; <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
* Informasi download dan instalasi eclipse dapat diperoleh di;
* <http://www.csee.umbc.edu/courses/undergraduate/341/fall08/Lectures/Eclipse/intro-to-eclipse.pdf>

# Rangkuman/Teori

## [Menjelaskan tentang tipe data *String*]

*String* dalam pemrograman Java adalah suatu class, dan penanganannya tidak sama dengan type data primitive (spt; int, char, double, boolean,..dsb).

Variable dengan tipe data String pada pemrograman Java untuk selanjutnya akan disebut sebagai object. Berikut adalah deklarasi untuk membuat object String;

**String *<nama\_object>*;**

Contoh:

**String x;** 🡨 perintah ini akan membuat object dengan nama x, dengan tipe String.

**String pesan = “Hallo”;** 🡨 perintah ini akan membuat object dengan nama pesan, dengan tipe String, dan mengisikan tulisan Hallo pada object pesan.

### Method pada *String*

Method yang dapat digunakan oleh object tipe String diantaranya adalah:

* **length()**

Untuk mengetahui panjang karakter dari suatu string.

Contoh:

String kalimat = “ABCDEFGHI”;

int panjang = kalimat.length();

* **charAt(***int* *n***)**

Untuk mengetahui karakter yang tersimpan pada posisi ke-n dalam suatu String.

Contoh:

String kalimat = “ABCDEFGHI”;

char huruf = kalimat.charAt(4);

* **substring(***int b***)**

Untuk mengetahui potongan string yang tersimpan pada suatu String mulai dari posisi ke-b sampai posisi akhir String.

* **substring(***int b, int e***)**

Untuk mengetahui potongan string yang tersimpan pada suatu String mulai dari posisi ke-b, sampai posisi ke-e.

Contoh:

String kalimat = “ABCDEFGHI”;

String potongan1 = kalimat.substring(4);

String potongan2 = kalimat.substring(4, 6);

* **lastIndexOf(***<karakterYangDicari>***)**

Untuk mengetahui posisi index terakhir dari suatu karakter yang dicari pada suatu String.

Contoh:

String kalimat = “AB#CDE#FGHI”;

int idx = kalimat.lastIndexOf(‘#’);

* **indexOf(***<karakterYangDicari>, <indexAwalPencarian>***)**

Untuk mengetahui posisi index dari suatu karakter yang dicari pada suatu String, pencarian dimulai pada posisi index seperti yang dituliskan pada *<indexAwalPencarian>*.

Contoh:

String kalimat = “AB#CDE#FGHI”;

int idx = kalimat.indexOf(‘#’, 3);

* **toCharArray()**

Untuk mengubah object string menjadi array karakter. Biasa digunakan untuk menyalin seluruh karakter pada object String ke object array.

Contoh:

String kalimat = “AB#CDE#FGHI”;

Char [] arrayKarakter = kalimat.toCharArray();

### Konversi pada String

Terdapat beberapa cara konversi angka menjadi string:

* **typeAsal.toString(***angka***)**

Konversi angka yang memiliki type seperti yang disebutkan pada typeAsal menjadi string.

Contoh:

String tipeDouble = Double.toString(14.2); //hasilnya adalah “14.2”

String tipeFloat = Float.toString(1420000000); //hasilnya adalah “1.42E9”

int nilai = 3;

System.out.println(Integer.toString(nilai\*4); //hasilnya adalah “12”

* **Integer.toString(***angka, format***)**

konversi dari integer menjadi string dengan format bilangan yang ditentukan

format: **2** (untuk bilangan biner), **8** (untuk bilangan oktal), **10** (untuk bilangan desimal), **16** (untuk bilangan heksadesimal).

Contoh:

int nilai = 3;

System.out.println(Integer.toString(nilai\*4, 2); //hasilnya adalah “1100”

System.out.println(Integer.toString(nilai\*4, 16); //hasilnya adalah “C”

Cara konversi dari String menjadi angka:

* **Integer.parseInt(***string***)**;

Melakukan konversi dari string (string suatu angka) menjadi angka tipe integer.

**Double.parseDouble(***string***)**;

Melakukan konversi dari string (string suatu angka) menjadi angka tipe double.

Contoh:

String diameter = "22";

String pi = "3.14";

Double kelilingLingkaran **= Integer.parseInt**(diameter) \* **Double.parseDouble**(pi);

System.out.println(kelilingLingkaran);

## [Array]

Array dalam pemrograman Java adalah suatu type data reference sehingga jika kita membuat variable dengan type array maka untuk selanjutnya akan disebut sebagai object.

Berikut adalah beberapa cara deklarasi untuk membuat object Array;

**int [] array1;**

**array1 = new int [100];**

Berdasarkan instruksi-instruksi di atas kita membuat object **array1** dengan kapasistas tampung sebanyak **100 angka tipe integer**.

atau

**char [] arrayC = new char [100];**

Berdasarkan instruksi di atas kita membuat object **arrayC** dengan kapasistas tampung sebanyak **100 karakter**.

atau

**int [] array2 = { 3, 4, 10, 6 };**

Berdasarkan instruksi di atas kita membuat object **array2** dengan kapasistas tampung sebanyak **4 angka tipe integer**, dan array1 sudah diisi dengan angka 3, 4, 10, 6.

