

PRAKTIKUM JAVA I

MODUL 1

MENGENAL JAVA

TUJUAN

- Memahami cara kompilasi dan eksekusi pada Java
- Memahami sintaks dasar Java
- Memahami penggunaan Git untuk mengumpulkan tugas

ALAT DAN BAHAN

- Notepad / Notepad++ / text editor lainnya
- IntelliJ IDEA (<https://www.jetbrains.com/idea/download/>)
- JDK (<http://jdk.java.net/14/>)
- Git (<https://git-scm.com/>)
- GitHub (<https://github.com/>)

DASAR TEORI

Java memiliki tagline Write Once, Run Anywhere (WORA), hal ini dimungkinkan karena Java berjalan pada Java Virtual Machine (JVM), sehingga bisa dijalankan di berbagai macam platform. Sintaks mirip dengan C dan C++ dan dibangun oleh James Gosling pada tahun 1995 sewaktu masih menjadi karyawan Sun Microsystem yang belakangan diakuisisi oleh Oracle.

Kegiatan:

A. Output Hello World

1. Ekstrak Java pada folder yang mudah dicari, misalkan:

```
C:\Coding\openjdk
```

2. Tambahkan alamat

```
C:\Coding\openjdk\bin
```

pada System Variable path pada Environment Variables

3. Periksa versi Java sekaligus memeriksa apakah folder jdk sudah berhasil ditambahkan dengan membuka command prompt dan ketikkan perintah

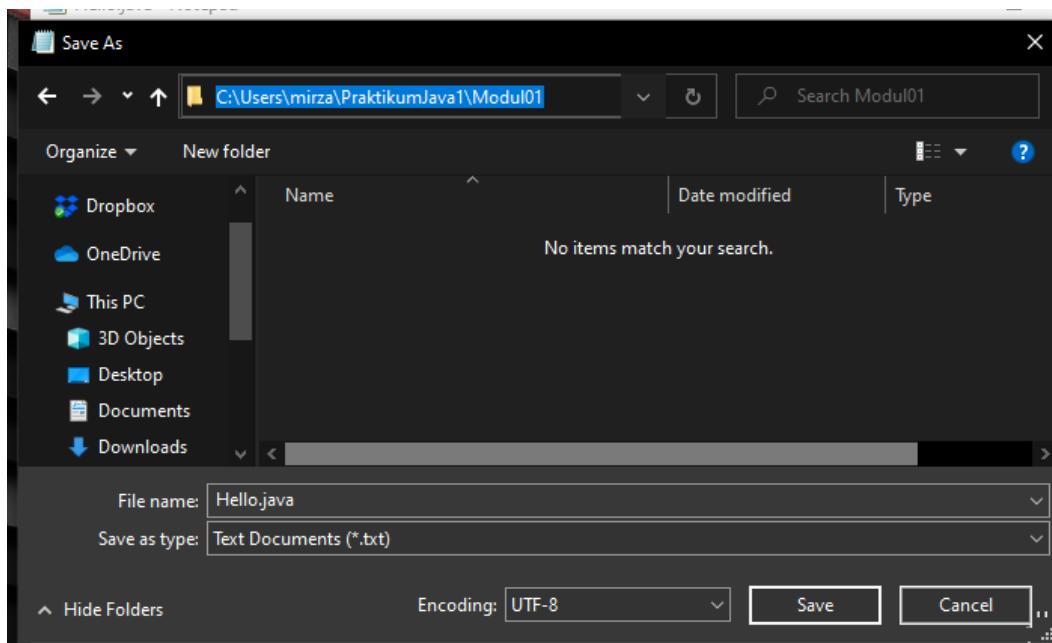
```
java -version
```

4. Buat sebuah file teks menggunakan text editor, isi dengan kode sebagai berikut, pastikan huruf besar kecilnya sama karena Java sensitif terhadap huruf besar dan kecil

```
public class Hello{
    public static void main (String [] args){
        System.out.println("Hello World !!");
    }
}
```

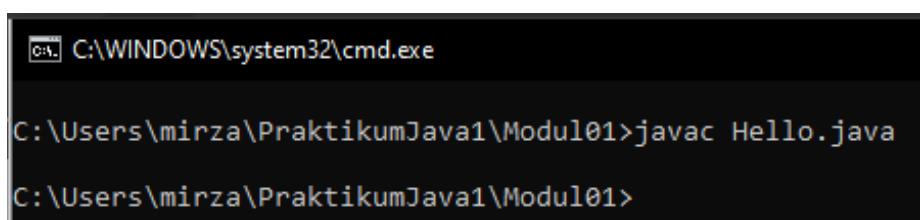
- Simpan dengan nama file `Hello.java`, pastikan nama file sama dengan nama class. Kemudian simpan dalam folder

`..\PraktikumJava1\Modul01\`



- Buka command prompt pada lokasi file `Hello.java`, compile file `Hello.java` dengan perintah

`javac Hello.java`

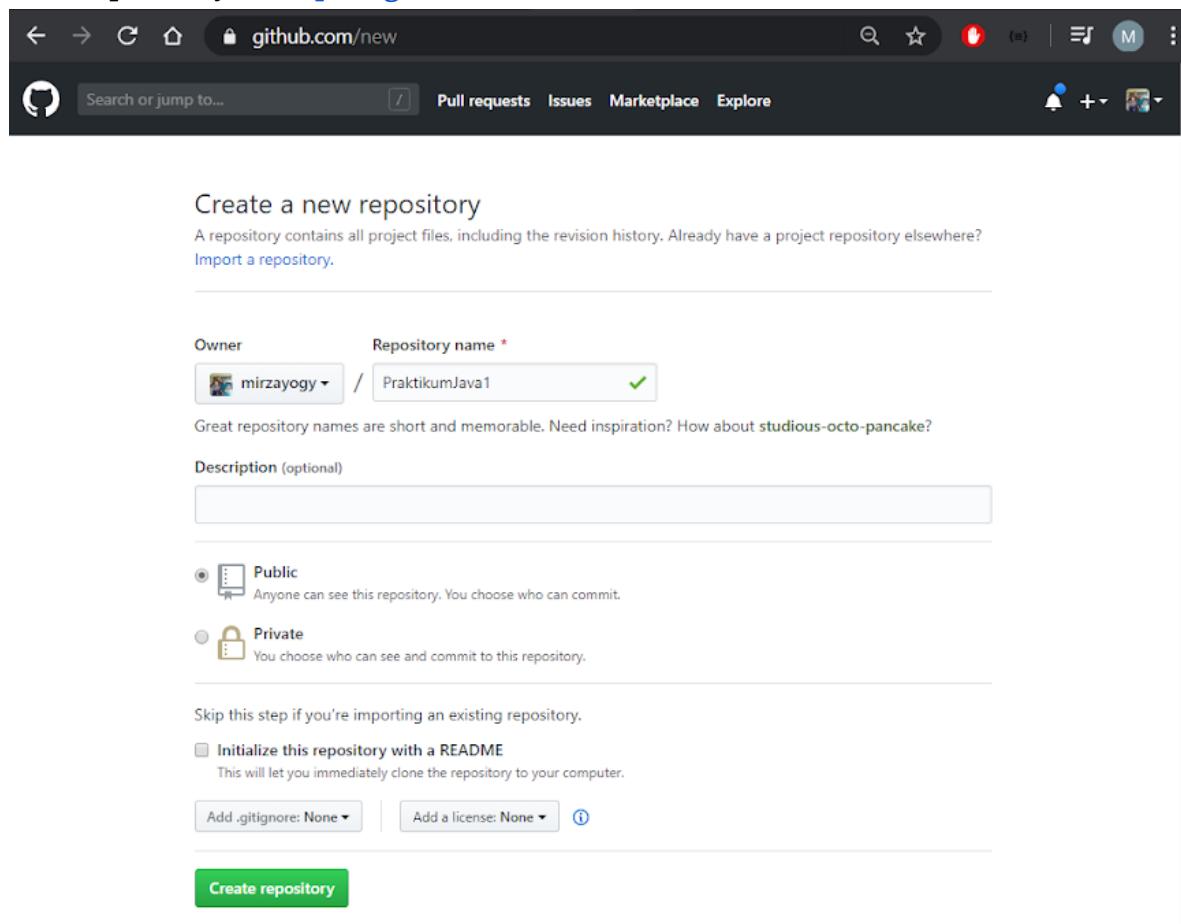


- Perintah `javac` akan menghasilkan sebuah file baru yang merupakan hasil kompilasi yaitu file `Hello.class`.
- Eksekusi file `Hello.class` dengan perintah

`java Hello`

B. Remote GitHub

1. Buat repository di <https://github.com>



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner: mirzayogy / Repository name: PraktikumJava1

Description (optional):

Visibility: Public (Anyone can see this repository. You choose who can commit.)

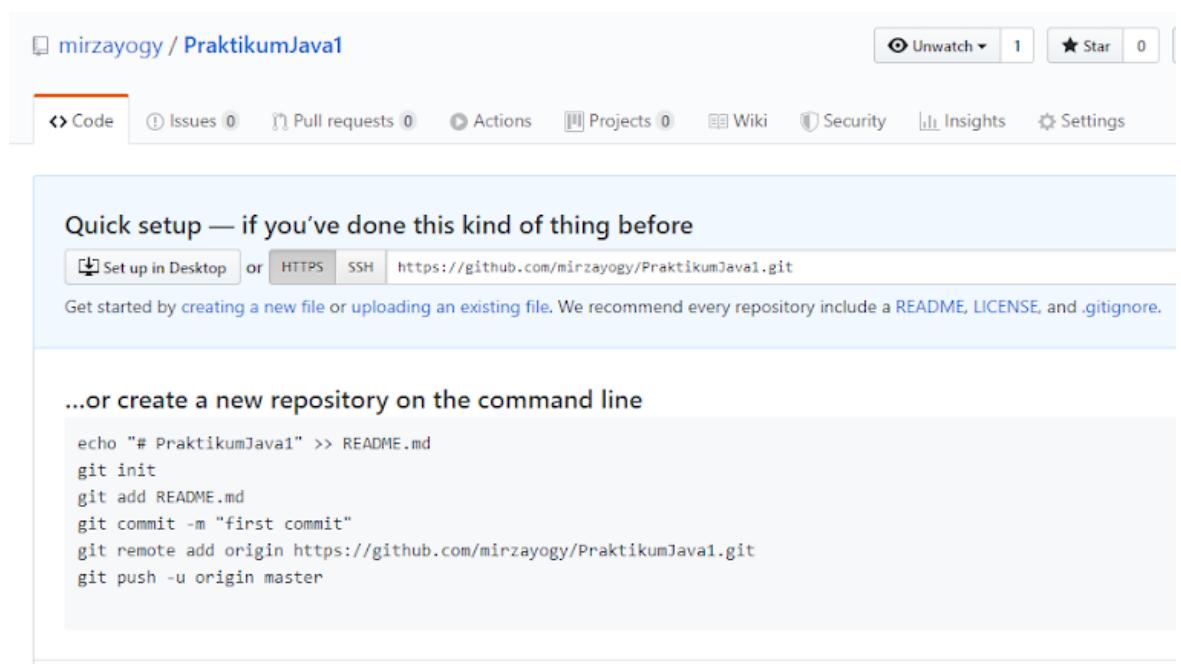
Private (You choose who can see and commit to this repository.)

Skip this step if you're importing an existing repository.

Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None | Add a license: None | ⓘ

Create repository



mirzayogy / PraktikumJava1

Unwatch 1 | Star 0

Code | Issues 0 | Pull requests 0 | Actions | Projects 0 | Wiki | Security | Insights | Settings

Quick setup — if you've done this kind of thing before

[Set up in Desktop](#) or [HTTPS](#) [SSH](#) <https://github.com/mirzayogy/PraktikumJava1.git>

Get started by creating a new file or uploading an existing file. We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# PraktikumJava1" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/mirzayogy/PraktikumJava1.git
git push -u origin master
```

2. Buka command prompt, posisikan pada folder berikut

```
C:\Users\mirza\PraktikumJava1\Modul01>cd..  
  
C:\Users\mirza\PraktikumJava1>
```

3. Jalankan perintah berikut

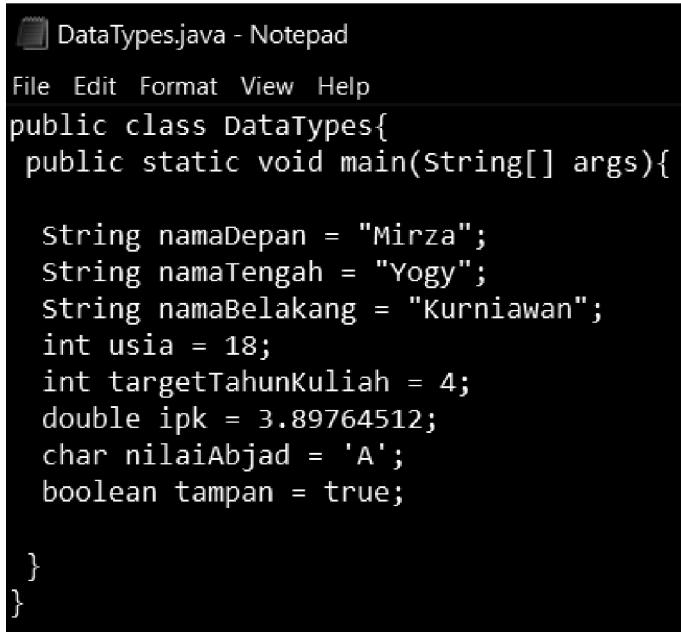
- git init (menginisiasi repo)
- git add . (perhatikan ada titik)
- git commit -m "First commit"
- git remote add origin
<https://github.com/username/PraktikumJava1.git>
- git push -u origin master

```
C:\WINDOWS\system32\cmd.exe  
C:\Users\mirza\PraktikumJava1\Modul01>cd..  
  
C:\Users\mirza\PraktikumJava1>git init  
Initialized empty Git repository in C:/Users/mirza/PraktikumJava1/.git/  
  
C:\Users\mirza\PraktikumJava1>git add .  
  
