

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2
Kelas : 4IA06
Praktikum ke- : 2
Tanggal : 22 Oktober 2024
Materi : Anatomi Kelas, Struktur Aplikasi JAVA, Object Orented Programming (OOP) dengan JAVA
NPM : 51421517
Nama : William Devin Septianus Pranggono
Ketua Asisten :
Paraf Asisten :
Nama Asisten : Gilbert Jefferson Faozato Mendrofa
Jumlah Lembar : 7 lembar

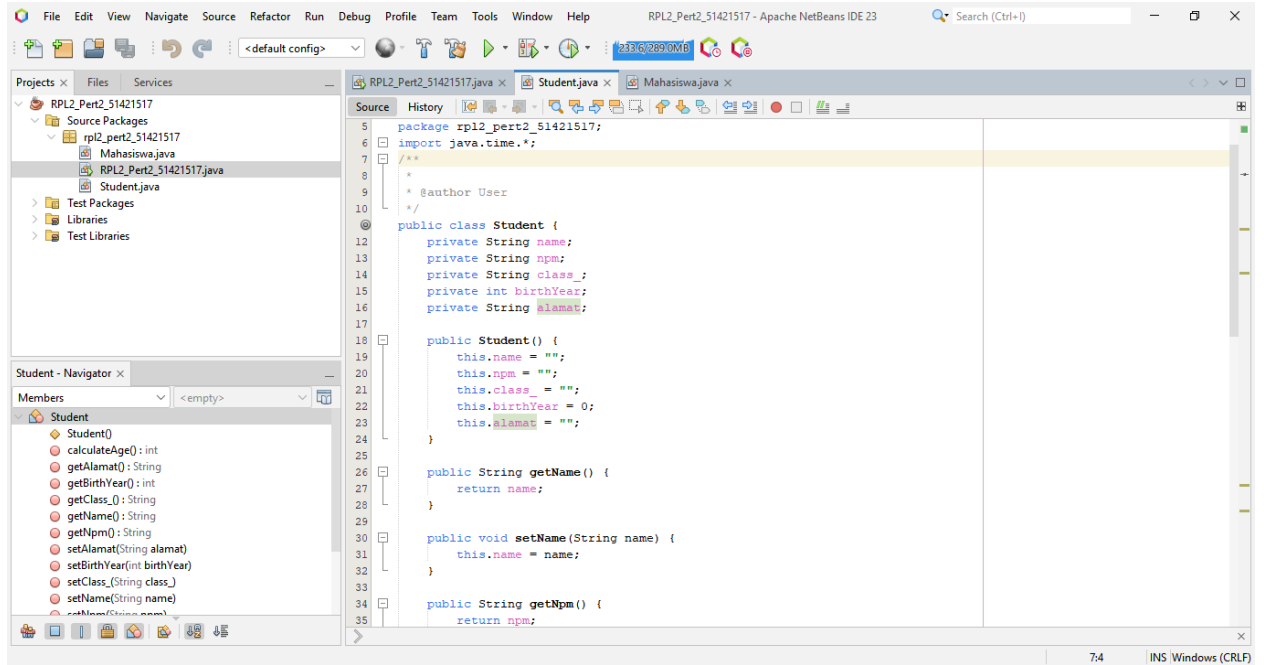
LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