### Mengisikan nilai ke object tipe array

Untuk mengisikan suatu nilai pada object tipe array digunakan instruksi sebagai berikut:

**objArray[**<noIndeks>**]** = <nilai>

keterangan:

**objArray** adalah nama object dengan type array yang akan diisi dengan suatu nilai

**<noIndeks>** adalah no indeks posisi nilai akan diisi pada object array (indeks pada array dimulai dari 0)

Contoh:

char [] arrayC = new char[3];

arrayC[0] = ‘K’;

arrayC[1] = ‘a’;

arrayC[2] = ‘k’;

### Perulangan for Khusus untuk Object Array

Object array memiliki perulangan for khusus, berikut adalah penulisan instruksi perulangan tersebut:

**for(type <namaVariable>: <namaObjectArray>)**

keterangan:

**type** adalah type untuk variable yang menampung isi array, jadi type harus sesuai dengan type dari object array (jika object array tipe char, maka type adalah char).

**<namaVariable>** adalah nama variable yang nantinya akan menampung isi array.

**<namaObjectArray>** adalah nama object array yang akan dibaca isinya indeks demi indeks.

Perulangan akan dilakukan sebanyak kapasistas dari object array.

Contoh:

char [] arrayC = new char[3];

arrayC[0] = ‘K’;

arrayC[1] = ‘a’;

arrayC[2] = ‘k’;

**for(char huruf:arrayC) {**

System.out.print(huruf);

**}**

### Method length()

Method length adalah method yang dapat digunakan oleh object array untuk mengetahui kapasitas atau panjang dari suatu object array.

**<namaObjectArray>.length;**

keterangan:

**<namaObjectArray>** adalah nama dari object array yang akan dilihat kapasitasnya atau panjangnya.

Instruksi di atas akan menghasilkan suatu angka integer sebesar kapasistas atau panjang dari object array.

Contoh:

1.

int kapasistasArray = arrayC.length;

2.

for(int i=0;i<arrayC.length;i++){

System.out.print(arrayC[i]);

}

🡨 perintah ini akan membuat object dengan nama x, dengan tipe String.

**String pesan = “Hallo”;** 🡨 perintah ini akan membuat object dengan nama pesan, dengan tipe String, dan mengisikan tulisan Hallo pada object pesan.

## [Class Scanner]

Untuk membaca input melalui sistem konsol pada pemrograman Java dapat digunakan class Scanner. Untuk menggunakan class Scanner kita harus melakukan import class pada awal pemrograman seperti berikut:

**Import java.util.Scanner;**

Selanjutnyadidalam tubuh program buat object dengan type Scanner seperti berikut:

**Scanner inputKonsol** = **new Scanner(System.in);**

### Method

Selanjutnya untuk membaca masukan (*input*) dapat digunakan method-method pada object tersebut (method yang digunakan disesuaikan dengan tipe masukan yang akan dibaca).

**next()**

Untuk membaca input string

**hasNext()**

Untuk verifikasi apakah ada input string berikutnya (outputnya: boolean)

**nextInt()**

Untuk membaca integer

**hasNextInt()**

Untuk verifikasi apakah ada input integer berikutnya (outputnya: boolean)

**nextDouble()**

Untuk membaca double

**hasNextDouble()**

Untuk verifikasi apakah ada input double berikutnya (outputnya: boolean)

**nextLine()**

Untuk membaca satu baris String.

**hasNextLine()**

Untuk verifikasi apakah ada input baris berikutnya (outputnya: boolean)

Contoh:

**import java.util.Scanner;**

public class Scanner\_BacaInput {

public static void main(String[] args) {

**Scanner input\_1 = new Scanner(System.in);**

System.out.print("Nama : ");

String nama = input\_1.**nextLine();**

System.out.println("Usia : ");

int usia = input\_1.**nextInt();**

System.out.println("Terima kasih " + nama + "(" + usia + ")" );

}

}

## [FileReader]

Untuk membuka file dan membaca isinya digunakan class Scanner dan class FileReader. Class FileReader digunakan untuk membuka dan membaca file, sedangkan class Scanner digunakan untuk membaca/mengambil isi file.

Untuk menggunakan class Scanner dan class FileReader kita harus melakukan import class pada awal pemrograman seperti berikut:

**import java.util.Scanner;**

**import java.io.FileReader;**

**import java.io.FileNotFoundException;**

class FileNotFoundException digunakan untuk melakukan pemeriksaan, apakah file yang akan dibuka ada atau tidak ada.