C:\Users\mirza\PraktikumJava1>git commit -m "First commit"  
[master (root-commit) dd0e6ee] First commit  
 2 files changed, 5 insertions(+)  
  create mode 100644 Modul01/Hello.class  
  create mode 100644 Modul01/Hello.java  
  
C:\Users\mirza\PraktikumJava1>git remote add origin https://github.com/mirzayogy/PraktikumJava1.git  
  
C:\Users\mirza\PraktikumJava1>git push -u origin master  
Enumerating objects: 5, done.  
Counting objects: 100% (5/5), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (4/4), done.  
Writing objects: 100% (5/5), 675 bytes | 225.00 KiB/s, done.  
Total 5 (delta 0), reused 0 (delta 0)  
To https://github.com/mirzayogy/PraktikumJava1.git  
 * [new branch]      master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.  
  
C:\Users\mirza\PraktikumJava1>
```

4. Perhatikan repository di GitHub, pastikan sudah terdapat file yang sesuai

C. Deklarasi Tipe Data

- Buat file DataTypes simpan ke dalam folder Modulo1



```

DataTypes.java - Notepad

File Edit Format View Help
public class DataTypes{
    public static void main(String[] args){

        String namaDepan = "Mirza";
        String namaTengah = "Yogy";
        String namaBelakang = "Kurniawan";
        int usia = 18;
        int targetTahunKuliah = 4;
        double ipk = 3.89764512;
        char nilaiAbjad = 'A';
        boolean tampan = true;

    }
}

```

- Lakukan compile (`javac DataTypes.java`) untuk memeriksa kesalahan pada program.
- Tambahkan perubahan pada repository dengan perintah

`git add .` (perhatikan ada titik)

- Tandai commit/checkpoint dengan perintah

`git commit -m "deklarasi variabel dan tipe data"`

- Push ke remote repository (online)

`git push -u origin master`

D. Output

Perintah output yang biasa dipakai pada Java adalah

```
System.out.println();
```

Perhatikan bahwa huruf S pada System adalah huruf besar. Selain itu juga bisa menggunakan pesan error dengan perintah

```
System.err.println();
```

```

public class DataTypes{
    public static void main(String[] args){

        String namaDepan = "Mirza";
        String namaTengah = "Yogy";
        String namaBelakang = "Kurniawan";
        int usia = 18;
        int targetTahunKuliah = 4;
        double ipk = 3.89764512;
        char nilaiAbjad = 'A';
        boolean tampan = true;

        System.out.println("Nama depan : " + namaDepan);
        System.out.println("Nama tengah : " + namaTengah);
        System.out.println("Nama belakang : " + namaBelakang);
        System.out.println("Usia : " + usia);
        System.out.println("Target Kuliah : " + targetTahunKuliah + " tahun");
        System.out.println("IPK : " + ipk);
        System.out.println("Nilai PBO : " + nilaiAbjad);
        System.out.println("Tampan : " + tampan);
    }
}

```

Compile, Add, dan Commit dengan nama “Output dengan sout”

Output juga bisa dilakukan dengan menggunakan MessageDialog dari Swing dengan menambahkan beberapa perintah berikut

```

import javax.swing.JOptionPane;

public class DataTypes{
    public static void main(String[] args){

        String namaDepan = "Mirza";
        String namaTengah = "Yogy";
        String namaBelakang = "Kurniawan";
        int usia = 18;
        int targetTahunKuliah = 4;
        double ipk = 3.89764512;
        char nilaiAbjad = 'A';
        boolean tampan = true;

        System.out.println("Nama depan : " + namaDepan);
        System.out.println("Nama tengah : " + namaTengah);
        System.out.println("Nama belakang : " + namaBelakang);
        System.out.println("Usia : " + usia);
        System.out.println("Target Kuliah : " + targetTahunKuliah + " tahun");
        System.out.println("IPK : " + ipk);
        System.out.println("Nilai PBO : " + nilaiAbjad);
        System.out.println("Tampan : " + tampan);

        JOptionPane.showMessageDialog(null,"Hai, " + namaDepan + namaTengah + namaBelakang);
    }
}

```

Compile, Add, Commit dengan nama “Output dengan MessageDialog”

E. Input dengan Scanner

Scanner adalah “Class” yang digunakan untuk menerima inputan dari user, untuk bisa menggunakan class tersebut maka perlu dibuat objeknya terlebih dahulu untuk kemudian dijalankan “method” yang sesuai dengan kebutuhan inputnya.

1. Tambahkan import

```
import java.util.Scanner;
```

2. Buat objek dari Scanner di dalam blok main

```
Scanner scanner = new Scanner(System.in);
```

Perhatikan bahwa `scanner` dengan huruf S kecil merupakan objek, berbeda dengan `Scanner` dengan huruf S besar yang merupakan class-nya

3. Panggil method dari `scanner` sesuai dengan tipe datanya:

```
namaDepan = cin.nextLine();  
usia = cin.nextInt();  
ipk = cin.nextDouble();  
nilaiAbjad = cin.next().charAt(0);  
tampan = cin.nextBoolean();
```

4. Input bisa dilakukan pada saat kursor berkedip pada command prompt, sebaiknya diberikan tambahan tulisan yang memperjelas sehingga file `DataTypes.java` menjadi

```

import javax.swing.JOptionPane;
import java.util.Scanner;

public class DataTypes{
    public static void main(String[] args){

        Scanner scanner = new Scanner(System.in);

        String namaDepan = "Mirza";
        String namaTengah = "Yogy";
        String namaBelakang = "Kurniawan";
        int usia = 18;
        int targetTahunKuliah = 4;
        double ipk = 3.89764512;
        char nilaiAbjad = 'A';
        boolean tampan = true;

        System.out.print("Input Nama Depan : ");
        namaDepan = scanner.nextLine();
        System.out.print("Input Usia : ");
        usia = scanner.nextInt();
        System.out.print("Input IPK: ");
        ipk = scanner.nextDouble();
        System.out.print("");
        System.out.print("Input Nilai Abjad : ");
        nilaiAbjad = scanner.next().charAt(0);
        System.out.print("Tampan? : ");
        tampan = scanner.nextBoolean();

        System.out.println("===== OUTPUT =====");
        System.out.println("Nama depan : " + namaDepan);
        System.out.println("Nama tengah : " + namaTengah);
        System.out.println("Nama belakang : " + namaBelakang);
        System.out.println("Usia : " + usia);
        System.out.println("Target Kuliah : " + targetTahunKuliah + " tahun");
        System.out.println("IPK : " + ipk);
        System.out.println("Nilai PBO : " + nilaiAbjad);
        System.out.println("Tampan : " + tampan);

        JOptionPane.showMessageDialog(null,"Hai, " + namaDepan + namaTengah + namaBelakang);

    }
}

```

Add dan Commit dengan nama “Input dengan Scanner”

Input juga bisa dilakukan dengan menggunakan MessageDialog dari Swing dengan perintah berikut

```
namaBelakang = JOptionPane.showInputDialog("Nama Belakang : ");
```

Add dan Commit dengan nama “Input dengan InputDialog”

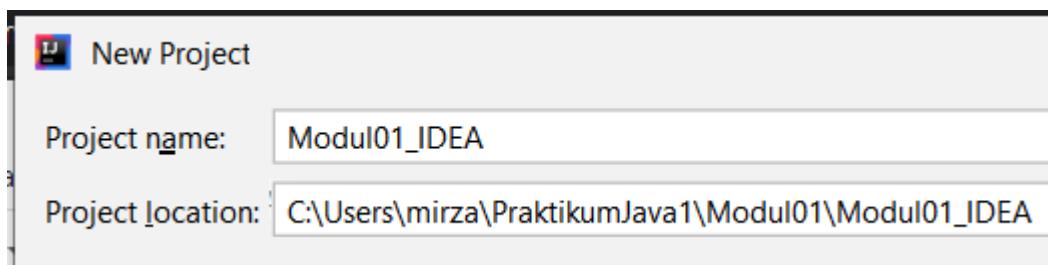
push remote repository dengan perintah

```
git push -u origin master
```

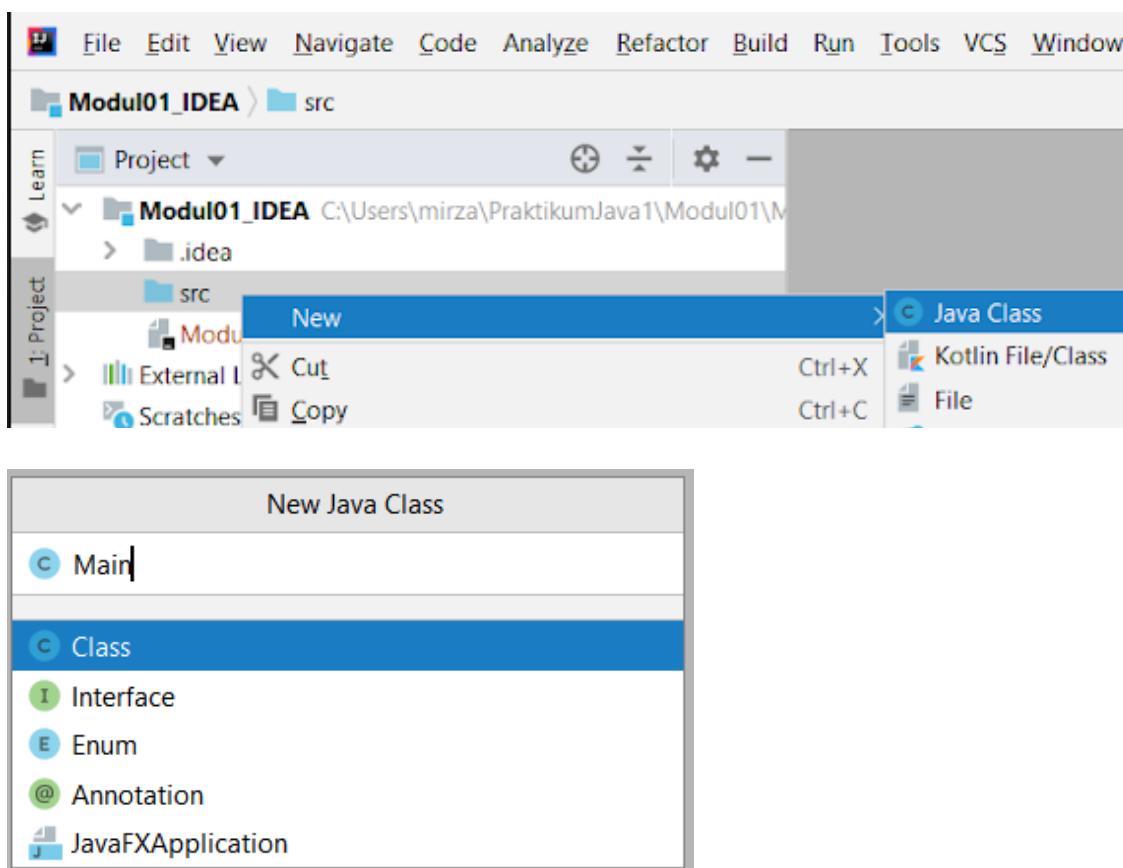
F. IntelliJ IDEA

IntelliJ IDEA adalah salah satu Integrated Development Environment (IDE) untuk mengembangkan aplikasi dalam bahasa Java. Aplikasi ini memiliki fitur modern yang pada dampaknya memerlukan spesifikasi komputer yang cukup tinggi. Sebagian besar kegiatan praktikum ini bisa saja dikerjakan menggunakan notepad tanpa menggunakan IDEA jika komputer terasa terlalu berat.

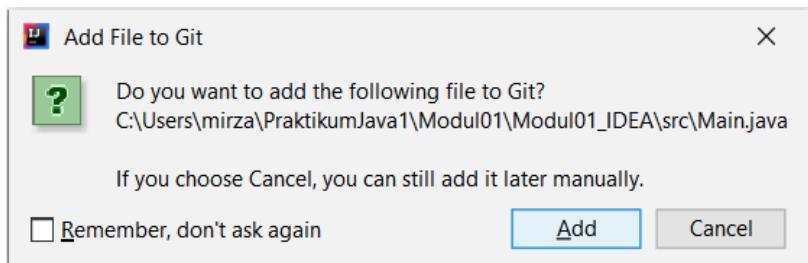
1. Buat Project Baru
2. Pilih Java -> Next -> Next
3. Buat Project dengan nama Modulo1_IDEA letakkan di dalam folder Modulo1



4. Buat class baru Main



5. Karena kita membuat project di dalam repository lokal maka IDEA menawarkan untuk Add to Git, pilih saja Add

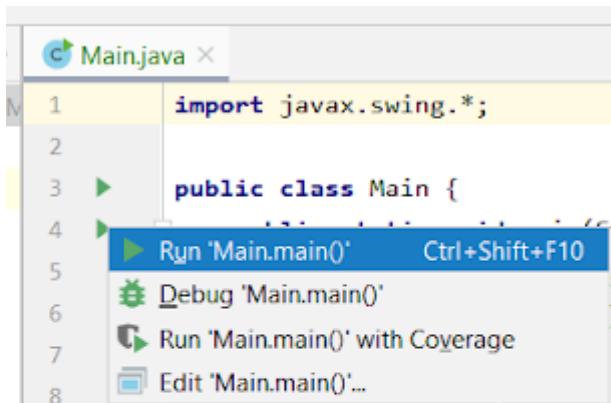


6. Bisa ditulis kembali variabel yang kita buat sebelumnya

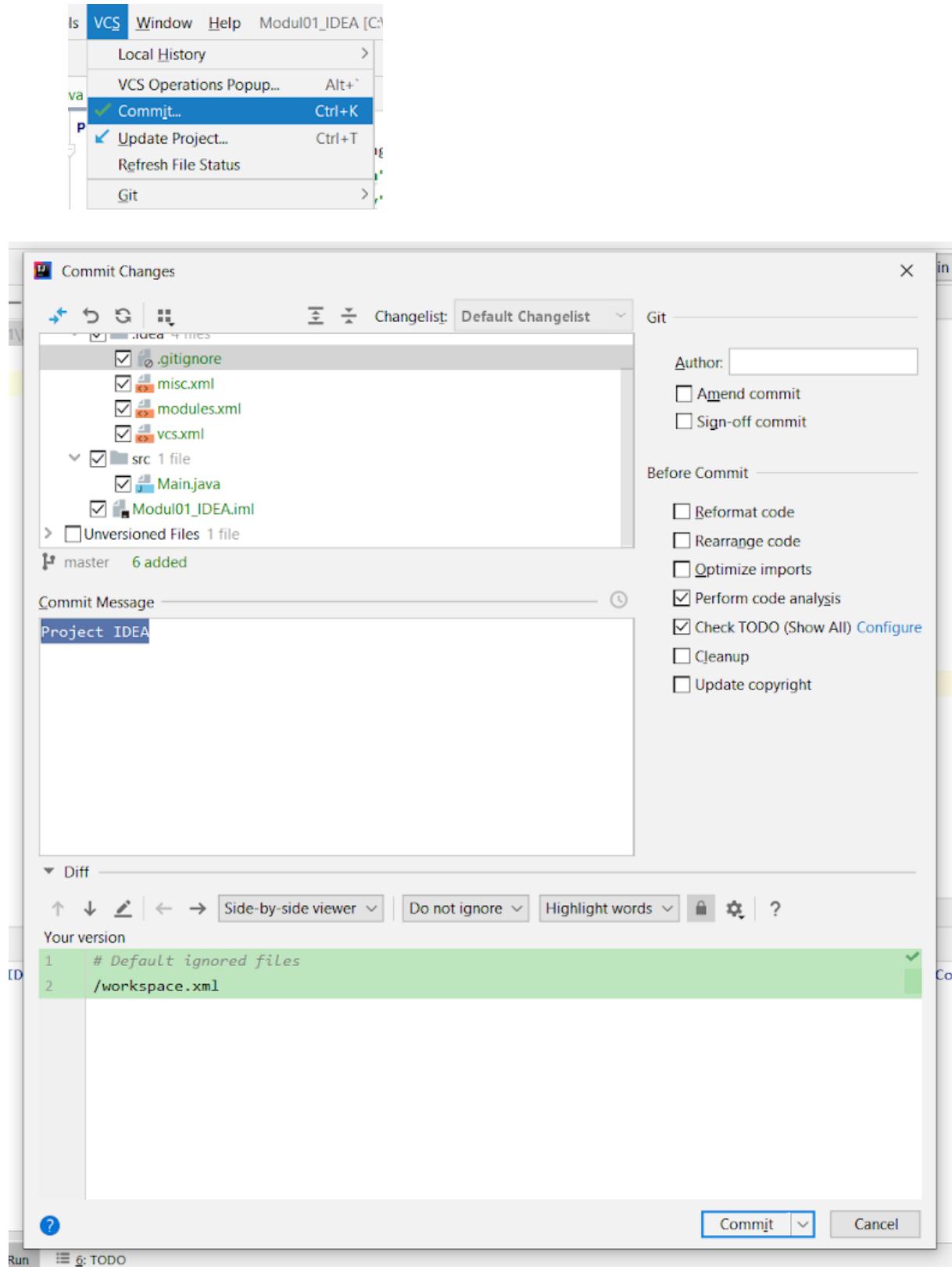
```