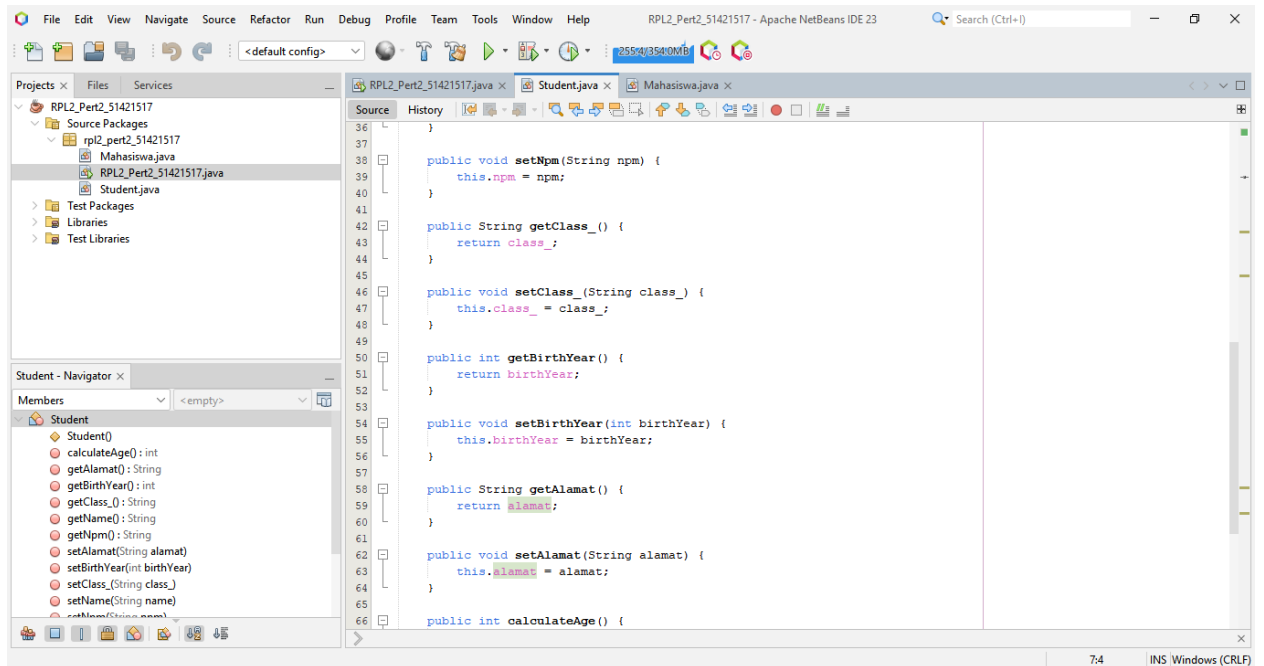
2024

LISTING PROGRAM

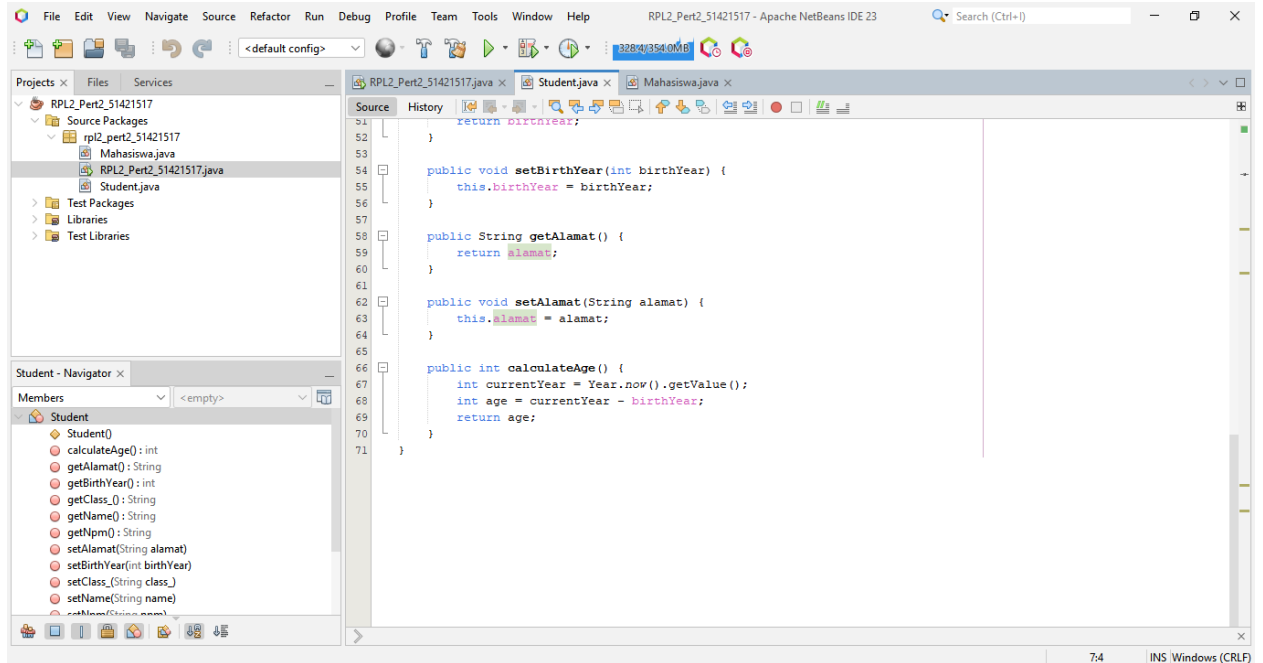
1. Student.java :



