

LAPORAN PRAKTIKUM

METODOLOGI PENGEMBANGAN

PERANGKAT LUNAK

MODUL 9



Versi 3.1

Penyusun : Tim Dosen MPPL

Nama : Radea Aji Prasajo
NIM : 064002200016
Nama Anggota : Radea Aji Prasajo (064002200016)
Aldi Surya Pranata (064002200039)
Albihan (064002200031)

Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Universitas Trisakti

2022

MODUL 9 : Pelaksanaan Tahapan Implementasi dan Pengujian Perangkat Lunak

Pokok Bahasan:

Pelaksanaan tahapan implementasi dan pengujian perangkat lunak.

Kode Pokok Bahasan:

IKS324.PRAK.2019.05

Melaksanakan tahapan implementasi perangkat lunak dengan melakukan instalasi dan menggunakan Eclipse IDE.

IKS324.PRAK.2019.06

Melaksanakan tahapan pengujian perangkat lunak.

No	Kemampuan Akhir Yang Diharapkan	Kriteria Penilaian dan Indikator	Jml Menit	Bobot Nilai (%)
1	- Mahasiswa mampu mengoperasikan IDE dengan benar (Sub CPMK 2.4).	Kriteria penilaian : Rubrik deskriptif. Indikator : Ketepatan waktu dalam menyelesaikan Laporan Praktikum 9 .	85	1,5
TOTAL			85	1,5

TUGAS PENDAHULUAN

Untuk dapat menjalankan modul praktikum ini silahkan membaca artikel berikut :

1. Eclipse
2. *Software testing*

LAB SETUP

Untuk dapat menjalankan praktikum ini maka harus disiapkan peralatan dan tempat pengambilan data sebagai berikut :

1. Studi kasus pengembangan perangkat lunak pada suatu organisasi / perusahaan / institusi
2. Eclipse

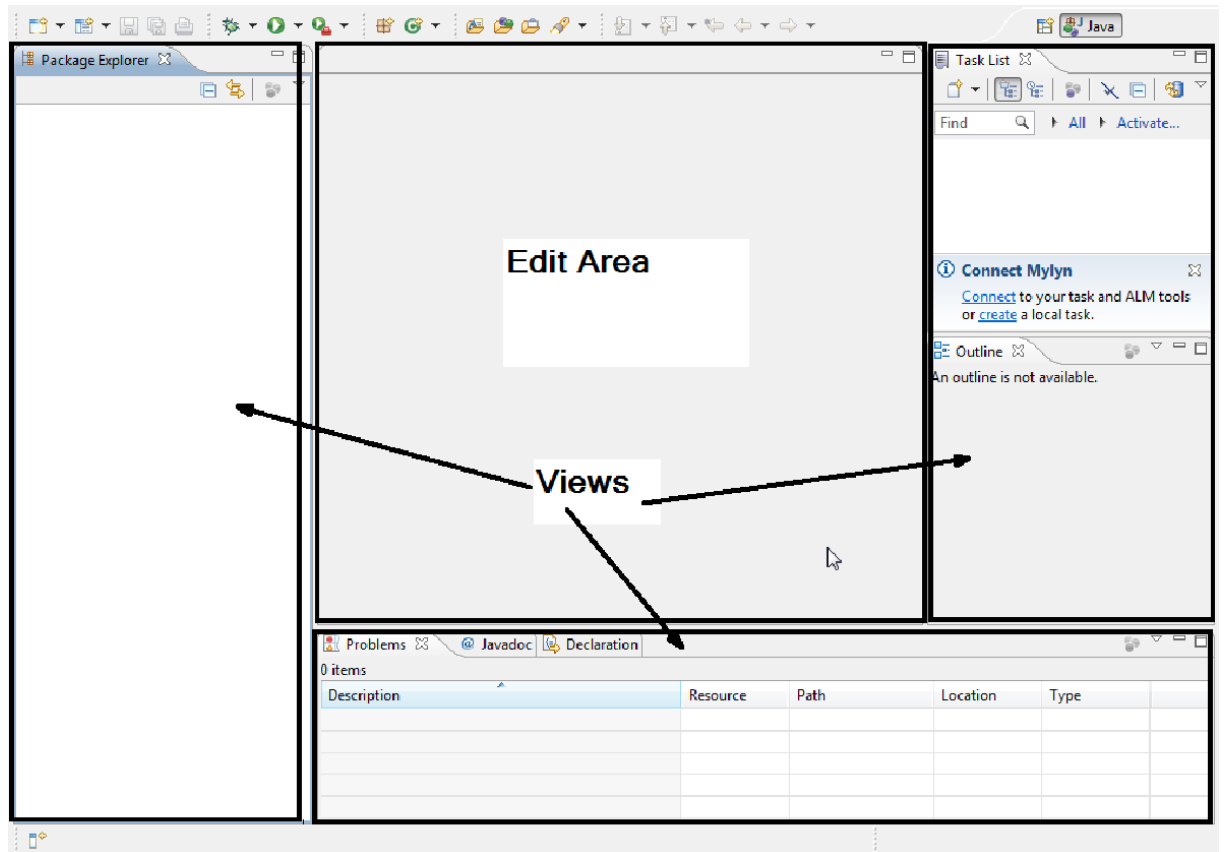
Eclipse merupakan sebuah *development environment* yang sangat *user-friendly* untuk membuat program JAVA. Eclipse sudah banyak dikenal dengan baik oleh banyak programmer JAVA. dan digunakan untuk berbagai pengembangan aplikasi. Sehingga mempelajari Eclipse dapat menambah wawasan dan pengetahuan tentang pemrograman JAVA.