1  public class Main {
2      public static void main(String[] args) {
3          String namaDepan = "Mirza";
4          String namaTengah = "Yogy";
5          String namaBelakang = "Kurniawan";
6          int usia = 18;
7          int targetTahunKuliah = 4;
8          double ipk = 3.89764512;
9          char nilaiAbjad = 'A';
10         boolean tampan = true;
11
12         System.out.println(namaDepan + namaTengah + namaBelakang);
13     }
14 }
```

7. Run Aplikasi



8. Commit dengan pesan “Project IDEA”



G. Operasi String

- Tambahkan pembuatan String dari char Array

```

15
16     char[] uniskaChar = {'U','N','I','S','K','A'};
17     String uniskaString = new String(uniskaChar);
18     System.out.println(uniskaString);
19

```

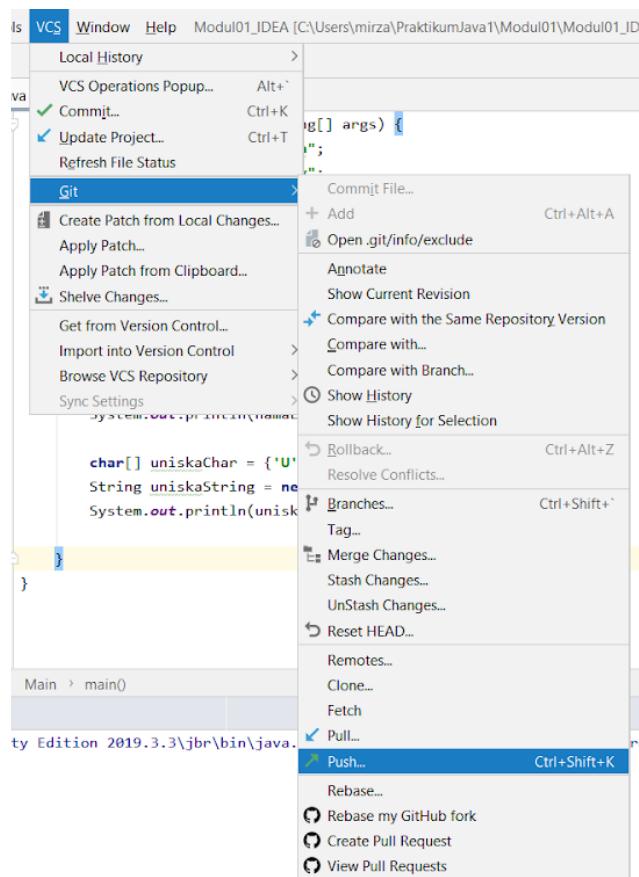
- Commit dengan pesan “Membuat String dari Array Char”
- Tambahkan operasi String lainnya

```

23     String namaLengkap = namaDepan + " " +
24         namaTengah + " " + namaBelakang;
25     System.out.println(namaLengkap.length());
26     System.out.println(namaLengkap.indexOf("NIS"));
27     System.out.println(namaLengkap.substring(5));
28     System.out.println(namaLengkap.substring(5,12));
29     System.out.println(namaLengkap.replace( target: "Yog", replacement: "Barr"));
30     System.out.println(namaLengkap.toUpperCase());
31     System.out.println(namaLengkap.toLowerCase());
32     System.out.println(namaLengkap.charAt(6));
33
34     String[] namaArray = namaLengkap.split( regex: " " );
35
36     for(String nama : namaArray){
37         System.out.println(nama);
38     }

```

- Run dan perhatikan hasil dari tiap perintah (method) yang dijalankan pada tipe String.
- Commit dengan pesan “Operasi Tipe String”
- Push ke GitHub dengan menekan tombol Ctrl + Shift + K



PRAKTIKUM JAVA I

MODUL 2

OOP JAVA

TUJUAN

- Memahami cara menerapkan konsep Object Oriented Programming (OOP) pada Java

ALAT DAN BAHAN

- IntelliJ IDEA (<https://www.jetbrains.com/idea/download/>)

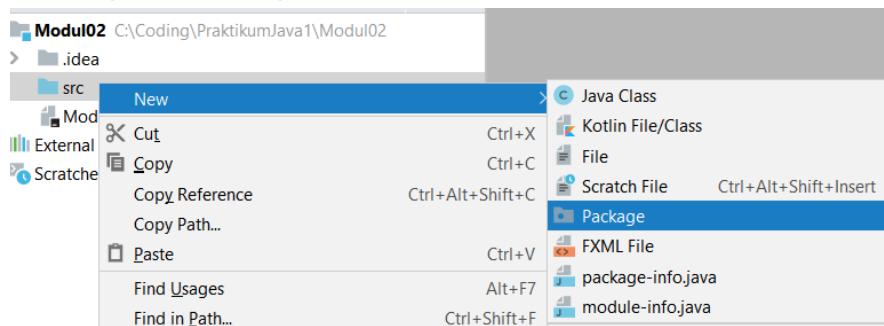
DASAR TEORI

Pemrograman Berorientasi Objek / Object Oriented Programming, adalah paradigma dalam pemrograman yang menggunakan objek nyata ke dalam penyusunan program, sehingga proses perancangan bisa berlangsung dengan lebih mudah dipahami. Java merupakan bahasa pemrograman yang murni OOP.

Kegiatan:

A. Membuat Class, Atribut, dan Objek

1. Buat package baru dengan nama “model”



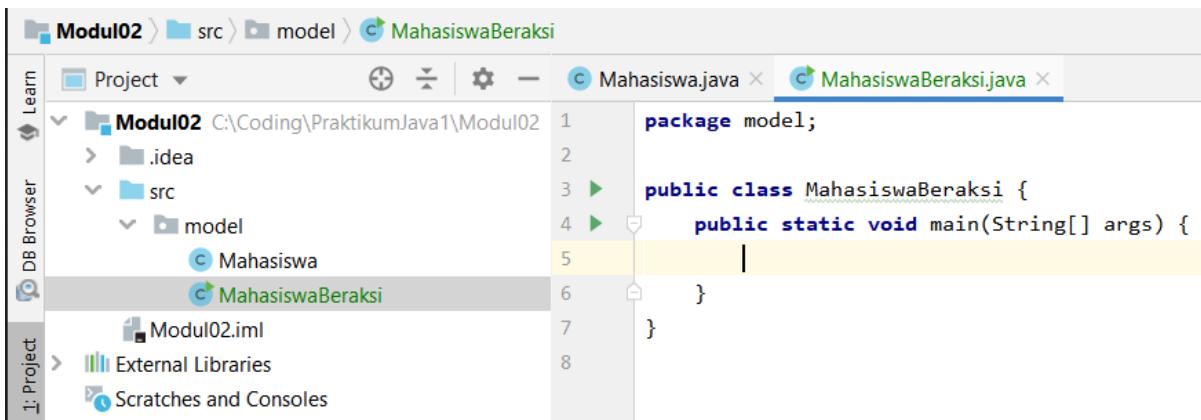
2. Buat class baru di dalamnya dengan nama “Mahasiswa”

```

1 package model;
2
3 import java.util.Date;
4
5 public class Mahasiswa {
6     String npm;
7     String nama;
8     Date tanggalLahir;
9 }

```

3. Buat class baru lainnya dengan nama “MahasiswaBeraksi”



4. Buat objek dari mahasiswa dengan mengetikkan perintah berikut

```

package model;

import java.util.Date;

public class MahasiswaBeraksi {
    public static void main(String[] args) {
        Mahasiswa mahasiswa = new Mahasiswa();
        mahasiswa.npm = "19639999";
        mahasiswa.nama = "Sembilu";
        mahasiswa.tanggalLahir = new Date();
    }
}

```

Baris 7 merupakan perintah yang membuat Objek dari sebuah Class atau bisa disebut “Instance”. Objek yang dibuat memiliki atribut sebagaimana yang sudah ditentukan di Class Mahasiswa. Jadi, tanpa perlu dideklarasikan lagi, atribut itu dimiliki oleh objek mahasiswa. Oleh karenanya Class sering disebut sebagai “cetakan” atau “blueprint” karena atribut hanya dirancang saja, sedangkan nilainya ditentukan di objek.

Perhatikan untuk penulisan nama Class gunakan awalan huruf besar, sedangkan nama objek gunakan awalan huruf kecil.

5. Coba tampilkan nilai atribut, lanjutkan dengan membuat objek kedua