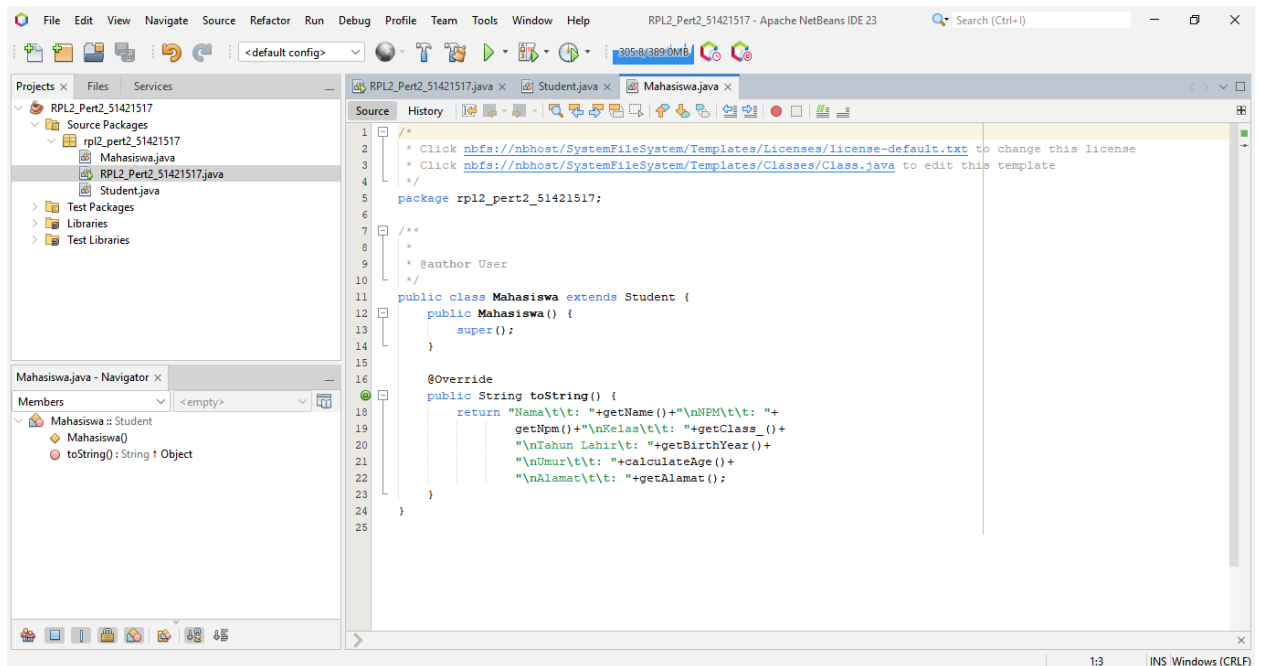
```
5 package rp12_pert2_51421517;
6 import java.time.*;
7
8 /**
9  * @author User
10  */
11
12 public class Student {
13     private String name;
14     private String npm;
15     private String class_;
16     private int birthYear;
17     private String alamat;
18
19     public Student() {
20         this.name = "";
21         this.npm = "";
22         this.class_ = "";
23         this.birthYear = 0;
24         this.alamat = "";
25     }
26
27     public String getName() {
28         return name;
29     }
30
31     public void setName(String name) {
32         this.name = name;
33     }
34
35     public String getNpm() {
36         return npm;
37     }
38
39     public int calculateAge() {
40         return birthYear;
41     }
42
43     public String getClass_() {
44         return class_;
45     }
46
47     public void setClass_(String class_) {
48         this.class_ = class_;
49     }
50
51     public int getBirthYear() {
52         return birthYear;
53     }
54
55     public void setBirthYear(int birthYear) {
56         this.birthYear = birthYear;
57     }
58
59     public String getAlamat() {
60         return alamat;
61     }
62
63     public void setAlamat(String alamat) {
64         this.alamat = alamat;
65     }
66
67     public int calculateAge() {
68         return birthYear;
69     }
70 }
```



