



**INSTITUT TEKNOLOGI DEL  
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO  
PROGRAM STUDI SARJANA SISTEM INFORMASI**

**RENCANA TUGAS MAHASISWA**

MATA KULIAH	Nama	Pemrograman Berorientasi Objek
	Kode	12S2203
	Kredit	3 sks
	Semester	IV
	Pengampu	Mario Simaremare (mario@del.ac.id)
TUGAS MAHASISWA	Minggu	1
	Bentuk	Tugas terstruktur (individual & peer)
	Waktu	2 x 60 menit terstruktur
	Judul	Language Fundamentals
SUBCAPAIAN PEMBELAJARAN MATA KULIAH (SubCPMK)		
SCPMK01	[C3] Mampu menulis program pada bahasa pemrograman yang mendukung penerapan pola berpikir berorientasi objek.	
DESKRIPSI TUGAS		
Pada tugas ini mahasiswa akan melakukan <i>environment setup</i> dengan menginstal beberapa <i>software tools</i> yang dipergunakan di sepanjang perkuliahan, melakukan registrasi ke <i>online source control system</i> (github.com), dan membuat repositori penyimpanan artefak masing-masing. Selanjutnya, mahasiswa akan mulai mempelajari hal-hal dasar dalam memprogram pada bahasa Java, seperti: pengenalan struktur program, pengenalan berbagai tipe data primitif dan <i>wrapper class</i> , variabel, array, collection, struktur percabangan, dan perulangan.		
METODE Pengerjaan TUGAS		
Setiap mahasiswa akan secara mandiri melakukan instalasi <i>software tools</i> yang diperlukan. Untuk penggunaan <i>source control</i> akan dipandu oleh asisten.		
BENTUK DAN FORMAT LUARAN		
1. <i>Software tools</i> terinstal dan dapat dipergunakan sebagaimana mestinya. 2. Akun github.com aktif dan telah membuat repository untukpenyimpanan artefak perkuliahan. 3. <i>Source code</i> hasil pelaksanaan tugas dalam Netbeans <i>project</i> yang dikompresi dalam format <b>zip</b> .		
INDIKATOR, KRITERIA DAN BOBOT PENILAIAN		
Untuk kedua luaran akan dinilai menggunakan kriteria dan bobot sebagai berikut: 1. Ketepatan pendeklarasian dan penggunaan variabel, tipe-tipe data primitif, dan operatornya. 2. Ketepatan pendeklarasian dan penggunaan kelas-kelas yang umum digunakan (misalnya <i>String</i> , dan <i>wrapper class</i> ). 3. Ketepatan pendeklarasian dan pengguaan <i>multi valued strctures</i> (array dan <i>collection</i> ). 4. Ketepatan penggunaan kelas dan objek untuk membaca <i>input</i> dan menulis <i>output</i> . 5. Ketepatan penggunaan struktur percabangan ( <i>if-else</i> , <i>switch-case</i> , & <i>ternary</i> ). 6. Ketepatan penggunaan struktur perulangan ( <i>for</i> , <i>while</i> , <i>do-while</i> ).		
PENGUMPULAN		
Tanggal/ waktu	Rabu, 05 Februari 2020/ 21.30	
Tempat	<a href="https://ecourse.del.ac.id/mod/assign/view.php?id=9430">https://ecourse.del.ac.id/mod/assign/view.php?id=9430</a>	
LAIN-LAIN		
Tidak ada		
PUSTAKA		
Utama:		
[1] Herbert Schildt. Java: The Complete Reference. 11 <sup>th</sup> . Oracle Press.		
Pendukung:		
[2] Java SE 8 Documentation (Official) <a href="https://docs.oracle.com/javase/8/docs/api/index.html">https://docs.oracle.com/javase/8/docs/api/index.html</a>		

## Tugas 1. Software Tools Installation

Silakan untuk melakukan instalasi empat *software tools* berikut ini:

1. Git Source Control.
2. SourceTree, a GUI for Git.
3. Java Development Kit 1.8 (64bit).
4. Apache Netbeans (64bit).

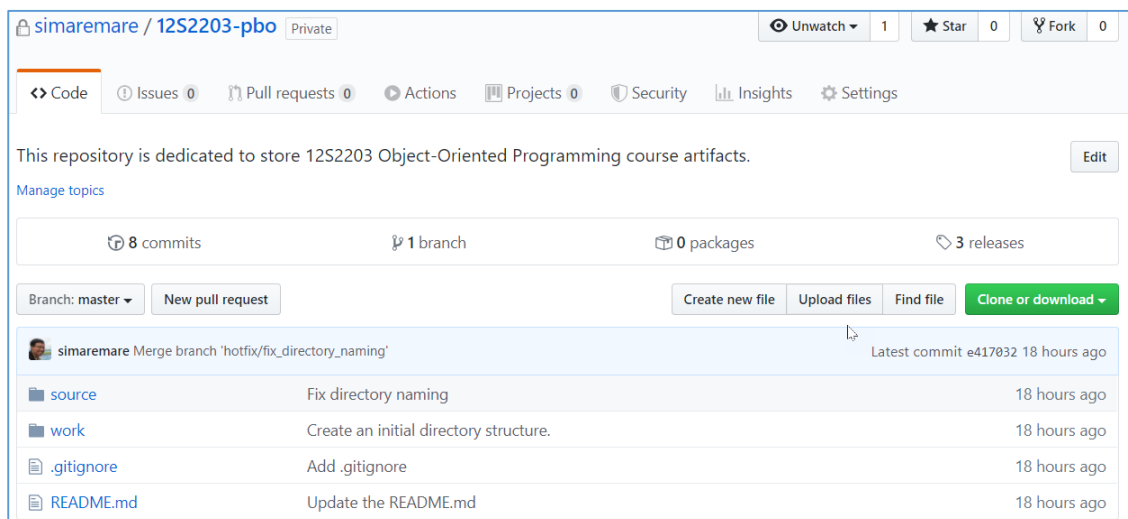
Harap memverifikasi bahwa keempat software tools tersebut telah berfungsi dengan baik.

## Tugas 2. Creating GitHub Account and Course Repository

Silakan untuk membuat akun di <https://github.com/> kemudian secara kolektif mencatat dan mengumpulkan *username* akun tersebut. *Username* diawali dengan simbol *at* (@).

Selanjutnya, untuk setiap kelompok, membuat satu repository penyimpanan artefak perkuliahan dengan nama **12S2203-pbo**. Pada *repository* tersebut, sertakan rekan kelompok anda sebagai salah satu *contributor* melalui menu **Settings > Collaborators**. Selain itu, sertakan juga akun dengan username **submissionrobot** sebagai *contributor* pada *repository* tersebut.

*Clone repository* anda dengan menggunakan SourceTree. Gunakan *directory-structure.zip* sebagai struktur dasar *repository* anda. Dengan demikian, *homepage* dari *repository* akan tampak seperti pada Gambar 1. *Commit* struktur direktori anda dengan *message* "Create directory structure" lalu push ke *origin server* (GitHub.com).