```

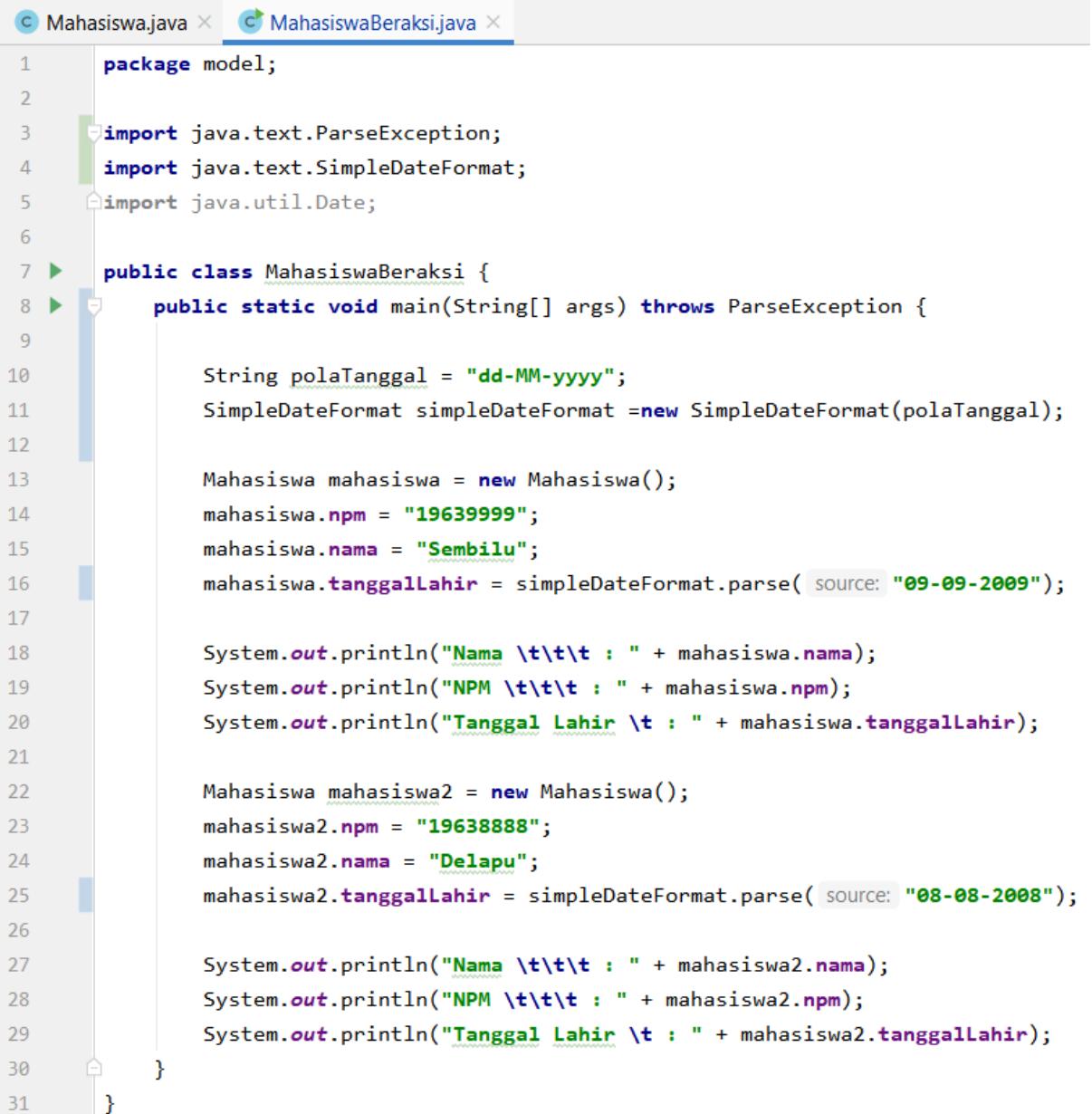
2
3     import java.util.Date;
4
5     public class MahasiswaBeraksi {
6         public static void main(String[] args) {
7             Mahasiswa mahasiswa = new Mahasiswa();
8             mahasiswa.npm = "19639999";
9             mahasiswa.nama = "Sembilan";
10            mahasiswa.tanggalLahir = new Date();
11
12            System.out.println("Nama \t\t\t : " + mahasiswa.nama);
13            System.out.println("NPM \t\t\t : " + mahasiswa.npm);
14            System.out.println("Tanggal Lahir \t : " + mahasiswa.tanggalLahir);
15
16            Mahasiswa mahasiswa2 = new Mahasiswa();
17            mahasiswa2.npm = "19638888";
18            mahasiswa2.nama = "Delapan";
19            mahasiswa2.tanggalLahir = new Date();
20
21            System.out.println("Nama \t\t\t : " + mahasiswa2.nama);
22            System.out.println("NPM \t\t\t : " + mahasiswa2.npm);
23            System.out.println("Tanggal Lahir \t : " + mahasiswa2.tanggalLahir);
24        }
25    }

```

Perintah “\t” merupakan escape sequence yang menandakan tabulasi/tab pada teks. Selain “\t” Java juga mengenal “\n” sebagai perintah untuk mengganti baris.

Perhatikan untuk objek kedua diberikan nama “mahasiswa2”. Penamaan objek ini bebas saja sebagaimana penamaan variabel pada umumnya, namun tetap berlaku aturan tidak boleh ada “spasi” dan tidak boleh dimulai selain abjad dan garis bawah “_”.

6. Berikutnya untuk membuat tipe data Date, bisa digunakan SimpleDateFormat dan parse. SimpleDateFormat juga merupakan Class sehingga di-instance terlebih dahulu dan tentukan format tanggal yang ingin ditampilkan atau dinput. Perintah parse mungkin menyebabkan error ParseException sehingga untuk sementara kita throws dulu saja, perhatikan deklarasi method main yang berubah.



```

1 package model;
2
3 import java.text.ParseException;
4 import java.text.SimpleDateFormat;
5 import java.util.Date;
6
7 public class MahasiswaBeraksi {
8     public static void main(String[] args) throws ParseException {
9
10         String polaTanggal = "dd-MM-yyyy";
11         SimpleDateFormat simpleDateFormat =new SimpleDateFormat(polaTanggal);
12
13         Mahasiswa mahasiswa = new Mahasiswa();
14         mahasiswa.npm = "19639999";
15         mahasiswa.nama = "Sembilan";
16         mahasiswa.tanggalLahir = simpleDateFormat.parse( source: "09-09-2009");
17
18         System.out.println("Nama \t\t\t : " + mahasiswa.nama);
19         System.out.println("NPM \t\t\t : " + mahasiswa.npm);
20         System.out.println("Tanggal Lahir \t : " + mahasiswa.tanggalLahir);
21
22         Mahasiswa mahasiswa2 = new Mahasiswa();
23         mahasiswa2.npm = "19638888";
24         mahasiswa2.nama = "Delapan";
25         mahasiswa2.tanggalLahir = simpleDateFormat.parse( source: "08-08-2008");
26
27         System.out.println("Nama \t\t\t : " + mahasiswa2.nama);
28         System.out.println("NPM \t\t\t : " + mahasiswa2.npm);
29         System.out.println("Tanggal Lahir \t : " + mahasiswa2.tanggalLahir);
30     }
31 }

```

7. Tanggal yang tampil dari program ini masih menampilkan format tanggal yang lengkap, untuk memformatnya kita bisa gunakan perintah

```

System.out.println("Nama \t\t\t : " + mahasiswa2.nama);
System.out.println("NPM \t\t\t : " + mahasiswa2.npm);
System.out.println("Tanggal Lahir \t : " + simpleDateFormat.format(mahasiswa2.tanggalLahir));

```

B. Membuat Method

Selain atribut, class juga mampu menampung method/behavior yang bisa dijalankan oleh objek

1. Kembali ke class “Mahasiswa” dan buat method “tampilkanAtribut”

```

1 package model;
2
3 import java.text.SimpleDateFormat;
4 import java.util.Date;
5
6 public class Mahasiswa {
7     String npm;
8     String nama;
9     Date tanggalLahir;
10
11     void tampilkanAtribut(){
12         String polaTanggal = "dd-MM-yyyy";
13         SimpleDateFormat simpleDateFormat =new SimpleDateFormat(polaTanggal);
14
15         System.out.println("Nama \t\t\t : " + this.nama);
16         System.out.println("NPM \t\t\t : " + this.npm);
17         System.out.println("Tanggal Lahir \t : " + simpleDateFormat.format(this.tanggalLahir));
18     }
19 }

```

Method ini memiliki awalan “void” atau hampa yang artinya method tidak memiliki nilai return / nilai balik sehingga tidak bisa ditampung ke dalam variabel lain, bisa diperhatikan juga bahwa didalam method tersebut tidak terdapat kata kunci “return”.

2. Contoh method lain yang lebih sederhana

```

void menyapa(){
    System.out.println("Hai nama aku : " + this.nama);
}

```

3. Panggil pada “MahasiswaBeraksi”

```

mahasiswa.tampilkanAtribut();
mahasiswa.menyapa();|

```

```
Mahasiswa mahasiswa2 = new Mahasiswa();
```

4. Selain method bernilai void, method juga bisa memiliki nilai balik / return value, dan nilai tersebut ditentukan terlebih dahulu tipe datanya. Misalkan juga kita ingin membuat method “hitungUsia” maka bisa ditentukan “usia” akan memiliki tipe data “int”.

```

int hitungUsia(){
    Calendar tanggalLahir = Calendar.getInstance();
    tanggalLahir.setTime(this.tanggalLahir);
    Calendar hariIni = Calendar.getInstance();

    int selisihTahun = hariIni.get(Calendar.YEAR) - tanggalLahir.get(Calendar.YEAR);

    if(hariIni.get(Calendar.MONTH) < tanggalLahir.get(Calendar.MONTH) ){
        selisihTahun--;
    }else{
        if(hariIni.get(Calendar.MONTH) == tanggalLahir.get(Calendar.MONTH)
        && hariIni.get(Calendar.DAY_OF_MONTH) < tanggalLahir.get(Calendar.DAY_OF_MONTH)){
            selisihTahun--;
        }
    }

    return selisihTahun;
}

```

5. Panggil method tersebut di “MahasiswaBeraksi”

```

mahasiswa.tampilkanAtribut();
mahasiswa.menyapa();
System.out.println("Usiaku : " + mahasiswa.hitungUsia() + " tahun");

```

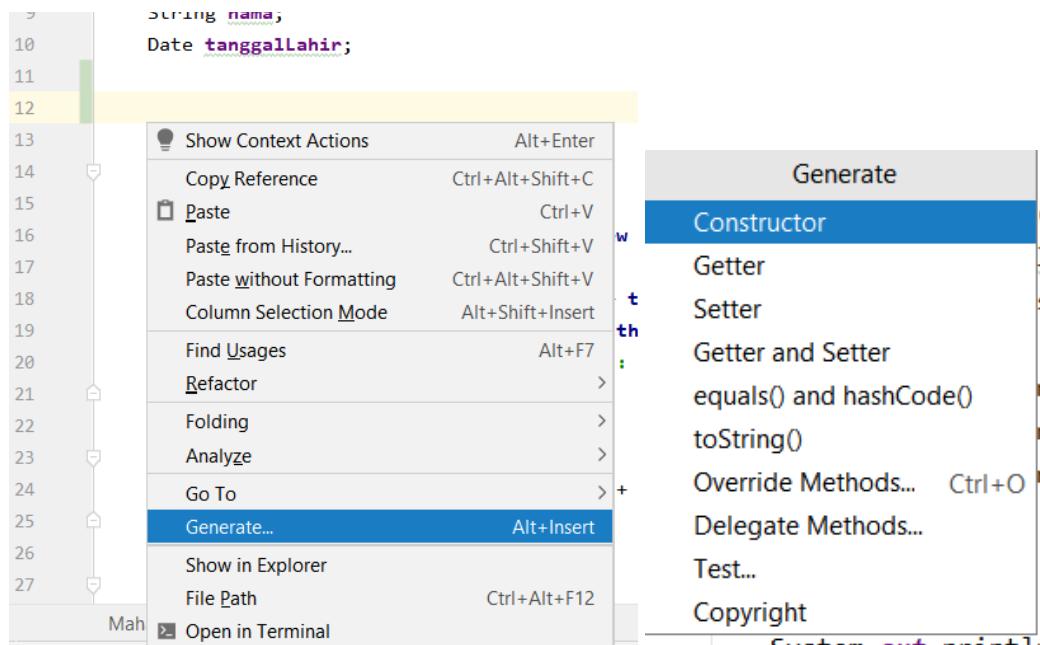
C. Membuat Constructor

Constructor merupakan method yang jalan pertama kali ketika sebuah objek di-instance. Sebelumnya kita sudah menggunakan Constructor pada saat meng-instance class Mahasiswa. Constructor yang dimaksud adalah “Mahasiswa()” pada “Mahasiswa mahasiswa = new Mahasiswa()”.

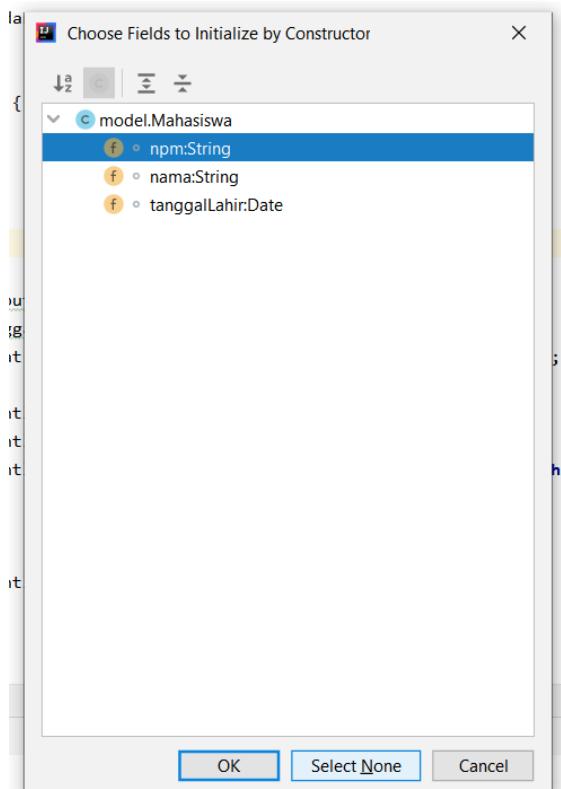
Bisa dilihat bahwa constructor memiliki nama yang sama persis dengan nama class-nya, dan secara default sudah ada constructor tanpa argumen yang sesuai dengan nama Class meskipun kita tidak pernah membuatnya.

Jika ingin mengubah perintah yang jalan ketika pertama kali objek di-instance maka kita bisa membuatnya sendiri.

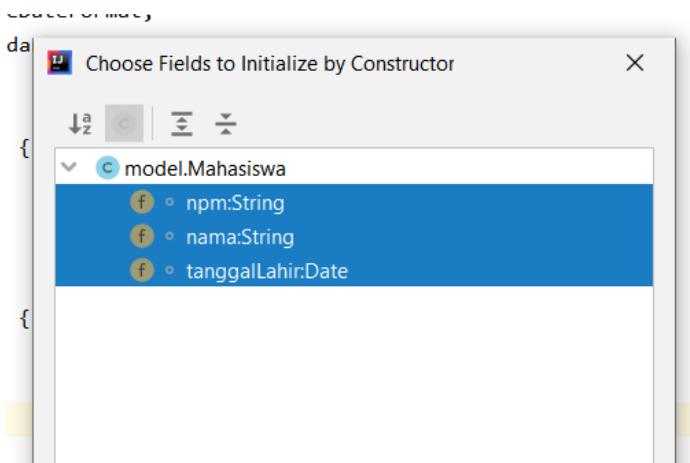
1. Klik kanan pada space kosong pilih “Generate” kemudian pilih “Constructor”



2. Berikutnya pada dialog yang muncul, bisa diklik “Select None” terlebih dahulu.



3. Ulangi “Generate” dan “Constructor”, berikutnya pilih semua atribut, dengan menekan kombinasi tombol shift dan klik



4. Berikut hasil generate-nya

```
public Mahasiswa() {  
}  
  
public Mahasiswa(String npm, String nama, Date tanggalLahir) {  
    this.npm = npm;  
    this.nama = nama;  
    this.tanggalLahir = tanggalLahir;  
}
```

5. Selain dengan “Generate”, constructor bisa saja ditulis sendiri, buat constructor berikut dengan diketik

```
public Mahasiswa(String npm, String nama, String tanggalLahir) throws ParseException {  
    String polaTanggal = "dd-MM-yyyy";  
    SimpleDateFormat simpleDateFormat =new SimpleDateFormat(polaTanggal);  
  
    this.npm = npm;  
    this.nama = nama;  
    this.tanggalLahir = simpleDateFormat.parse(tanggalLahir);  
}
```

6. Pada class “MahasiswaBeraksi” bisa diterapkan Constructor yang sudah dibuat pada object “mahasiswa2”

```

public class MahasiswaBeraksi {
    public static void main(String[] args) throws ParseException {

        String polaTanggal = "dd-MM-yyyy";
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat(polaTanggal);

        Mahasiswa mahasiswa = new Mahasiswa();
        mahasiswa.npm = "19639999";
        mahasiswa.nama = "Sembilu";
        mahasiswa.tanggalLahir = simpleDateFormat.parse("09-09-2009");

        mahasiswa.tampilkanAtribut();
        mahasiswa.menyapa();
        System.out.println("Usiaku : " + mahasiswa.hitungUsia() + " tahun");

        Mahasiswa mahasiswa2 = new Mahasiswa(npm: "19638888", nama: "Delapu", tanggalLahir: "08-08-2008");
        mahasiswa2.tampilkanAtribut();
    }
}

```

7. Pada class “Mahasiswa” bisa kita buat agar menjadi lebih sederhana.