Eclipse dikembangkan oleh Open Source Community. Proyek Eclipse dikelola oleh Eclipse Foundation. Untuk mengoperasikan Eclipse, dibutuhkan Java Runtime Environment (JRE). Eclipse berisikan berbagai macam komponen dan *plugin*, termasuk *JAVA compiler*.

Berikut tahapan untuk mengoperasikan Eclipse :

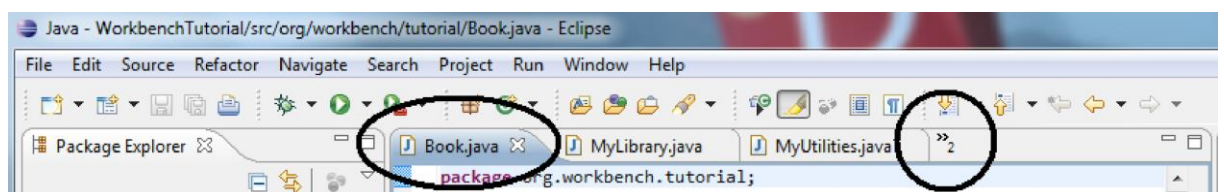
1. Pada komputer berbasis Windows, *double-click* pada *file* eclipse.exe. Sedangkan pada komputer berbasis Linux atau Mac, *double-click* pada *file* eclipse.
2. Ketika muncul *dialog box*, masukkan *pathname* untuk *workspace* (digunakan untuk menyimpan *JAVA Project*), dan tekan tombol **OK**.
3. Eclipse akan muncul dan menampilkan **Display Page**.
4. Tutup **Display Page**,

Eclipse menyediakan sebuah **Edit Area** dan **View**. **Edit Area** digunakan untuk memasukkan *JAVA source code*. Sedangkan **View** merupakan *sub-window* yang menampilkan informasi tentang *project*.



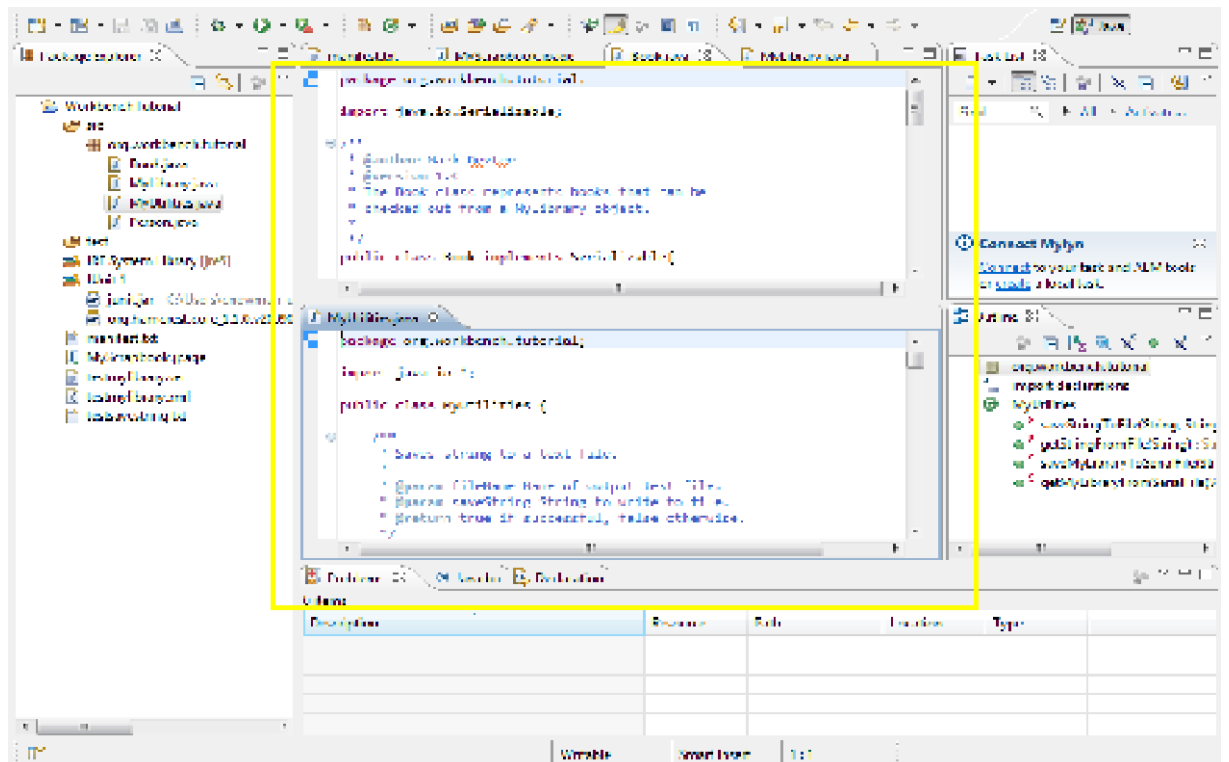
Gambar Edit Area dan View

Edit Area menggunakan tab jika Anda membuka lebih dari 1 *file*.



Gambar Tab Pada Edit Area

Edit Area dapat memiliki *multiple windows* pada *space* yang tersedia.