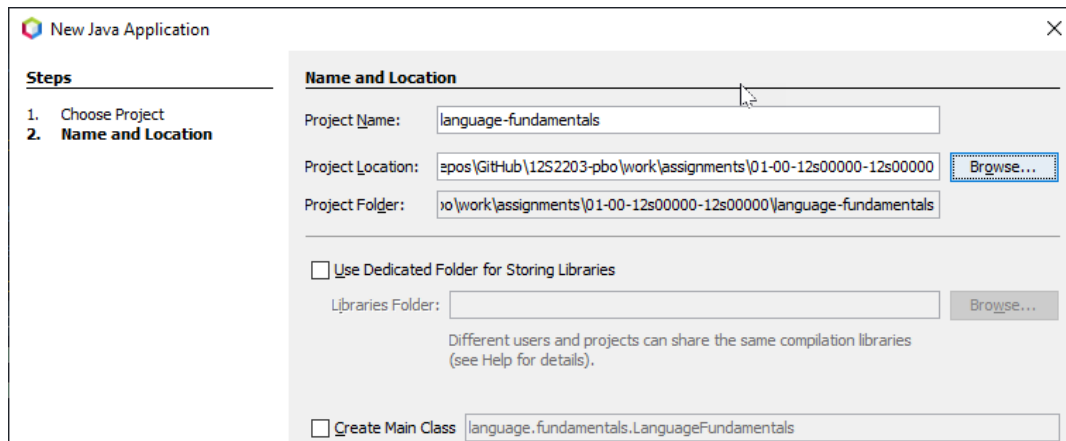
Gambar 1. Tampilan *repository homepage* pada GitHub

## Tugas 3. Creating NetBeans Project

### Working in peers.

Buatlah sebuah baru dengan format penamaan `xx-yy-nim1-nim2` pada `work\assignments\`, `xx` adalah nomor tugas dalam bentuk dua digit angka, sementara `yy` adalah nomor kelompok anda juga dalam bentuk dua digit angka. `nim1` dan `nim2` adalah NIM anggota kelompok dalam *lowercase* diurutkan dari yang lebih kecil secara alfabetis. Sebagai contoh, `01-06-12s18020-12s18050` adalah nama direktori untuk kelompok tugas pertama dari kelompok 06 yang beranggotakan mahasiswa dengan NIM 12s18020 dan 12s18050.

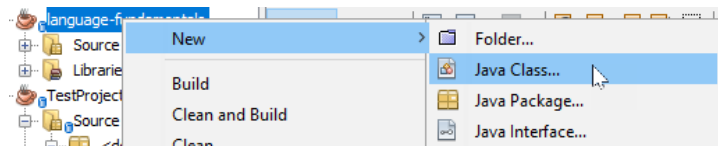
Buatlah sebuah NetBeans *project* (*Java with Ant*) dengan nama `language-fundamentals` pada direktori yang baru saja anda buat pada tahap sebelumnya Gambar 2. Perhatikan bahwa anda tidak perlu membuat *entry point* (*Main class*) pada saat membuat *project*. Anda akan membuatnya secara manual. Setelah selesai, silakan melakukan *committing* pekerjaan anda dengan message "Create language-fundamentals project" dan *pushing repository* ke *origin server* (GitHub.com).



Gambar 2. Pembuatan *project* (Java with Ant)

## Tugas 4. Creating The Entry Point

Program *entry point* adalah 'pintu' awal eksekusi program atau bagian yang akan dieksekusi pada saat program dijalankan. Pada Java, *entry point* ditandai dengan sebuah *main method* yang ditulis pada sebuah *class*.



Gambar 3. New > Java Class

Buatlah sebuah *class* baru dengan nama `Main` (Gambar 3) pada *project* anda. Tuliskan identitas penulis dengan `@author` *annotation* pada bagian atas *Main class* dengan format `yy-nim nama`. Contoh dapat dilihat pada Kode 1.

```
/**
 *
 * @author 06-12s18050 Elsa Marbun
 * @author 06-12S18020 Dita Sihombing
 */
public class Main {
}
```

Kode 1. *Main class*

Selanjutnya definisikan *main method* berikut pada *Main class* (Kode 2).