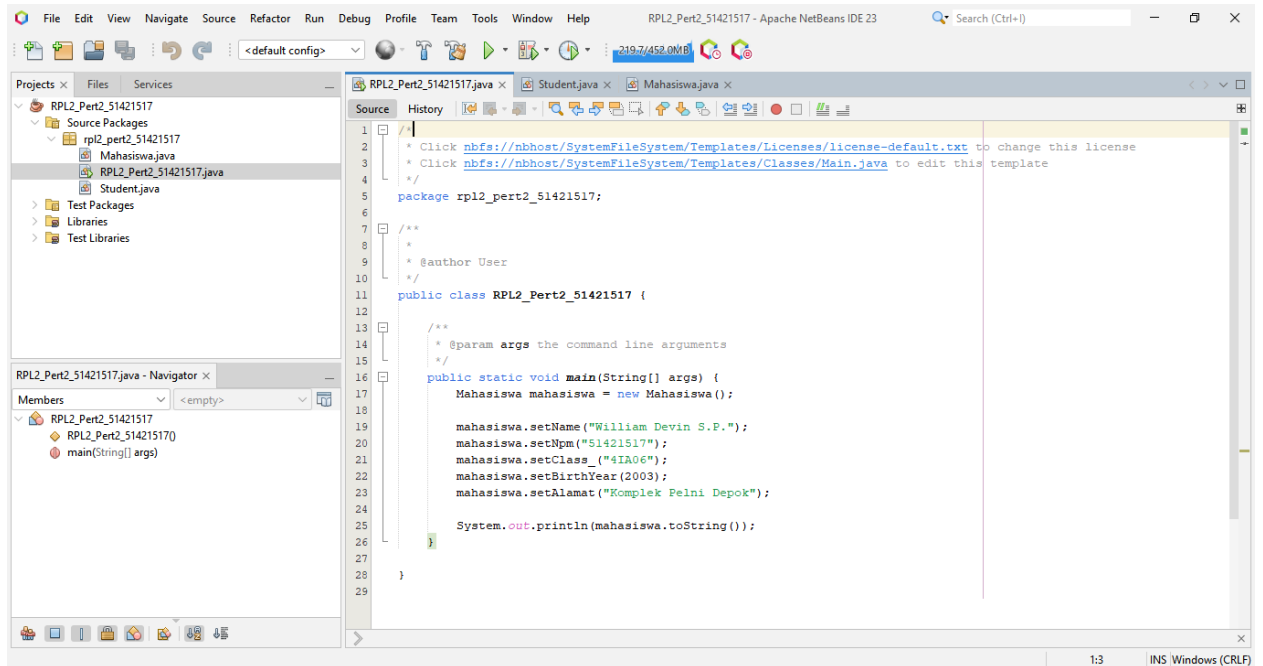
```
36 }
37
38 public void setName(String npm) {
39     this.npm = npm;
40 }
41
42 public String getClass_() {
43     return class_;
44 }
45
46 public void setClass_(String class_) {
47     this.class_ = class_;
48 }
49
50 public int getBirthYear() {
51     return birthYear;
52 }
53
54 public void setBirthYear(int birthYear) {
55     this.birthYear = birthYear;
56 }
57
58 public String getAlamat() {
59     return alamat;
60 }
61
62 public void setAlamat(String alamat) {
63     this.alamat = alamat;
64 }
65
66 public int calculateAge() {
67     return birthYear;
68 }
69 }
```



2. Mahasiswa.java :



3. RPL2_Pert2_51421517.java :



LOGIKA PROGRAM

1. Student.java :

```
import java.time.*;
```

Langkah pertama yang dilakukan adalah mengimpor modul java.time. Modul ini akan digunakan untuk menghitung usia berdasarkan tahun terkini.

Kemudian, langkah selanjutnya adalah membuat *class Student* beserta *constructor*-nya.

```
public class Student {
    private String name;
    private String npm;
    private String class_;
    private int birthYear;
    private String alamat;

    public Student() {
        this.name = "";
        this.npm = "";
        this.class_ = "";
        this.birthYear = 0;
        this.alamat = "";
    }
}
```

Identifier private pada setiap atribut berfungsi agar atribut-atribut tersebut tidak dapat diubah atau diakses secara langsung, sehingga memerlukan *method getter* dan *setter*. Kemudian, *constructor* dari *class* tersebut dibuat tanpa memerlukan parameter apa pun, dan menginisiasi nilai-nilai atribut dengan "" untuk atribut *string* dan 0 untuk atribut *int*.

Karena atribut-atribut pada *class* bersifat *private*, maka *method getter* dan *setter* harus didefinisikan untuk memungkinkan modifikasi ataupun akses terhadap atribut dari kelas tersebut.

```
public String getName() {  
    return name;  
}  
  
public void setName(String name) {  
    this.name = name;  
}  
  
public String getNpm() {  
    return npm;  
}  
  
public void setNpm(String npm) {  
    this.npm = npm;  
}  
  
public String getClass_() {  
    return class_;  
}  
  
public void setClass_(String class_) {  
    this.class_ = class_;  
}  
  
public int getBirthYear() {  
    return birthYear;  
}  
  
public void setBirthYear(int birthYear) {  
    this.birthYear = birthYear;  
}
```

```

public String getAlamat() {
    return alamat;
}

public void setAlamat(String alamat) {
    this.alamat = alamat;
}

```

Method-method getter hanya mengembalikan atribut terkait, sedangkan *method-method setter* menerima parameter atribut yang ingin dimodifikasi nilainya, dan kemudian mengganti nilai atribut dari *instance* dengan *statement this.attribute = value*.