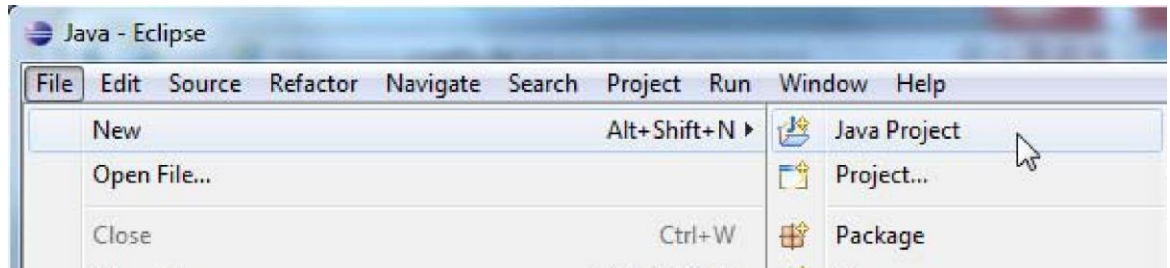


Gambar *Multiple Windows* Pada Edit Area

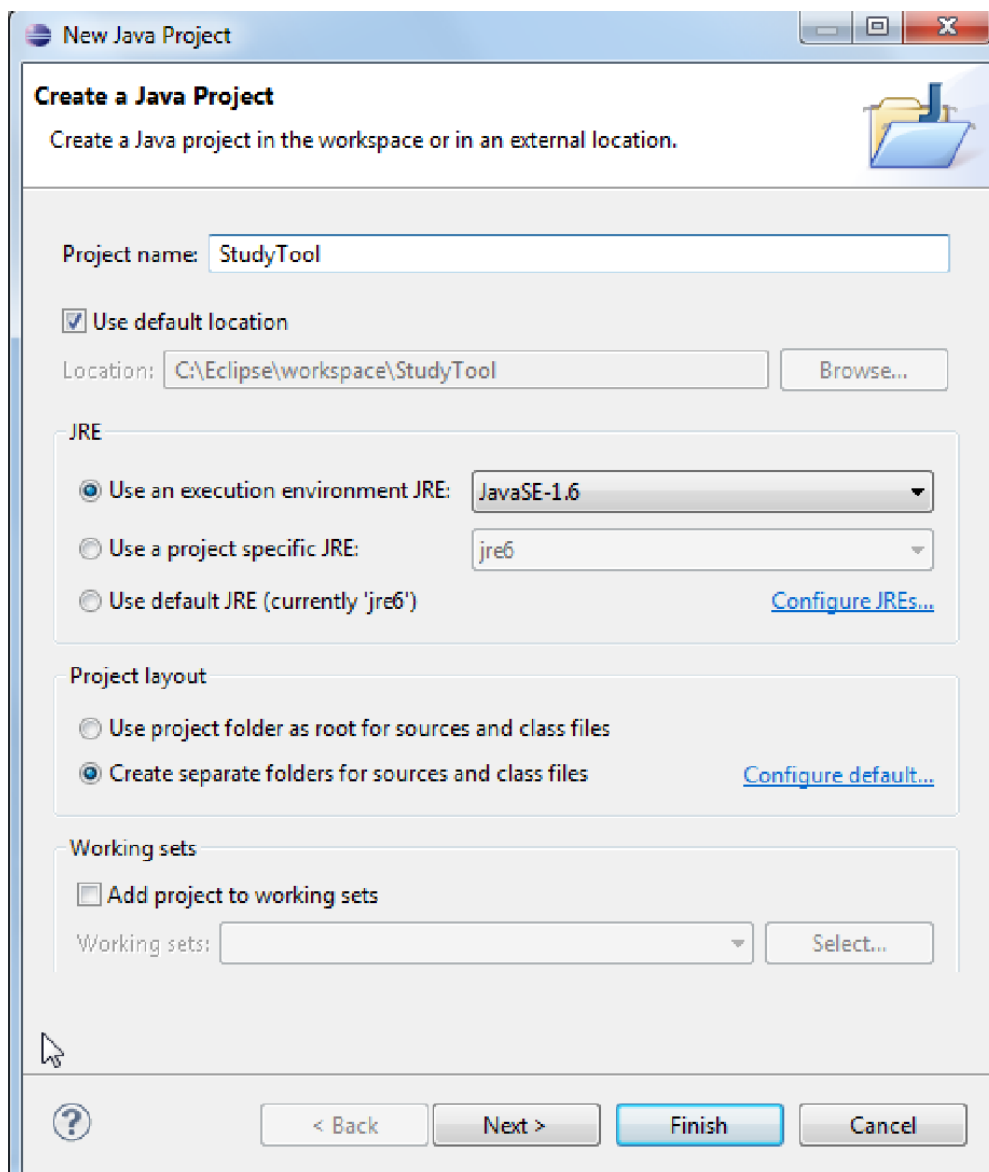
Langkah untuk membuat program JAVA :

1. **Buat Project**

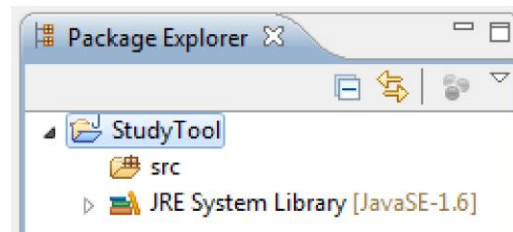
- Pilih **File** → **New** → **Java Project**



