Fondasi Pemrograman

Modul Praktikum

# Topik

Dasar Class dan Object

# Tujuan

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Menggunakan aplikasi BlueJ untuk membuat program Java
2. Membuat class sederhana
3. Membuat object dari suatu class

# 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 JDK dapat diperoleh di;

<http://www.bluej.org>

# Rangkuman/Teori

## [menjelaskan cara penggunaan aplikasi BlueJ]

BlueJ merupakan salah satu aplikasi IDE (Integrated Development Environment) yang dapat digunakan untuk membantu pembuatan program java.

Untuk menjalankan BlueJ dapat dilakukan dengan click icon BlueJ pada menu;

**BlueJ**



Gambar 1. Tampilan Aplikasi BlueJ

### Membuat Project

Sebelum membuat class kita harus membuat project yang akan menampung class-class yang kita buat.

Project 🡪 New Project



Gambar 2. Membuat Project Baru

Pilih lokasi folder tempat menampung project pada **look in:** , kemudian tulis nama project pada **Folder name:** , tekan tombol Create.

### Membuat Class

Selanjutnya untuk membuat class, tekan tombol **New Class** pada panel BlueJ.



Gambar 3. Membuat Class Baru

Untuk masuk ke editor dan mengetik kode program click dua kali pada gambar class (pada contoh gambar di atas gambar class adalah kotak dengan teks Hello.



Gambar 4. Tampilan Code Editor

BlueJ akan membuatkan template untuk contoh pada editor, jika kita tidak ingin menggunakan template contoh tersebut kita dapat di hapus kode-kode yang berada diantara simbol **{** **}** (mulai dari **{** pertama sampai **}** terakhir).



Gambar 5. Tampilan Code Editor, Setelah kode yang tidak diperlukan dihapus.

### [menjelaskan class pada pemrograman java]

Class merupakan suatu rancangan untuk membuat suatu object. “**class**” – pada pemrograman java terdiri dari ***fields*** yang berfungsi untuk menyimpan data dan ***methods*** yang nantinya akan digunakan sebagai instances.

Instruksi untuk membuat class;

***<visibility> class <nama\_class>;***

Keterangan:

**<visibility> :** public 🡪 dapat diakses oleh semua class

 Protected 🡪 hanya dapat diakses oleh class dalam package yang sama.

 Private 🡪 hanya dapat diakses oleh class itu sendiri.

**<nama\_class> :** nama class sesuai peruntukan class

Fieldsdapat dianalogikan seperti halnya sebuah variable. Berikut instruksi untuk membuat field pada class.

***<visibility> <tipe\_data> <nama\_field>;***

Keterangan:

**<visibility> :** public 🡪 dapat diakses oleh semua class

 Protected 🡪 hanya dapat diakses oleh class dalam package yang sama.

 Private 🡪 hanya dapat diakses oleh class itu sendiri.

**<tipe\_data>** : tipe data yang akan digunakan, sesuai peruntukan data yang akan ditampung (int, char, double, String, …dst)

**<nama\_field>** : nama field sesuai peruntukan field.

Methdo dapat dianalogikan seperti halnya sebuah prosedur atau fungsi yang berisi kumpulan instruksi untuk mengolah atau me. Berikut instruksi untuk membuat field pada class.

**<visibility> <tipe\_return> <nama\_method>(<tipe\_data> <par>)**

{

 Statement instruksi …… ;

}

Keterangan:

**<visibility> :** public 🡪 dapat diakses oleh semua class

 Protected 🡪 hanya dapat diakses oleh class dalam package yang sama.

 Private 🡪 hanya dapat diakses oleh class itu sendiri.

**<tipe\_return>** : jika method tidak mengembalikan nilai, maka ditulis **void**.

Jika mengembalikan nilai, maka ditulis tipe data yang akan dikembalikan/dikirim (int, char, double, String, …dst)

**<nama\_method>**: nama method sesuai peruntukan method.

**<tipe\_data>** : tipe data yang akan digunakan, sesuai peruntukan data yang akan ditampung parameter (int, char, double, String, …dst).

**<par>** : nama parameter yang akan menampung nilai.

# 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: Membuat Class

Ikuti langkah-langkah berikut ini untuk membuat program *hello world.* Langkah-langkah membuat program:

|  |
| --- |
| Hello |
| String namaInt usia |
| void isiData(parNama, pasUsia)String sayHello() |

1. Buka BlueJ, buat project latihanObject
2. Buat class sesuai dengan diagram class di atas.
3. Compile class tersebut.



1. Buat instance dari class Hello dengan nama Hello1 pada panel 🡪 click kanan pada class 🡪 New Hello()



1. Buat object dengan nama **hai** pada command line.



1. Jalankan method isiData()
2. Jalankan method sayHello()

## Praktek 2: Membuat Object

1. Buka BlueJ, buat project latihanObject
2. Buat class testHello.
3. Buat object hai dari class Hello pada main method.
4. Buat program selanjutnya untuk mengisi data pada tiap field.
5. Tampilkan hasil dari pemanggilan method sayHello()

# Analisa Hasil Praktek

Berdasarkan praktek yang telah Anda lakukan, buat kesimpulan:

# Tugas

# Appendix 1: Form Data Praktek

Nama : ……………………….

NIM : ……………………….

Tgl : ……………………….

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