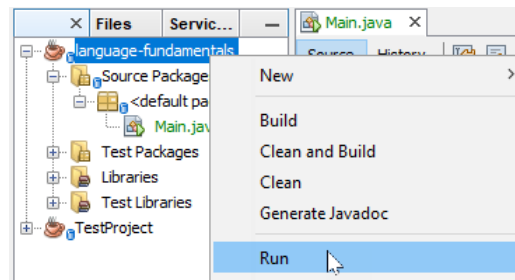
```
/**
 *
 * @author 06-12s18050 Elsa Marbun
 * @author 06-12S18020 Dita Sihombing
 */
public class Main {

    /* program entry point */
    public static void main(String[] _args) {
        System.out.println("Hello, world!");
    }
}
```

Kode 2. *main method*

Untuk mengeksekusi program, klik-kanan pada *project* lalu pilih **Run**. Anda dapat melihat hasil eksekusi pada panel *Output*. Pada konteks ini, proses kompilasi dan eksekusi dilakukan secara

sekuensial dan diotomasi oleh NetBeans. Silakan mencoba untuk melakukan kompilasi dan eksekusi secara manual melalui *Command Line Interface* (CMD).



Gambar 4. Mengeksekusi program

## Tugas 5. Declaring and Manipulating Primitive Type Variables

Berikutnya, anda akan melakukan pendeklarasian variabel dari berbagai tipe data primitif dan memanipulasi nilainya. Pada bagian akhir dari *main method*, tambahkan *snipet* Kode 3 berikut.

```
// declaring and manipulating primitive variables
int gear = 1;
boolean result = true;
char capitalC = 'C';
byte b = 100;

System.out.println(gear + ", " + result + ", " + capitalC + ", " + b);

gear++;
result = !result;
capitalC += 4;

System.out.println(gear + ", " + result + ", " + capitalC + ", " + b);

// 26 in hexadecimal and binary
int hexVal = 0x1a;
int binVal = 0b11010;

System.out.println(hexVal + ", " + binVal);
```

Kode 3. Deklarasi variabel dan manipulasi nilai

Eksekusi program dan amati hasilnya.

## Tugas 6. Declaring and Manipulating Objects

Berikutnya anda akan melakukan deklarasi dan instansiasi objek dari kelas-kelas yang umum digunakan seperti *String* dan *wrapper classes*. Pada bagian akhir dari *main method*, tambahkan *snipet* Kode 4 berikut.

```
// instantiating and manipulating objects
String name = "Wiro Sableng";
Short gearObject = 1;
Boolean resultObj = true;

System.out.println(name + ", " + gearObject + ", " + resultObj);

name = name.toUpperCase();
gearObject++;
resultObj = (result == false) || !resultObj;

System.out.println(name + ", " + gearObject + ", " + resultObj);
```

Kode 4. Deklarasi objek, instansiasi, dan manipulasi nilai

Eksekusi program dan amati hasilnya.

## Tugas 7. Using Array, Collections, and Loop

*Array* dan *collections* digunakan untuk menyimpan *multivalue* dalam satu objek. Pada bagian ini anda akan menggunakan array dan collection sederhana. Tambahkan *snipet* Kode 5 pada akhir dari *main method*.

```
// array, collections, and loop
int[][] numbers = new int[10][10];

for (int i = 0; i < numbers.length; ++i) {
    for (int j = 0; j < numbers[i].length; ++j) {
        numbers[i][j] = i + j;
    }
}

for (int i = 0; i < numbers.length; ++i) {
    for (int j = 0; j < numbers[i].length; ++j) {
        System.out.print(numbers[i][j] + " ");
    }
    System.out.println();
}
```

Kode 5. Deklarasi objek, instansiasi, dan manipulasi nilai

Eksekusi program dan amati hasilnya.

**Challenge:** modifikasi Kode 5 sehingga menggunakan *ArrayList*, salah satu implementasi dari *collection* yang mengaplikasikan konsep *list*.

## Submission

Silakan untuk mengkompresi direktori kerja anda, misal 01-06-12s18020-12s18050 dengan format kompresi **zip**. Kemudian upload ke *ecourse* oleh salah satu anggota kelompok.

Selanjutnya, *commit* pekerjaan anda dan mem-push ke *origin server* (GitHub.com).

- E O F -