- Masukkan **Project Name** dan klik **Finish**

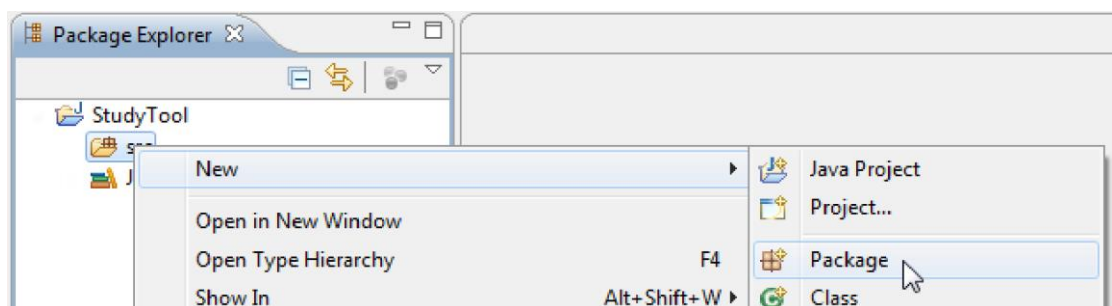


- Seluruh informasi tentang *project* di-set ke **Default** dan klik **Finish**
- Setelah itu, *project* sudah selesai dibuat dan ditampilkan sebagai folder
- **Package view** dapat dilihat di sebelah kiri dari **Edit Area**

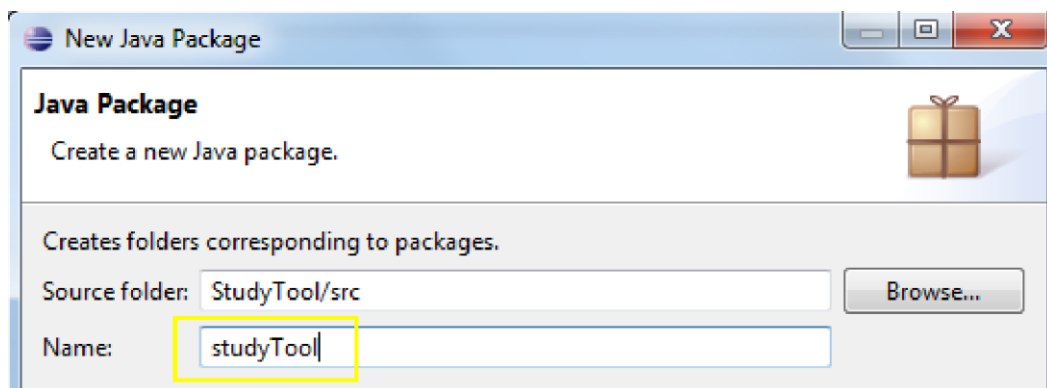


2. Buat **Package**

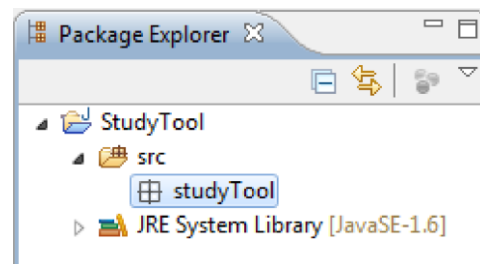
- Pada tab **Package Explorer**, pilih **<Nama Project>**, lalu klik kanan dan pilih **New → Package**