```

public class Mahasiswa {
    String npm;
    String nama;
    Date tanggalLahir;
    final String POLA_TANGGAL = "dd-MM-yyyy";
    final SimpleDateFormat SDF = new SimpleDateFormat(POLA_TANGGAL);

    void tampilkanAtribut(){
        System.out.println("Nama \t\t\t : " + this.nama);
        System.out.println("NPM \t\t\t : " + this.npm);
        System.out.println("Tanggal Lahir \t : " + SDF.format(this.tanggalLahir));
    }

    public Mahasiswa(String npm, String nama, String tanggalLahir) {
        this.npm = npm;
        this.nama = nama;
        try {
            this.tanggalLahir = SDF.parse(tanggalLahir);
        } catch (ParseException e) {
            System.err.println("Kesalahan pada tanggal lahir");
        }
    }
}

```

D. Enkapsulasi

Enkapsulasi merupakan proses membungkus data ke dalam unit tertentu sehingga tidak dapat diakses oleh class atau function. Implementasinya pada Java menggunakan modifier berikut

Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Gambar dari <https://www.petanikode.com/java-oop-modifier/>

1. Buat sebuah class Main di luar package model.



2. Instance Mahasiswa dan jalankan method "tampilkanAtribut()"

```

public class Main {
    public static void main(String[] args) {
        Mahasiswa m1 = new Mahasiswa( npm: "19637777", nama: "Pitung", tanggalLahir: "07-07-07");
        m1.tampilkanAtribut();
        System.out.println(m1.nama);
    }
}

```

Nampak tanda merah dengan keterangan bahwa method tersebut tidak bersifat public sehingga tidak bisa diakses oleh Class di luar package yang sama. Agar bisa dijalankan kita tambahkan modifier public pada method tersebut

```

public void tampilkanAtribut(){
    System.out.println("Nama \t\t\t : " + this.nama);
    System.out.println("NPM \t\t\t : " + this.npm);
    System.out.println("Tanggal Lahir \t : " + SDF.format(this.tanggalLahir));
}

```

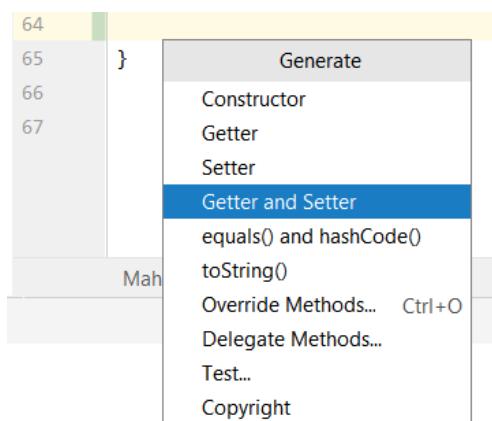
- Kembali ke class “Main”, tanda merah tadi sudah hilang, yang tersisa adalah tanda merah pada saat mengakses atribut “nama”

```

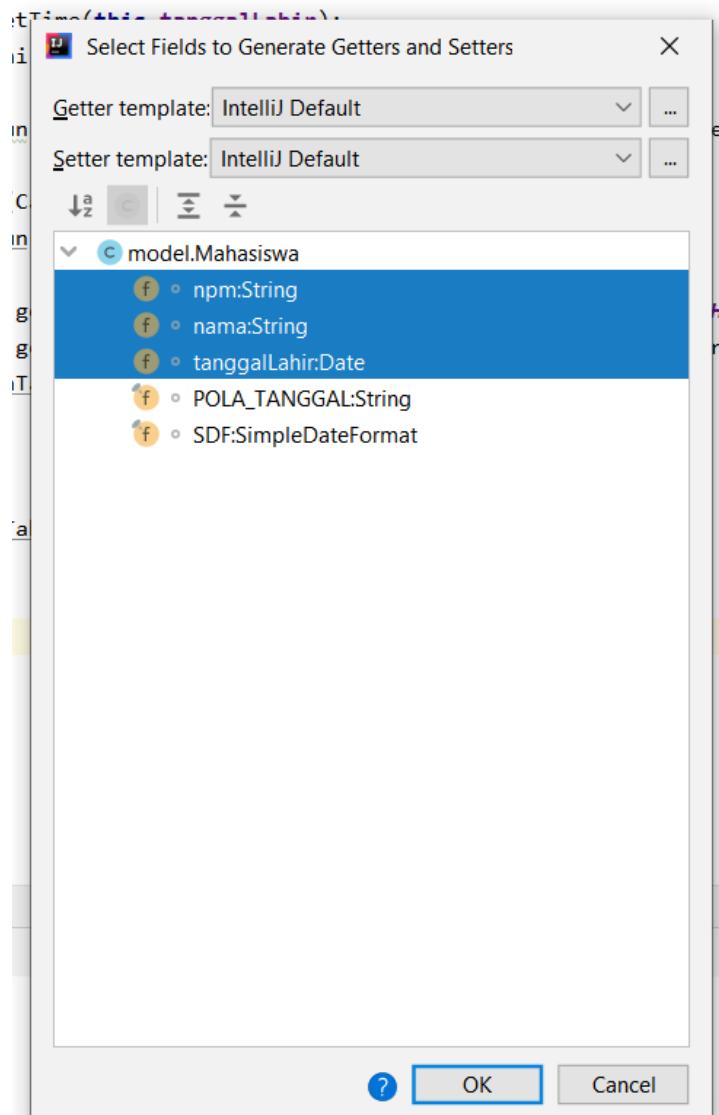
public class Main {
    public static void main(String[] args) {
        Mahasiswa m1 = new Mahasiswa( npm: "19637777", nama: "Pitung", tanggalLahir: "07-07-07");
        m1.tampilkanAtribut();
        System.out.println(m1.nama);
    }
}

```

- Khusus untuk atribut kita tidak serta merta memberikan modifier public kepadanya, karena atribut ini rentan maka lebih aman jika kita gunakan method untuk menentukan (set) dan mengambil (get) nilainya. Pada method tersebut bisa dilakukan modifikasi ataupun validasi sehingga atribut lebih aman. Untuk membuat method tersebut, ikut cara meng-generate constructor, klik kanan di tempat kosong -> Generate -> Getter and Setter



- Pilih atribut yang ingin dibuat Getter and Setter-nya, biasanya seluruh atribut perlu dibuatkan Getter and Setter



6. Sehingga pada class “Main” kita bisa ambil nilai “nama” dengan method “getNama()”.

```
public class Main {  
    public static void main(String[] args) {  
        Mahasiswa m1 = new Mahasiswa( npm: "19637777", nama: "Pitung", tanggalLahir: "07-07-07");  
        m1.tampilkanAtribut();  
        System.out.println(m1.getNama());  
    }  
}
```

PRAKTIKUM JAVA I

MODUL 3

JAVA GUI DENGAN INTELLIJ IDEA

TUJUAN

- Memahami cara membangun *Graphical User Interface* (GUI) dalam Java berbasis teks
- Memahami sintaks dasar Java
- Memahami penggunaan Git untuk mengumpulkan tugas

ALAT DAN BAHAN

- IntelliJ IDEA (<https://www.jetbrains.com/idea/download/>)

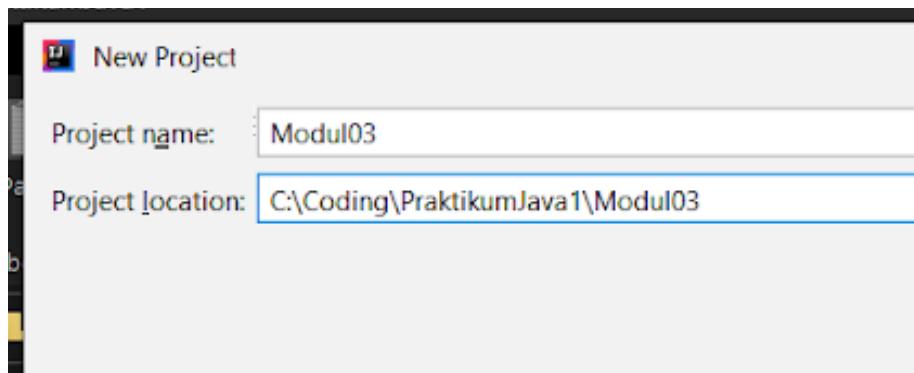
DASAR TEORI

Java memiliki fitur untuk GUI yang mana bisa dikembangkan dengan murni teks saja, dan juga dikembangkan dengan *drag-drop* tergantung pada IDE-nya. IntelliJ IDEA memiliki pendekatan yang berbeda dalam pengembangan GUI dibandingkan NetBeans - IDE Java lainnya, meskipun dalam hal ini dapat dilihat NetBeans lebih mudah digunakan. Meskipun demikian GUI dalam Java bisa dikembangkan hanya dengan teks, bahkan tanpa IDE, cukup bermodalkan notepad. IDE akan mempermudah proses manajemen package, debug, dan mengurangi kemungkinan error.

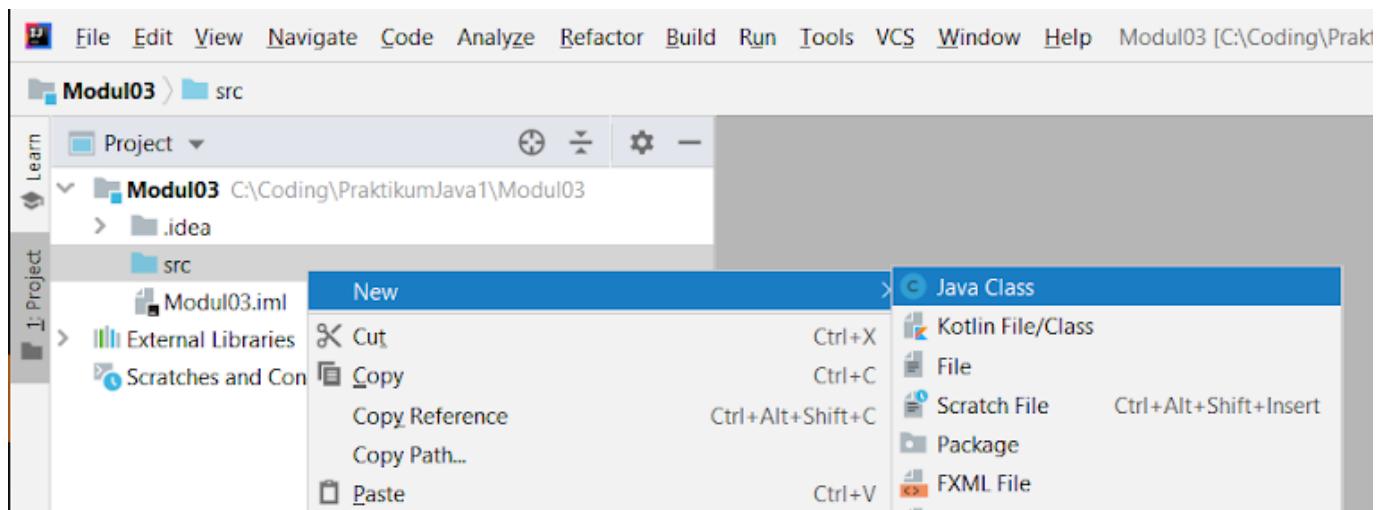
Kegiatan:

E. [Menampilkan Frame Pertama Kali](#)

8. Beri nama project Modulo3



9. Buat sebuah class baru “GuiByText”



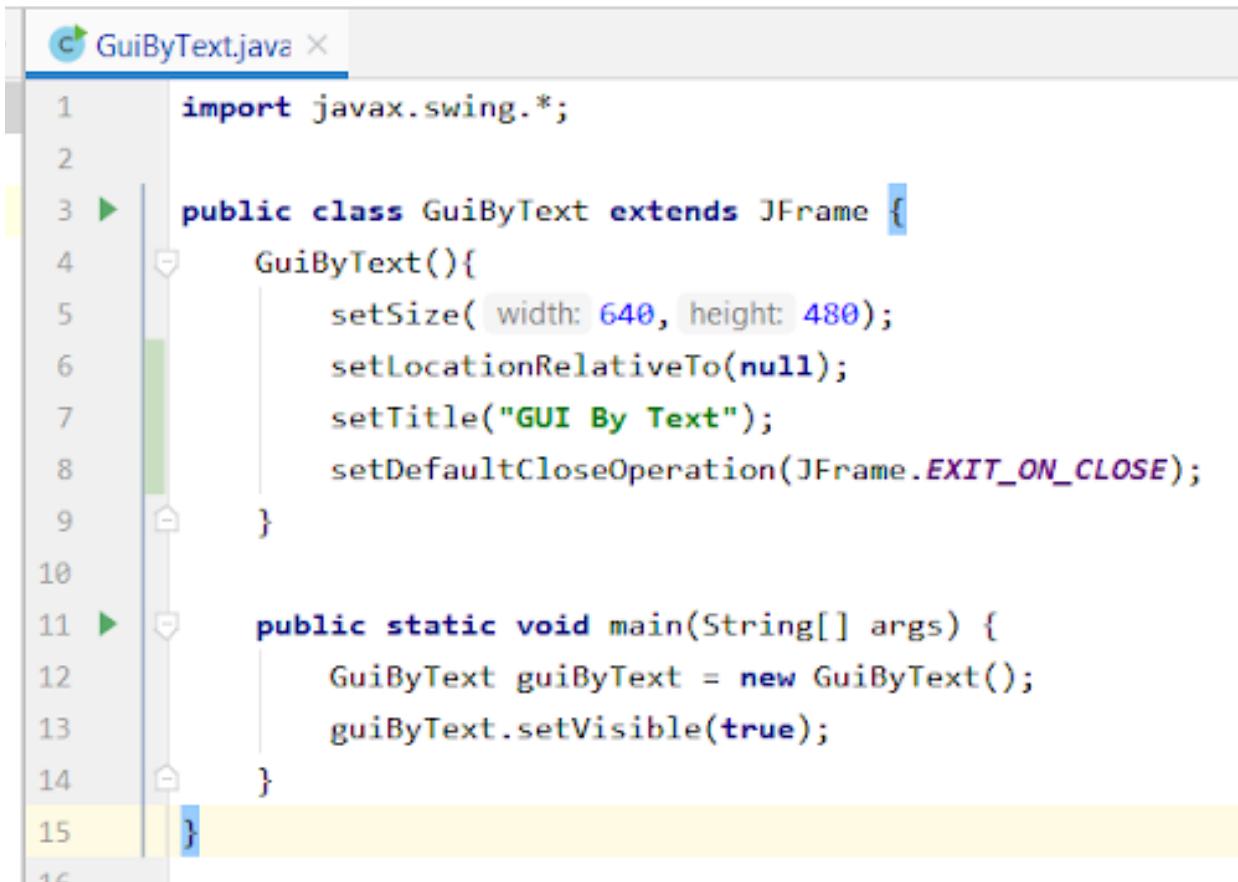
10. Tuliskan source code berikut