Method terakhir yang didefinisikan adalah *calculateAge*, yang berfungsi untuk menghitung usia berdasarkan tahun lahir sebagai parameter dari *method* tersebut.

```

public int calculateAge() {
    int currentYear = Year.now().getValue();
    int age = currentYear - birthYear;
    return age;
}

```

2. Mahasiswa.java :

Class Mahasiswa merupakan *subclass* dari *class* Student sebagai *superclass*, sehingga menggunakan *keyword extends* untuk mendefinisikan *class* Student.

```

public class Mahasiswa extends Student {
    public Mahasiswa() {
        super();
    }

    @Override
    public String toString() {
        return "Nama\t\t: "+getName()+"\nNPM\t\t: "+
            getNpm()+"\nKelas\t\t: "+getClass_()+
            "\nTahun Lahir\t: "+getBirthYear()+
            "\nUmur\t\t: "+calculateAge()+
            "\nAlamat\t\t: "+getAlamat();
    }
}

```

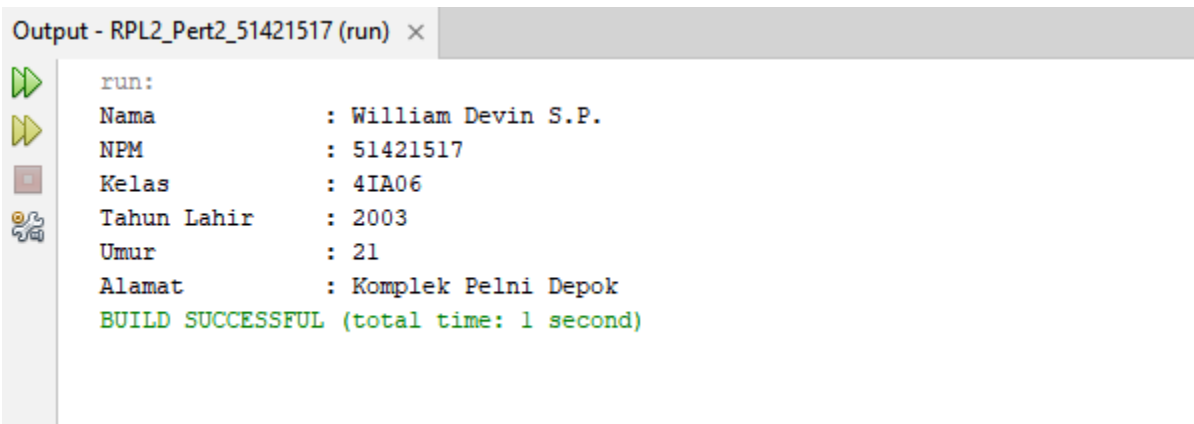
Method toString akan mengembalikan *String* yang berisi informasi atribut-atribut dari *class* dengan menggunakan *method getter*.

3. RPL2_Pert2_51421517.java :

Ini adalah *class Main* pada *project*, dan akan membuat *instance* Mahasiswa. Kemudian, karena *class* Mahasiswa adalah *subclass* dari *class Student*, maka *class* Mahasiswa mewarisi atribut-atribut dan *method-method* milik *class Student*. Kemudian, *method-method setter* dipanggil untuk mengisi nilai atribut-atribut dari *instance*. Lalu, hasilnya dicetak pada *output* menggunakan *method toString*.

```
public class RPL2_Pert2_51421517 {  
  
    public static void main(String[] args) {  
        Mahasiswa mahasiswa = new Mahasiswa();  
  
        mahasiswa.setName("William Devin S.P.");  
        mahasiswa.setNpm("51421517");  
        mahasiswa.setClass_("4IA06");  
        mahasiswa.setBirthYear(2003);  
        mahasiswa.setAlamat("Komplek Pelnı Depok");  
  
        System.out.println(mahasiswa.toString());  
    }  
}
```

OUTPUT PROGRAM



The screenshot shows an IDE output window titled "Output - RPL2_Pert2_51421517 (run) x". On the left, there are icons for running (green play button), debugging (yellow play button), and a red stop button. The output text is as follows:

```
run:  
Nama           : William Devin S.P.  
NPM            : 51421517  
Kelas         : 4IA06  
Tahun Lahir    : 2003  
Umur           : 21  
Alamat         : Komplek Pelnı Depok  
BUILD SUCCESSFUL (total time: 1 second)
```