- Ketikkan nama *package* dan klik **Finish**

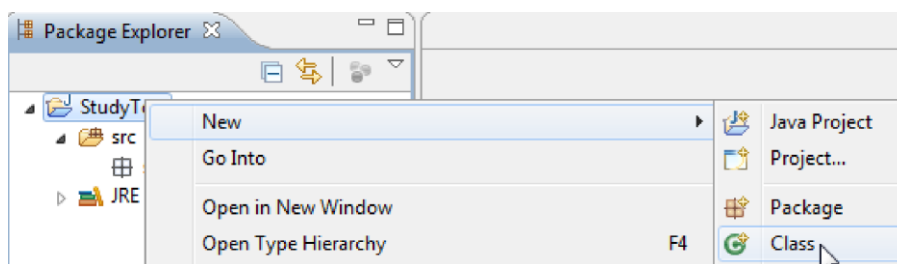


- Sehingga *package* tersebut akan tampil pada **Package Explorer**

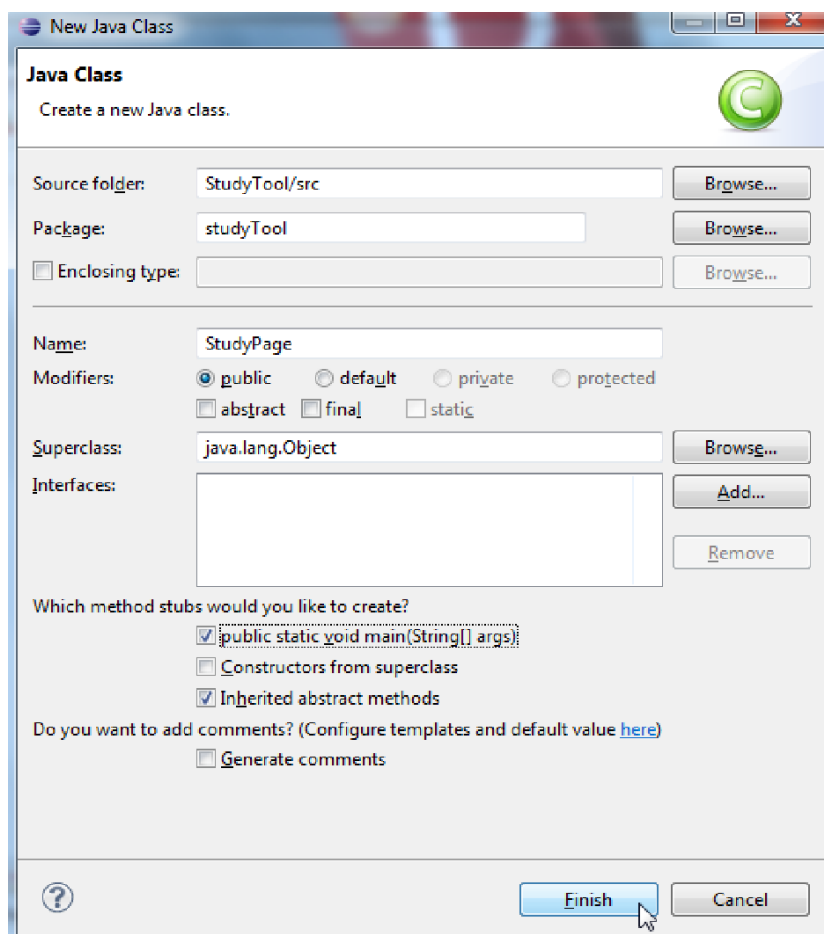


3. Pilih **Class**

- Klik kanan pada **<Nama Project>** dan pilih **New → Class**



- Ketikkan nama **Class** dan klik **Finish**

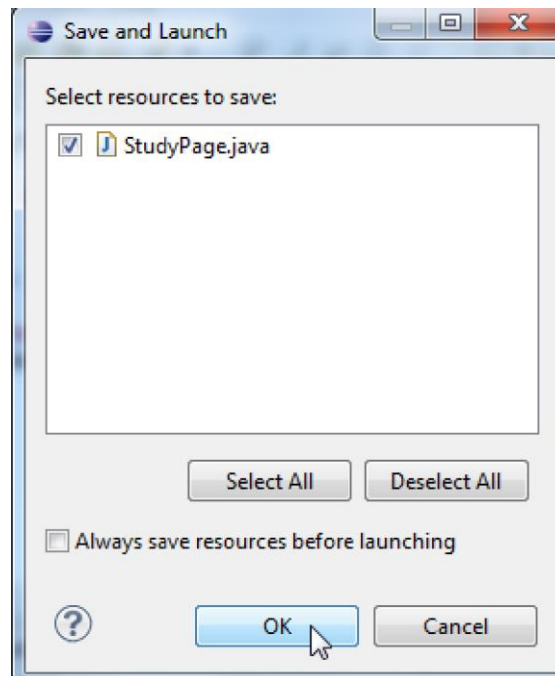


4. Create and run JAVA code

- Ketikkan *source code* pada **Main Class**

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println("Test Java Code");  
    }  
}
```

- Klik kanan pad *file* .java
- Pilih **Run As → Java Application**
- Simpan **Class** jika ditanyakan dengan mengklik **OK**



- Hasil akan ditampilkan pada **Console View**



TUGAS

Pelajari studi kasus. Kemudian lengkapi isian berikut ini.

Latihan 9.1. Membuat *Class*

File : <ReminderWater>.<.java>

NOTE : Extension diisi dengan .java

```
package MPPL;

public class ReminderWater {
    protected boolean Gender;
    protected double BeratBadan;
    protected int Usia;

    public ReminderWater(boolean Gender, double BeratBadan, int Usia) {
        this.Gender = Gender;
        this.BeratBadan = BeratBadan;
        this.Usia = Usia;
    }

    public void setGender(boolean Gender){
        this.Gender = Gender;
    }

    public void setBeratBadan(double BeratBadan){
        this.BeratBadan = BeratBadan;
    }

    public void setUsia(int Usia){
        this.Usia = Usia;
    }

    public boolean getGender(){
        return this.Gender;
    }

    public double getBeratBadan(){
        return this.BeratBadan;
    }

    public int getUsia(){
        return this.Usia;
    }
}
```

File : <Laporan >.<.java>

NOTE : Extension diisi dengan .java

```
package MPPL;

import java.util.*;

import static MPPL.Waktu.waktu;

public class Laporan {
    static int MaksimalAir = 2000;
    static boolean Jenis = true;
    static double Berat_BadanLaporan;
    static int UsiaLaporan;
    static boolean sembunyi = true;
    public static void main(String[] args){
        // Register
        Scanner Regis = new Scanner(System.in);
        boolean g = true;
        while(g == true){
            System.out.print("Gender(Laki-Laki/Perempuan) = ");
            String Gen = Regis.nextLine();
            Gen = Gen.toLowerCase();
            if (Gen.equals("laki-laki")){
                g = false;
            } else if (Gen.equals("perempuan")) {
                Jenis = false;
                g = false;
            } else {
                System.out.println("Kesalahan, silahkan ulangi lagi!");
            }
        }

        System.out.print("Berat Badan(kg) = ");
        Berat_BadanLaporan = Regis.nextDouble();

        System.out.print("Usia(tahun) = ");
        UsiaLaporan = Regis.nextInt();

        ReminderWater[] Data = new ReminderWater[1];
        Data[0] = new ReminderWater(Jenis, Berat_BadanLaporan, UsiaLaporan);

        // Laporan Air Minum
        int UkuranGelas, JumlahAir, PenguranganJumlahAir;
        boolean i = true;
        UkuranGelas = 100;
```