```
GuiByText.java
1 import javax.swing.*;
2
3 public class GuiByText extends JFrame {
4     GuiByText(){
5         setSize( width: 640, height: 480);
6     }
7
8     public static void main(String[] args) {
9         GuiByText guiByText = new GuiByText();
10        guiByText.setVisible(true);
11    }
12 }
```

11. Jalankan dan analisa JFrame yang dihasilkan, bandingkan dengan source code yang ditulis.
12. Kenapa ada GuiByText() di dalam class GuiByText? Apa fungsinya?
13. apa maksud dari perintah setSize() beserta argument di dalamnya?
14. apa maksud dari setVisible()? apa yang terjadi kalau argument nya diganti jadi false?

15. Commit.

16. Tambahkan setLocationRelativeTo(null), setTitle, dan setDefaultCloseOperation



```

1 import javax.swing.*;
2
3 public class GuiByText extends JFrame {
4     GuiByText(){
5         setSize( width: 640, height: 480);
6         setLocationRelativeTo(null);
7         setTitle("GUI By Text");
8         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9     }
10
11 public static void main(String[] args) {
12     GuiByText guiByText = new GuiByText();
13     guiByText.setVisible(true);
14 }
15 }
```

17. Jalankan dan analisis apa dampak dari ketiga perintah tersebut pada JFrame

F. Menambahkan dan Meletakkan Komponen

18. Tambahkan komponen JLabel dan JTextField

19. JLabel berfungsi untuk menampilkan tulisan, sedangkan JTextField untuk menerima inputan

20. Buat method meletakkanKomponen()

21. Method setLayout(null) menentukan bahwa layout pada frame kali ini adalah null atau tidak ada, sehingga setiap komponen akan diletakkan secara manual berdasarkan kordinatnya

22. Method add() berarti menambahkan komponen yang sebelumnya sudah di-instance ke dalam JFrame

23. Karena berlayout null, maka komponen tadi diletakkan secara manual dengan method setBounds()

24. Gambar Source Code ada dihalaman berikutnya

25. Commit



```

1 import javax.swing.*;
2
3 public class GuiByText extends JFrame {
4
5     JLabel label1 = new JLabel( text: "Coba Input 1");
6     JTextField textField1 = new JTextField();
7
8     GuiByText(){
9         setSize( width: 640, height: 480);
10        setLocationRelativeTo(null);
11        setTitle("GUI By Text");
12        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13    }
14
15    void meletakkanKomponen(){
16        getContentPane().setLayout(null);
17
18        getContentPane().add(label1);
19        getContentPane().add(textField1);
20
21        label1.setBounds( x: 16, y: 32, width: 100, height: 25);
22        textField1.setBounds( x: 124, y: 32, width: 100, height: 25);
23    }
24
25    public static void main(String[] args) {
26        GuiByText guiByText = new GuiByText();
27        guiByText.meletakkanKomponen();
28        guiByText.setVisible(true);
29    }
30}

```

G. Menambahkan Event Listener

Event listener adalah perintah yang akan berjalan ketika suatu event terjadi pada program. Contoh paling mudah adalah event Click pada komponen, biasanya pada komponen Button. Berikut langkahnya:

1. Instance sebuah Button dengan nama button1
2. lakukan add() dan setBounds() pada method meletakkanKomponen()
3. sehingga source code lengkapnya menjadi seperti berikut:

```

1 import javax.swing.*;
2
3 public class GuiByText extends JFrame {
4
5     JLabel label1 = new JLabel( text: "Coba Input 1");
6     JTextField textField1 = new JTextField();
7     JButton button1 = new JButton( text: "Greet");
8
9     GuiByText(){
10         setSize( width: 640, height: 480);
11         setLocationRelativeTo(null);
12         setTitle("GUI By Text");
13         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
14     }
15
16     void meletakkanKomponen(){
17         getContentPane().setLayout(null);
18
19         getContentPane().add(label1);
20         getContentPane().add(textField1);
21         getContentPane().add(button1);
22
23         label1.setBounds( x: 16, y: 32, width: 100, height: 25);
24         textField1.setBounds( x: 124, y: 32, width: 100, height: 25);
25         button1.setBounds( x: 232, y: 32, width: 80, height: 25);
26     }
27
28     public static void main(String[] args) {
29         GuiByText guiByText = new GuiByText();
30         guiByText.meletakkanKomponen();
31         guiByText.setVisible(true);
32     }
33 }

```

4. buat sebuah method menambahkanListener()
5. tambahkan eventListener pada button1 dengan perintah
button1.addActionListener.. dan seterusnya

```

30 void menambahkanListener(){
31     button1.addActionListener(new ActionListener() {
32         @Override
33         public void actionPerformed(ActionEvent actionEvent) {
34             String nama = textField1.getText();
35             JOptionPane.showMessageDialog( parentComponent: null, message: "Hai " + nama);
36         }
37     });
38 }

```

6. baris 34 adalah pembuatan sebuah variabel nama dengan tipe data String yang nilainya mengambil dari teks yang tertulis pada textField1
7. baris 35 adalah showMessageDialog yang menggunakan variabel nama
8. Commit

H. Contoh Perhitungan Luas

1. Tambahkan 3 JLabel, 3 TextField, dan 1 Button

```

9 JButton button1 = new JButton( text: greet );
10
11 JLabel label2 = new JLabel( text: "Panjang");
12 JLabel label3 = new JLabel( text: "Lebar");
13 JLabel label4 = new JLabel( text: "Hasil");
14 JTextField tPanjang = new JTextField();
15 JTextField tLebar = new JTextField();
16 JTextField tHasil= new JTextField();
17 JButton bHitung = new JButton( text: "Hitung Luas");
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

```

2. Jalankan add() pada method meletakkanKomponen()

```

1 JButton button1 = new JButton( text: greet );
2 JPanel panel = new JPanel();
3 panel.setLayout( layout );
4
5 panel.add(button1);
6 getContentPane().add(panel);
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

```

3. Tambahkan setBounds() untuk komponen baru

```
44     label2.setBounds( x: 16, y: 80, width: 100, height: 25);
45     tPanjang.setBounds( x: 124, y: 80, width: 100, height: 25);
46     label3.setBounds( x: 16, y: 113, width: 100, height: 25);
47     tLebar.setBounds( x: 124, y: 113, width: 100, height: 25);
48     bHitung.setBounds( x: 16, y: 146, width: 208, height: 25);
49     label4.setBounds( x: 16, y: 179, width: 100, height: 25);
50     tHasil.setBounds( x: 124, y: 179, width: 100, height: 25);
```

4. Tambahkan actionPerformed untuk button bHitung