Selanjutnya di dalam tubuh program buat object dengan type Scanner seperti berikut:

**Scanner inputFile** = **new Scanner(new FileReader(<namaFile));**

Parameter yang digunakan pada Scanner adalah **new FileReader(<namaFile>)** (menggantikan System.in), hal ini untuk tujuan memberitahu Scanner bahwa yang dibaca adalah dari file (melalui FileReader). Instruksi FileReader memiliki exception/pengecualian kalau file yang ingin dibuka tidak ditemukan, untuk itu maka instruksi tersebut harus diletakkan di dalam block **try … catch** dengan menggunakan **FileNotFoundException** sebagai pendeteksi kejadian (even handler).

Contoh:

**import java.util.Scanner;**

**import java.io.FileReader;**

**import java.io.FileNotFoundException;**

public class membacaDariFile {

public static void main(String[] args) {

String barisIsi;

try {

Scanner **inputFile** = new Scanner(**new FileReader("daftarNama.txt")**);

while(**inputFile.hasNextLine()**){

barisIsi = **inputFile.nextLine()**;

System.out.println(barisIsi);

}

}

catch(FileNotFoundException e) {

System.out.println(e);

}

}

}

# Praktek

Pada bagian ini, akan dibahas mengenai praktek yang akan dilakukan. Untuk mempermudah pencatatan hasil praktek dan penarikan kesimpulan terhadap hasil praktek tersebut, Anda dapat menggunakan form pada **Appendix 1.**

## Praktek 1: Array

1. Buat class dengan nama: **lab\_3\_Array**
2. Tambahkan kode program untuk membuat object array dengan nama kalimat, dan isi kalimat (object array) dengan karakter seperti berikut:

'k', 'a', 'k', 'i', 'k', 'a', 'k', 'a', 'k', 'k', 'e', 'n', 'a', 'k', 'a', 'y', 'u', 't', 'a', 'k', 'b', 'e', 'r', 'p', 'a', 'k', 'u', 'k', 'a', 'k', 'u', 'k', 'a', 'k', 'u'

1. Tambahkan kode program sehingga program anda dapat menampilkan seluruh isi object array.
2. Tambahkan kode program sehingga program anda dapat menghitung jumlah huruf k yang terdapat pada object array.
3. Tambahkan kode program sehingga tampilan output hasil program anda pada konsol adalah seperti contoh berikut:

**Isi object array: kakikakakkenakayutakberpakukakukaku**

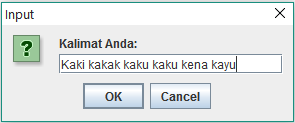
**------------------------------**

**Jumlah huruf dalam array= 35**

**Jumlah huruf k dalam array= 13**

## Praktek 2: String

1. Buat class dengan nama: **lab\_3\_String**.
2. Tambahkan kode program untuk membaca kalimat masukan user.



1. Tambahkan kode program yang menampilkan seluruh kalimat masukan dari user.
2. Tambahkan kode program yang menampilkan seluruh masukan dari user, tetapi ditampilkan kata demi kata. Dan buat tampilan output hasil program anda pada konsol adalah seperti contoh berikut:

**Kaki kakak kaku kaku kena kayu**

**Kaki**

**kakak**

**kaku**

**kaku**

**kena**

**kayu**

1. Jika ada kesalahan, lakukan perbaikan yang diperlukan.

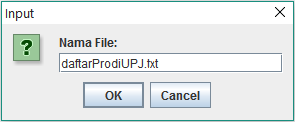
## Praktek 3: Membaca File teks

1. Buat file teks dengan nama: **daftarNama.txt** pada project anda.

Caranya:

* + - click kanan pada project 🡪 new 🡪 Untitled Teks File.
    - Tulis isinya dengan nama teman anda (satu baris untuk tiap satu nama).
    - Click Save 🡪 pilih project anda 🡪 tulis nama file: daftarNama.txt

1. Buat class dengan nama: **lab\_3\_FileReader**.
2. Tambahkan kode program sehingga meminta user memasukkan nama file teks, dengan tampilan seperti berikut:

****

kemudian nama file teks masukan user tersebut dibaca oleh program anda, dan isi file teks tersebut ditampilkan baris demi baris.

1. Buat tampilan output program anda agar memiliki tampilan hasil pada konsol seperti contoh berikut:

**Daftar Prodi di UPJ**

**SIF**

**KOM**

**TIF**

**DKV**

**ARS**

**MGT**

**TSI**

**AKT**

**PSI**

**DPI**

# Analisa Hasil Praktek

Berdasarkan praktek yang telah Anda lakukan, buat kesimpulan:

# Tugas

1. Buat program yang akan membaca isi file teks, kemudian program tersebut menghitung jumlah huruf tertentu (sesuai keinginan user).
2. Buat program pengolahan String yang menggunakan method **startsWith()** pada class String. Mengenai penggunaan method **startsWith()** anda cari informasi-nya pada text book atau internet.

# Appendix 1: Form Data Praktek

Nama : ……………………….

NIM : ……………………….

Tgl : ……………………….

|  |  |
| --- | --- |
| **Penggalan *Source code*** | **Keluaran / *Output*** |
|  |  |
|  |  |
|  |  |