```

JumlahAir = 0;
double MaksimalAirMinum = Berat_BadanLaporan * 30;
Scanner inputmenu = new Scanner(System.in);

while (i){
    System.out.println();
    System.out.println("Jumlah Air: " + JumlahAir + "ml");
    System.out.println("Maksimal Kebutuhan Air: " + MaksimalAirMinum +
"ml");
    System.out.println("1. Menambah Jumlah Air.\n" +
        "2. Mengubah Gelas yang Digunakan.\n" +
        "3. Mengurangi Jumlah Air.\n" +
        "4. Riwayat Air Minum.\n" +
        "5. Pengaturan.\n" +
        "6. Keluar.\n\n");
    System.out.print("Pilih Menu: ");
    int pilihan = inputmenu.nextInt();

    switch (pilihan) {
        case 1 :
            //Nambah jumlah air
            JumlahAir = JumlahAir + UkuranGelas;
            System.out.printf("Menambahkan jumlah air: " + UkuranGelas + "ml"
+ "\n");
            System.out.println("Total Air yang telah diminum: " + JumlahAir);
            waktu();
            break;

        case 2 :
            //Gelas yang digunakan
            Scanner input = new Scanner(System.in);
            System.out.print("Masukkan gelas yang digunakan(dalam ml): ");
            UkuranGelas = input.nextInt();
            break;

        case 3 :
            //Pengurangan jumlah air
            PenguranganJumlahAir = UkuranGelas;
            JumlahAir -= PenguranganJumlahAir;
            System.out.println("Jumlah Air berkurang menjadi " + JumlahAir +
"ml.");
            break;

        case 4 :
            //Menampilkan Riwayat

```

```

        if (sembunyi == false){
            System.out.println("Riwayat Dirahasiakan!");
        } else {
            DisplayRiwayat(JumlahAir);
        }
        break;

case 5:
    //Pengaturan
    Scanner ubah = new Scanner(System.in);
    Scanner menulagi = new Scanner(System.in);
    boolean loop = true;

    while (loop == true){

        System.out.print("1. Sembunyikan Riwayat.\n" +
            "2. Ubah Data Pengguna.\n" +
            "3. Tentang Kami.\n" +
            "4. Keluar dari Pengaturan.\n\n" +
            "Pilih Pengaturan = ");
        int MenuPengaturan = menulagi.nextInt();

        switch (MenuPengaturan) {
            case 1:
                //Display Riwayat
                System.out.println("pilih(true/false?): ");
                sembunyi = ubah.nextBoolean();
                break;

            case 2:
                //Ubah Data
                System.out.println("Gender: " + Jenis);
                System.out.println("Berat Badan: " + Berat_BadanLaporan);
                System.out.println("Usia: " + UsiaLaporan + "\n");
                System.out.println("1. Mengubah Gender." +
                    "2. Mengubah Berat Badan." +
                    "3. Mengubah Usia." +
                    "4. Tidak ada yang diubah.");
                Scanner milih = new Scanner(System.in);
                boolean lakukanlagi = true;

                while (lakukanlagi == true){
                    int pilihlagi = milih.nextInt();
                    switch (pilihlagi){
                        case 1:

```

```

        Jenis = ubah.nextBoolean();
        System.out.println("Ada lagi?");
        break;
    case 2:
        Berat_BadanLaporan = ubah.nextDouble();
        System.out.println("Ada lagi?");
        break;
    case 3:
        UsiaLaporan = ubah.nextInt();
        System.out.println("Ada lagi?");
        break;
    case 4:
        lakukanlagi = false;
        break;
    }
}
break;

case 3:
    //Tentang Kami
    System.out.print("Radea Aji Prasajo (064002200016)\n" +
        "Aldi Surya Pranata (064002200039)\n" +
        "Albihan (064002200031)\n\n");
    break;

case 4:
    //Keluar dari Pengaturan
    loop = false;
    break;

    } break;
} break;

case 6:
    //Aplikasi Dihentikan
    i = false;
    break;
}
}
}
public static void DisplayRiwayat(int Air){
    System.out.println(Air + "ml ");
    waktu();
}
}
}

```

File : <Waktu>.<.java>

NOTE : Extension diisi dengan .java

```
package MPPL;

import java.time.format.DateTimeFormatter;
import java.time.LocalDateTime;

public class Waktu {

    public static void waktu(){
        DateTimeFormatter dtf = DateTimeFormatter.ofPattern("HH:mm:ss");
        LocalDateTime now = LocalDateTime.now();
        System.out.println(dtf.format(now));
    }
}
```

File : <Riwayat >.<.java >

NOTE : Extension diisi dengan .java

```
package MPPL;

import static MPPL.Waktu.waktu;