```
61
62     bHitung.addActionListener(actionEvent -> {
63         int panjang = Integer.parseInt(tPanjang.getText());
64         int lebar = Integer.parseInt(tLebar.getText());
65
66         int hasil = panjang * lebar;
67         String sHasil = String.valueOf(hasil);
68
69         tHasil.setText(sHasil);
70     });
71 }
```

PRAKTIKUM JAVA I

MODUL 4

KONEKSI MYSQL

TUJUAN

- Memahami cara membangun *Graphical User Interface (GUI)* dengan drag drop IntelliJ
- Memahami cara menghubungkan ke database MySQL

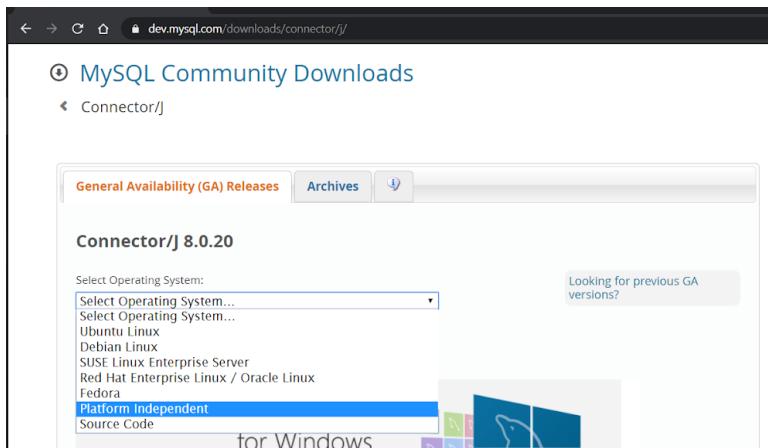
ALAT DAN BAHAN

- IntelliJ IDEA (<https://www.jetbrains.com/idea/download/>)
- MySQL (xampp)
- mysql-connector-java

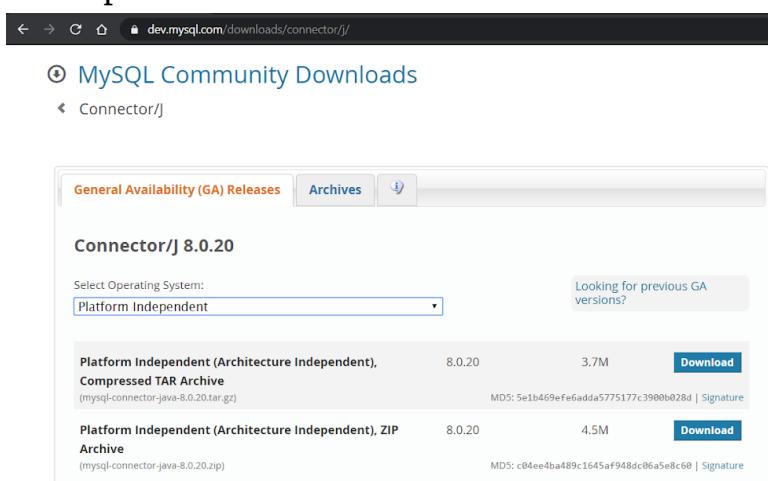
Kegiatan:

A. Download mysql-connector-java

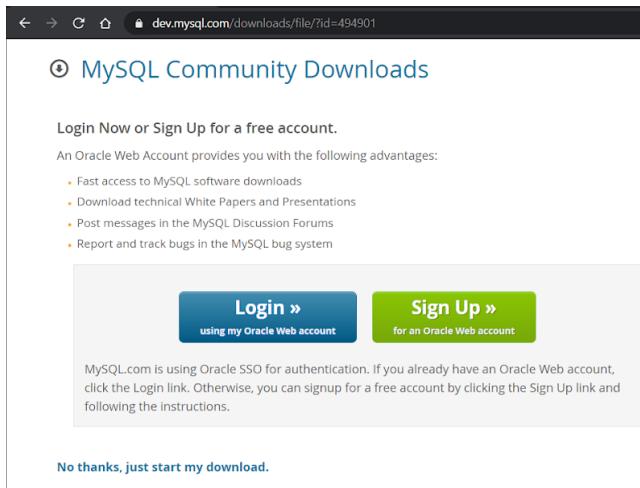
1. Buka halaman <https://dev.mysql.com/downloads/connector/j/>
2. pilih Platform Independent



3. Pilih zip



4. Pilih No thanks...



5. Extract ke folder yang mudah dicari.

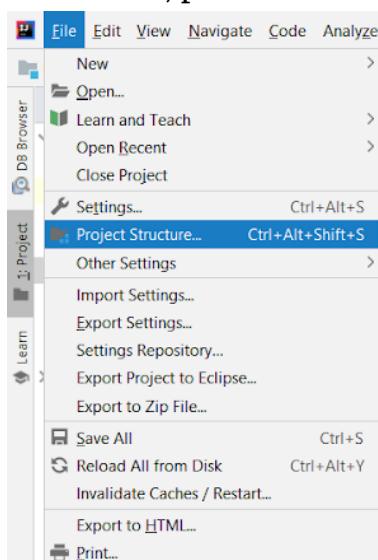
B. Membuat Database

1. Buat sebuah database pada PhpMyAdmin dengan nama db_pekerjaan_rumah
2. Buat table Matakuliah

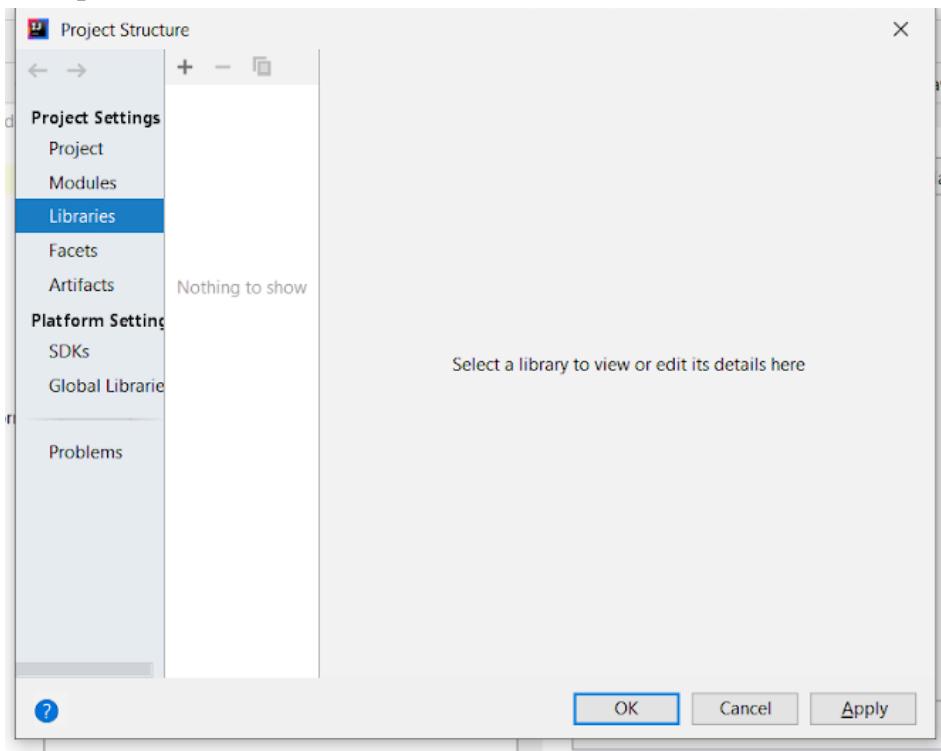
```
CREATE TABLE `db_pekerjaan_rumah`.`matakuliah` (
  `id_matakuliah` INT NOT NULL AUTO_INCREMENT ,
  `nama_matakuliah` VARCHAR(200) NOT NULL ,
  `singkatan_matakuliah` VARCHAR(9) NOT NULL ,
  `nama_dosen` VARCHAR(200) NOT NULL ,
  `kontak_dosen` VARCHAR(200) NOT NULL ,
  `aktif` BOOLEAN NOT NULL ,
  PRIMARY KEY (`id_matakuliah`)) ENGINE = InnoDB;
```

C. Membuat koneksi dalam project IntelliJ IDEA

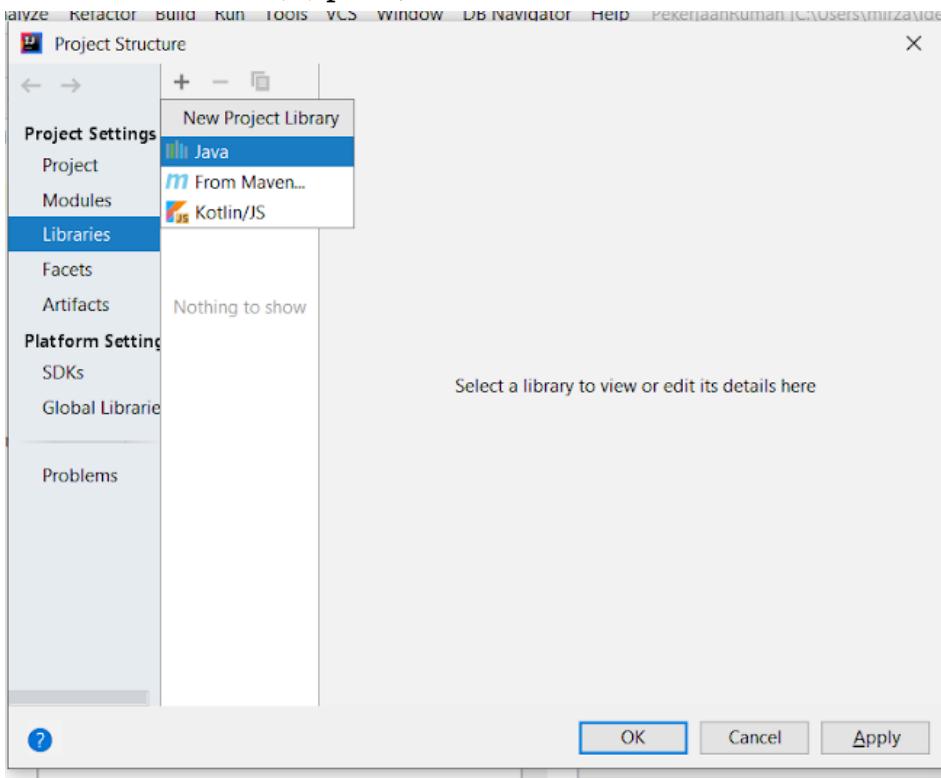
1. Buat project baru
2. Pada menu, pilih File->Project Structure



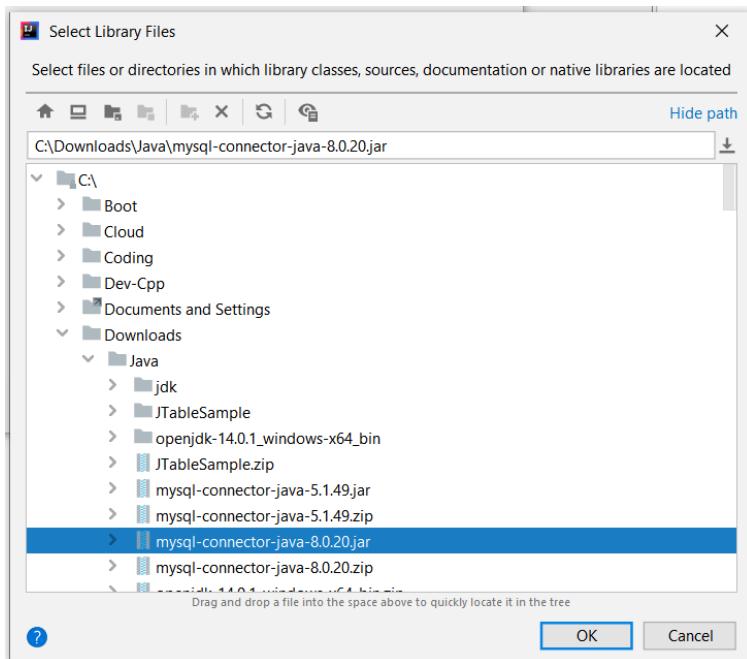
3. Pilih pada Libraries



4. Klik tanda tambah (+) pilih Java



5. Cari file yang sudah di download tadi -> OK -> OK -> OK



6. Buat package “helper”, didalamnya buat class “MyConnection”
 7. Pada class tersebut buat atribut

```
private static String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
private static String URL = "jdbc:mysql://localhost:3306/";
private static String DB_NAME = "db_pekerjaan_rumah";
private static String TIMEZONE = "?serverTimezone=Asia/Makassar";
private static String USER = "root";
private static String PASS = "";
```

8. Buat method

```
public Connection getConnection(){
    Connection con = null;

    try {
        Class.forName(JDBC_DRIVER);
        System.out.println("Sedang menghubungkan...");
        con = DriverManager.getConnection(url: URL+DB_NAME+TIMEZONE,USER,PASS);
        System.out.println("Berhasil");
        return con;
    } catch (ClassNotFoundException | SQLException e) {
        e.printStackTrace();
        return null;
    }
}
```

9. Buat main method untuk menguji koneksi

```

public static void main(String[] args) {
    MyConnection myConnection = new MyConnection();
    myConnection.getConnection();
}

```

D. Menampilkan Data

1. Isi data pada table matakuliah, contoh:

id_matakuliah	nama_matakuliah	singkatan_matakuliah	nama_dosen	kontak_dosen	aktif
1	Pemrograman Berorientasi Objek 1	PBO1	Mirza Yogy Kurniawan	0818000000	1
2	Pemrograman Web	Web 1	Sandika Galih	0823000000	0
3	Jaringan Komputer	Jarkom	Onno W. Purbo	0833088088	1

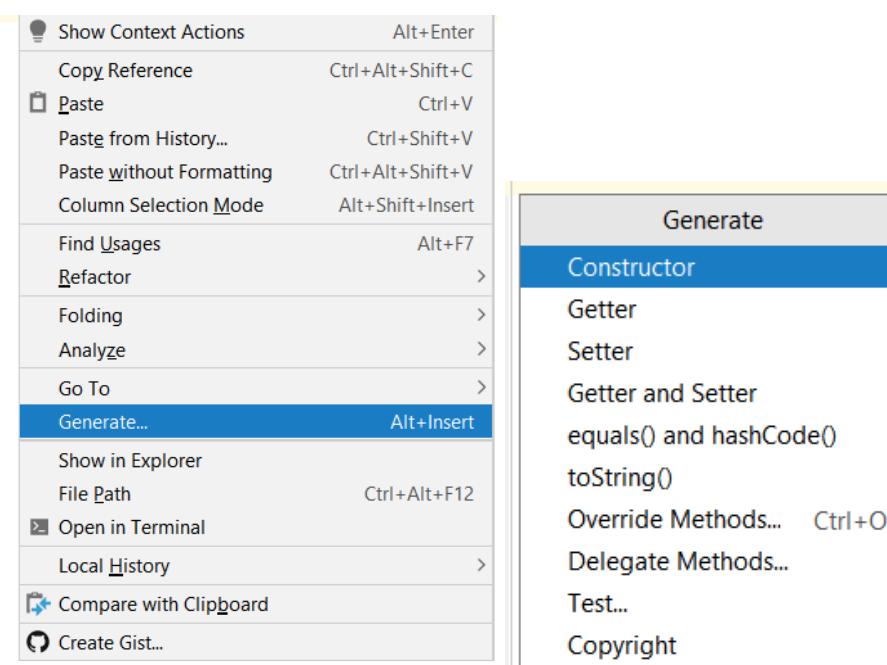
2. Buat package “model”, didalamnya buat class “Matakuliah”
3. Isikan dengan atribut berikut

```

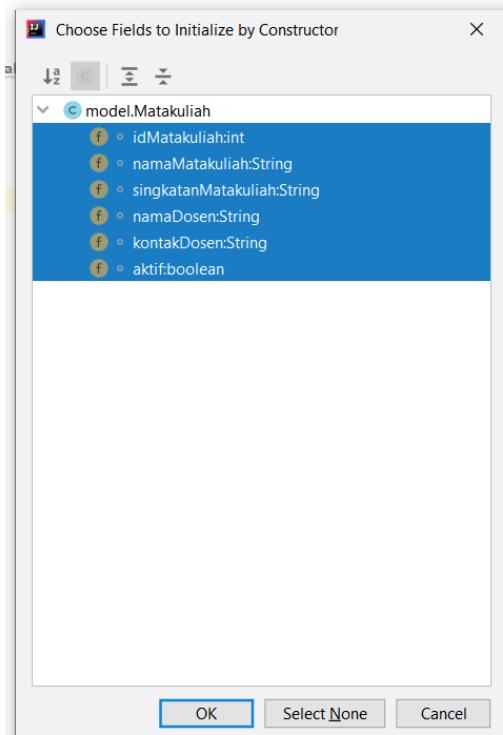
int idMatakuliah;
String namaMatakuliah;
String singkatanMatakuliah;
String namaDosen;
String kontakDosen;
boolean aktif;

```

4. Buat **constructor** Matakuliah, klik kanan di dalam class



5. Pilih semua atribut, gunakan tombol Shift, OK



6. Dengan cara yang sama, buat **Getter and Setter**
7. Kembali ke method “main” class “MyConnection”
8. Ganti isinya sebagai berikut

```

public static void main(String[] args) {
    Matakuliah matakuliah;
    ArrayList<Matakuliah> matakuliahList = new ArrayList<>();
    MyConnection myConnection = new MyConnection();
    Connection con = myConnection.getConnection();

    String selectQuery = "Select * FROM matakuliah ";
    Statement statement;
    ResultSet resultSet;

    try{
        statement = con.createStatement();
        resultSet = statement.executeQuery(selectQuery);
        while(resultSet.next()){
            matakuliah = new Matakuliah(
                resultSet.getInt( "id_matakuliah"),
                resultSet.getString( "nama_matakuliah"),
                resultSet.getString( "singkatan_matakuliah"),
                resultSet.getString( "nama_dosen"),
                resultSet.getString( "kontak_dosen"),
                resultSet.getBoolean( "aktif")
            );
            matakuliahList.add(matakuliah);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

```

```

        for (Matakuliah value : matakuliahList) {
            System.out.print(value.getIdMatakuliah());
            System.out.print("\t");
            System.out.print(value.getSingkatanMatakuliah());
            System.out.print("\t");
            System.out.print(value.getNamaDosen());
            System.out.println();
        }
    }
}

```

9. Jalankan dan perhatikan isinya

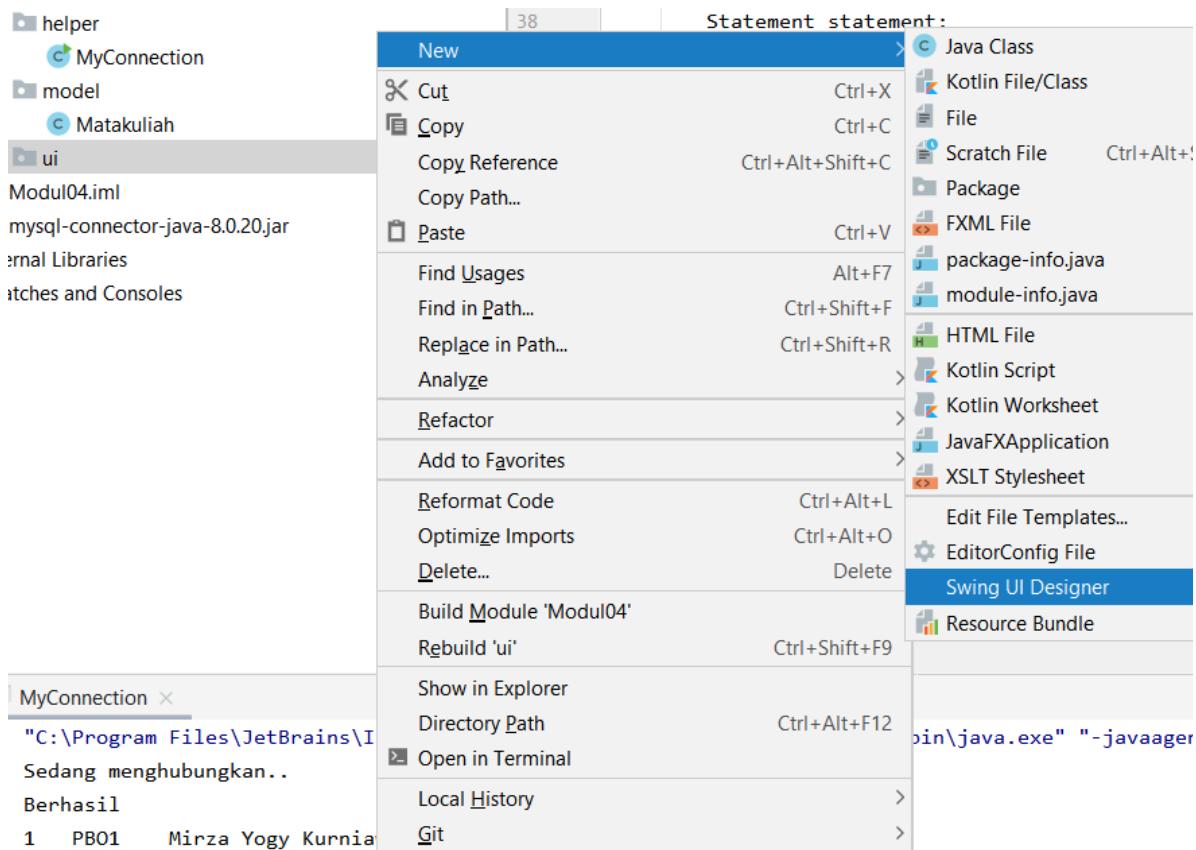
```

Run: MyConnection x
[C:\Program Files\JetBrains\IntelliJ IDEA
Sedang menghubungkan..
Berhasil
1 PBO1 Mirza Yogy Kurniawan
2 Web 1 Sandika Galih
3 Jarkom Onno W. Purbo
Process finished with exit code 0

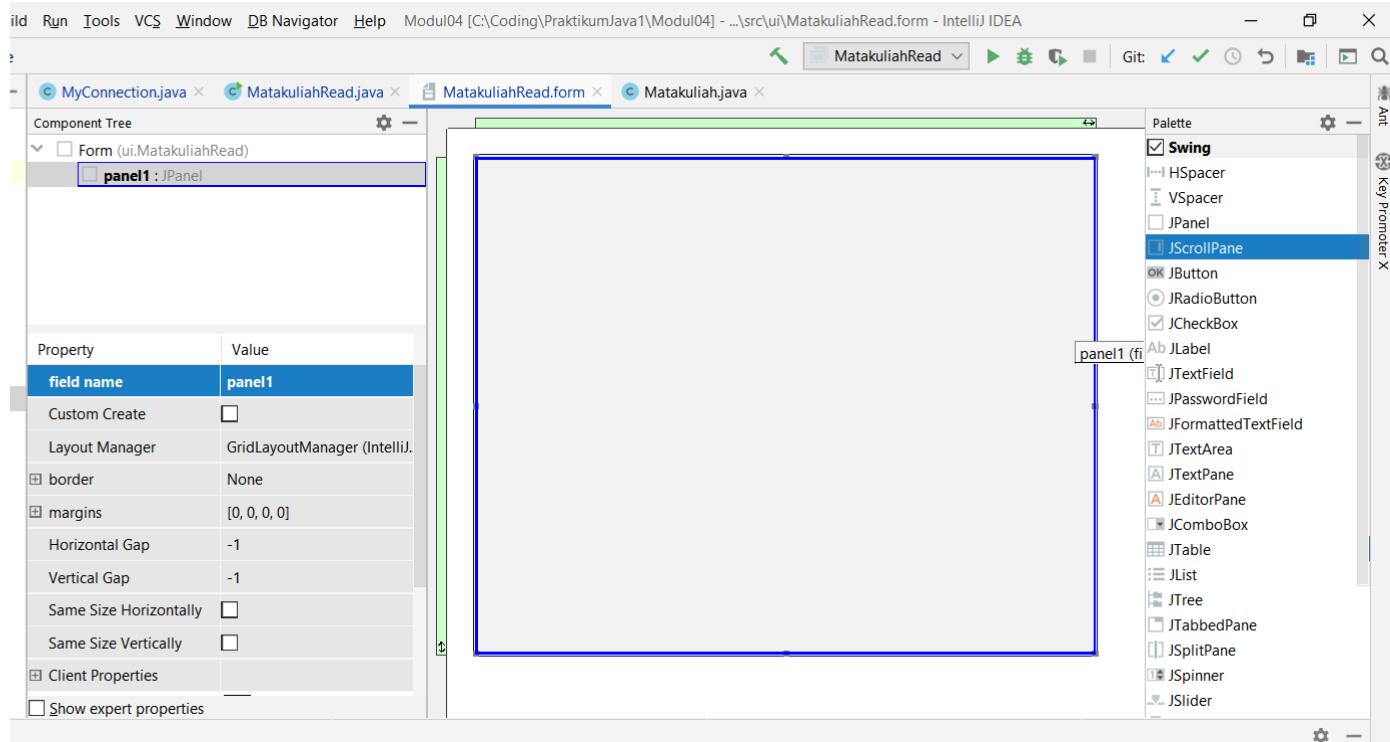
```

E. Menampilkan Data dengan JTable

10. Buat package “ui” dan buat GUI dengan cara

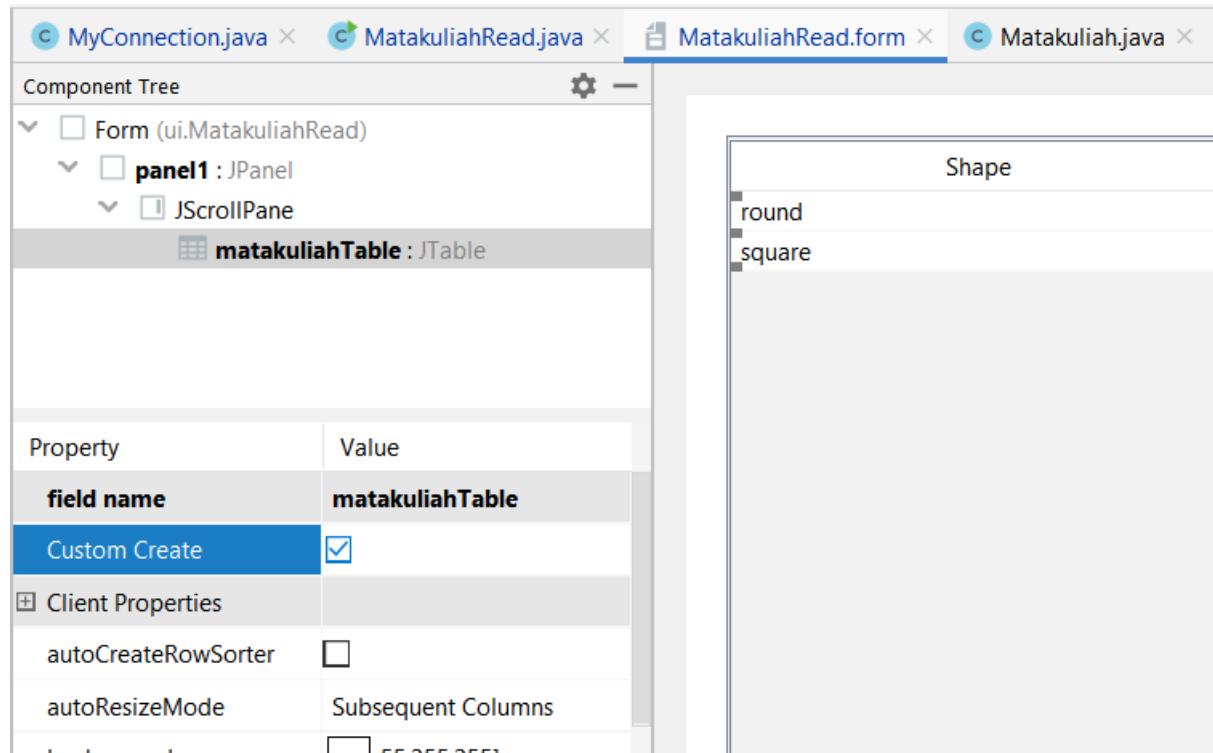


11. Letakkan JScrollPane pada tengah panel



12. Letakkan JTable juga pada tengah panel, ganti property field name menjadi matakuliahTable

13. Beri centang pada komponen table1->Property->Custom Create



14. Sehingga muncul method createUiComponents

15. Isi method tersebut

```

private void createUIComponents() {

    Matakuliah matakuliah;
    ArrayList<Matakuliah> matakuliahList = new ArrayList<>();
    MyConnection myConnection = new MyConnection();
    Connection con = myConnection.getConnection();

    String selectQuery = "Select * FROM matakuliah ";
    Statement statement;
    ResultSet resultSet;

    try{
        statement = con.createStatement();
        resultSet = statement.executeQuery(selectQuery);
        while(resultSet.next()){
            matakuliah = new Matakuliah(
                resultSet.getInt( s: "id_matakuliah"),
                resultSet.getString( s: "nama_matakuliah"),
                resultSet.getString( s: "singkatan_matakuliah"),
                resultSet.getString( s: "nama_dosen"),
                resultSet.getString( s: "kontak_dosen"),
                resultSet.getBoolean( s: "aktif")
            );
            matakuliahList.add(matakuliah);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    Object[][] row = new Object[matakuliahList.size()][6];

    for (int i = 0; i < matakuliahList.size(); i++) {
        row[i][0] = matakuliahList.get(i).getIdMatakuliah();
        row[i][1] = matakuliahList.get(i).getNamaMatakuliah();
        row[i][2] = matakuliahList.get(i).getSingkatanMatakuliah();
        row[i][3] = matakuliahList.get(i).getNamaDosen();
        row[i][4] = matakuliahList.get(i).getKontakDosen();
        row[i][5] = matakuliahList.get(i).isAktif();
    }
}

```

```

        String matakuliahHeader[] = {"ID", "Matakuliah", "Singkatan", "Dosen", "Kontak", "Aktif"};
        TableModel tableModel = new DefaultTableModel(row, matakuliahHeader);
        matakuliahTable = new JTable(tableModel);

    }

```

16. Tambahkan extends JFrame pada class MatakuliahRead

```
public class MatakuliahRead extends JFrame {
```

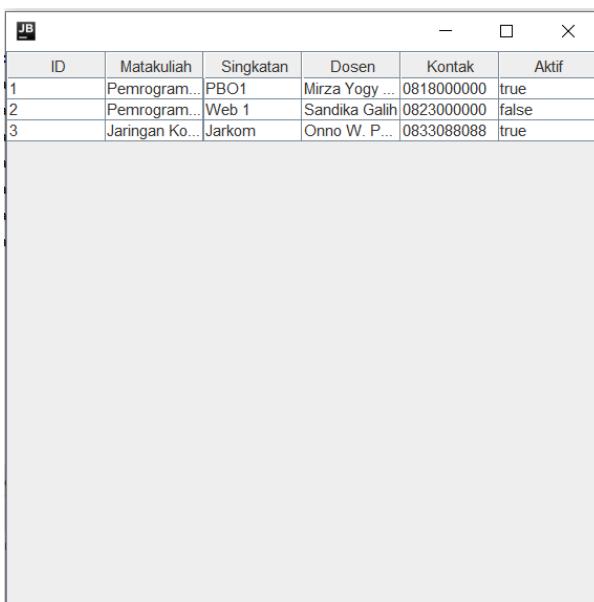
17. Buat class main dan isi:

```

public static void main(String[] args) {
    MatakuliahRead matakuliahRead = new MatakuliahRead();
    matakuliahRead.setContentPane(new MatakuliahRead().panel1);
    matakuliahRead.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    matakuliahRead.createUIComponents();
    matakuliahRead.pack();
    matakuliahRead.setLocationRelativeTo(null);
    matakuliahRead.setVisible(true);
}

```

18. Jalankan



ID	Matakuliah	Singkatan	Dosen	Kontak	Aktif
1	Pemrogram...	PBO1	Mirza Yogy...	0818000000	true
2	Pemrogram...	Web 1	Sandika Galih	0823000000	false
3	Jaringan Ko...	Jarkom	Onno W. P...	0833088088	true

19. Coba tambahkan Label, Button dan TextField

Cari Mata Kuliah/Dosen	<input type="text"/>	<input type="button" value="Cari"/>
Shape	Color	
round	red	
square	green	

	<input type="button" value="Tambah"/>	<input type="button" value="Ubah"/>	<input type="button" value="Hapus"/>	<input type="button" value="Batal"/>
--	---------------------------------------	-------------------------------------	--------------------------------------	--------------------------------------

20. Selanjutnya dilanjutkan di Praktikum Java 2 :)