public class Riwayat {

    public static void DisplayRiwayat(int Air){
        System.out.println(Air + "ml");
        waktu();
    }
}
```

Latihan 9.2. Membuat *Test Case Scenario*

Project Name : Menyimpan Data Pengguna

Sub Module Name : <Menyimpan data untuk melakukan pemrosesan jumlah>

Type of Component : ☐ Screen
☐ Model
☐ Controller
☐ Form Printout / Report
☒ Struktur DB
☐ Lainnya : _____

List of Component Inspected :

Test Case ID	Description of Test Case	Expected Result	Actual Result	Pass or Fail
class ReminderWater		Mampu menyimpan data pengguna(Gender, Berat Badan, Usia, Waktu Tidur, dan Waktu Bangun) supaya bisa menghitung jumlah maksimum air yang dibutuhkan dan	Hanya bisa menyimpan tiga data(Gender, Berat Badan, dan Usia). Dalam perhitungan jumlah maksimum air yang dibutuhkan, hanya menggunakan	Pass

	<p>Input :</p> <pre> 1 package MPPL; 2 3 3 usages 4 public class ReminderWater { 5 3 usages 6 protected boolean Gender; 7 3 usages 8 protected double BeratBadan; 9 3 usages 10 protected int Usia; 11 12 1 usage 13 public ReminderWater(boolean Gender, double BeratBadan, int Usia) { 14 this.Gender = Gender; 15 this.BeratBadan = BeratBadan; 16 this.Usia = Usia; 17 } 18 19 no usages 20 public void setGender(boolean Gender) { this.Gender = Gender; } 21 22 no usages 23 public void setBeratBadan(double BeratBadan) { this.BeratBadan = BeratBadan; } 24 25 no usages 26 public void setUsia(int Usia) { this.Usia = Usia; } 27 28 no usages 29 public boolean getGender() { return this.Gender; } 30 31 no usages 32 public double getBeratBadan() { return this.BeratBadan; } 33 34 no usages 35 public int getUsia() { return this.Usia; } 36 37 } 38 39 </pre> <p>Action : Sebagai menyimpan data dan dapat dibuatkan objek. Serta melakukan <i>getter and setter</i>.</p>	<p>menentukan kapan waktunya untuk minum.</p>	<p>data Berat Badan yang digunakan. Serta tidak dapat kapan waktunya untuk minum.</p>	
--	---	---	---	--

Screenshot :

```
"C:\Program Files\Java\jdk-19\bin\bin\java.exe"  
Gender(Laki-Laki/Perempuan) = perempuan  
Berat Badan(kg) = 60  
Usia(tahun) = 25
```

Sub Module Name : <Melakukan pembuatan data baru, operasi perhitungan, mengubah sistem, dan menghentikan proses>

Type of Component : () Screen
(☒) Model
() Controller
(☒) Form Printout / Report
() Struktur DB
() Lainnya : _____

List of Component Inspected :

Test Case ID	Description of Test Case	Expected Result	Actual Result	Pass or Fail
class Laporan		Hanya melakukan operasi penambahan, pengurangan, dan menampilkan hasil operasi.	Dijadikan sebagai program utama untuk melakukan pembuatan data pengguna, operasi perhitungan, mengatur sistem aplikasi, dan	Pass

	<p>Input :</p> <pre> 1 package MPPL; 2 3 import java.util.*; 4 5 import static MPPL.Waktu.waktu; 6 7 public class Laporan { 8 static int MaksimalAir = 2000; 9 static boolean Jenis = true; 10 static double Berat_BadanLaporan; 11 static int UsiaLaporan; 12 static boolean sembunyi = true; 13 public static void main(String[] args){ 14 // Register 15 Scanner Regis = new Scanner(System.in); 16 boolean g = true; 17 while(g == true){ 18 System.out.print("Gender(Laki-Laki/Perempuan) = "); 19 String Gen = Regis.nextLine(); 20 Gen = Gen.toLowerCase(); 21 if (Gen.equals("laki-laki")){ 22 g = false; 23 } else if (Gen.equals("perempuan")) { 24 Jenis = false; 25 g = false; 26 } else { </pre>		menghentikan proses program.	
--	--	--	------------------------------	--

```

27         System.out.println("Kesalahan, silahkan ulangi lagi!");
28     }
29 }
30
31 System.out.print("Berat Badan(kg) = ");
32 Berat_BadanLaporan = Regis.nextDouble();
33
34 System.out.print("Usia(tahun) = ");
35 UsiaLaporan = Regis.nextInt();
36
37 ReminderWater[] Data = new ReminderWater[1];
38 Data[0] = new ReminderWater(Jenis, Berat_BadanLaporan, UsiaLaporan);
39
40 // Laporan Air Minum
41 int UkuranGelas, JumlahAir, PenguranganJumlahAir;
42 boolean i = true;
43 UkuranGelas = 100;
44 JumlahAir = 0;
45 double MaksimalAirMinum = Berat_BadanLaporan * 30;
46 Scanner inputmenu = new Scanner(System.in);
47
48 while (i){
49     System.out.println();
50     System.out.println("Jumlah Air: " + JumlahAir + "ml");
51     System.out.println("Maksimal Kebutuhan Air: " + MaksimalAirMinum + "ml");
52     System.out.println("1. Menambah Jumlah Air.\n" +
53         "2. Mengubah Gelas yang Digunakan.\n" +
54         "3. Mengurangi Jumlah Air.\n" +
55         "4. Riwayat Air Minum.\n" +
56         "5. Pengaturan.\n" +
57         "6. Keluar.\n\n");

```

Action :

Sebagai program utama(main method) yang digunakan untuk pembuatan data pengguna, operasi perhitungan, mengatur sistem aplikasi, dan menghentikan proses program.

Screenshot :

Jumlah Air: 0ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.

Pilih Menu: 2
Masukkan gelas yang digunakan(dalam ml): 250

Jumlah Air: 0ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.

Pilih Menu: 1
Menambahkan jumlah air: 250ml
Total Air yang telah diminum: 250
12:49:46

Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.

Pilih Menu: 1
Menambahkan jumlah air: 250ml
Total Air yang telah diminum: 500
12:49:49

Jumlah Air: 500ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.

Pilih Menu: 3
Jumlah Air berkurang menjadi 250ml.

Pilih Menu: 3
Jumlah Air berkurang menjadi 250ml.

Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.

Pilih Menu: 4
250ml
12:50:10

Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.

Pilih Menu: 5
1. Sembunyikan Riwayat.
2. Ubah Data Pengguna.
3. Tentang Kami.
4. Keluar dari Pengaturan.

Pilih Pengaturan = 1
pilih(true/false?):
true

Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.

Pilih Menu: 4
250ml
12:50:30

```
Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.
```

```
Pilih Menu: 5
1. Sembunyikan Riwayat.
2. Ubah Data Pengguna.
3. Tentang Kami.
4. Keluar dari Pengaturan.
```

```
Pilih Pengaturan = 1
pilih(true/false?):
false
```

```
Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.
```

```
Pilih Menu: 4
Riwayat Dirahasiakan!
```

```
Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.
```

```
Pilih Menu: 5
1. Sembunyikan Riwayat.
2. Ubah Data Pengguna.
3. Tentang Kami.
4. Keluar dari Pengaturan.
```

```
Pilih Pengaturan = 2
Gender: false
Berat Badan: 60.0
Usia: 25
```

```
1. Mengubah Gender.2. Mengubah Berat Badan.3. Mengubah Usia.4. Tidak ada yang diubah.
```

```
2
```

```
50
```

```
Ada lagi?
```

```
4
```

```
Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.
```

```
Pilih Menu: 5
```

```
1. Sembunyikan Riwayat.
2. Ubah Data Pengguna.
3. Tentang Kami.
4. Keluar dari Pengaturan.
```

```
Pilih Pengaturan = 3
```

```
Radea Aji Prasajo (064002200016)
Aldi Surya Pranata (064002200039)
Albihan (064002200031)
```

```
Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.
```

```
Pilih Menu: 5
```

```
1. Sembunyikan Riwayat.
2. Ubah Data Pengguna.
3. Tentang Kami.
4. Keluar dari Pengaturan.
```

```
Pilih Pengaturan = 4
```

```
Jumlah Air: 250ml
Maksimal Kebutuhan Air: 1800.0ml
1. Menambah Jumlah Air.
2. Mengubah Gelas yang Digunakan.
3. Mengurangi Jumlah Air.
4. Riwayat Air Minum.
5. Pengaturan.
6. Keluar.
```

```
Pilih Menu: 6
```

```
Process finished with exit code 0
```


Sub Module Name : <Menampilkan hasil perhitungan dan waktu sekarang>

Type of Component : (☒) Screen

(☐) Model

(☐) Controller

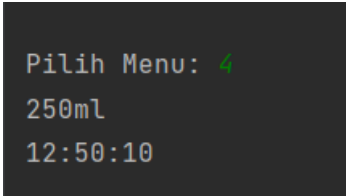
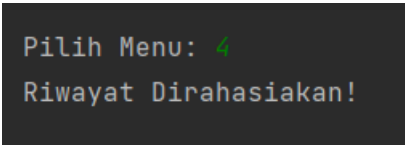
(☐) Form Printout / Report

(☐) Struktur DB

(☐) Lainnya : _____

List of Component Inspected :

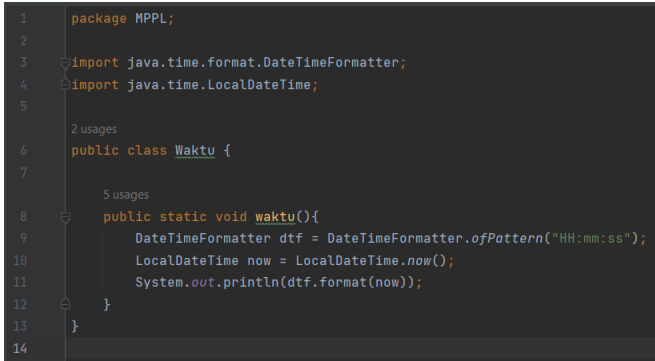
Test Case ID	Description of Test Case	Expected Result	Actual Result	Pass or Fail
Class Riwayat	Input : <pre> package MPPL; import static MPPL.Waktu.waktu; no usages public class Riwayat { no usages public static void DisplayRiwayat(int Air){ System.out.println(Air + "ml"); waktu(); } } </pre>	Menampilkan hasil laporan air minum dan waktu sekarang. Hasil laporan hari sebelumnya juga ditampilkan.	Hanya menampilkan hasil yang sekarang saja dan tidak dapat menampilkan hasil sebelumnya.	Pass

	Action : Menampilkan data hasil operasi perhitungan yang terdapat pada program utama, dan menampilkan operasi waktu yang terdapat pada program waktu.			
Screenshot : <div>   </div>				

Sub Module Name : <Melakukan operasi waktu>

Type of Component : () Screen
 () Model
 (☒) Controller
 () Form Printout / Report
 () Struktur DB
 () Lainnya : _____

List of Component Inspected :

Test Case ID	Description of Test Case	Expected Result	Actual Result	Pass or Fail
Class Waktu	<p>Input :</p>  <pre> 1 package MPPL; 2 3 import java.time.Format.DateTimeFormatter; 4 import java.time.LocalDateTime; 5 6 2 usages 7 public class Waktu { 8 9 5 usages 10 public static void waktu(){ 11 DateTimeFormatter dtf = DateTimeFormatter.ofPattern("HH:mm:ss"); 12 LocalDateTime now = LocalDateTime.now(); 13 System.out.println(dtf.format(now)); 14 } 15 } </pre> <p>Action :</p> <p>Melakukan operasi waktu dan menampilkan waktu sekarang.</p>	Operasi waktu dan menampilkan waktu sekarang telah ditambahkan di program utam sehingga tidak perlu membuat file/kode program yang baru.	Melakukan Operasi waktu dan menampilkan waktu sekarang di file/kode program yang baru.	Pass
Screenshot :				

Pilih Menu: 1

Menambahkan jumlah air: 250ml

Total Air yang telah diminum: 500

12:49:49