

# **LAPORAN PRAKTIKUM**

## **METODOLOGI PENGEMBANGAN**

### **PERANGKAT LUNAK**

#### **MODUL 9**



**Versi 3.1**

**Penyusun : Tim Dosen MPPL**

|             |          |                                |
|-------------|----------|--------------------------------|
| <b>Nama</b> | <b>:</b> | <b>Andri Martin</b>            |
|             |          | <b>Audi Aulia</b>              |
|             |          | <b>Bintang Rakha Daniswara</b> |
|             |          | <b>Rhena Tabella</b>           |
| <b>NIM</b>  | <b>:</b> | <b>064002200010</b>            |
|             |          | <b>064002200003</b>            |
|             |          | <b>064002200011</b>            |
|             |          | <b>064002200004</b>            |

**Jurusan Teknik Informatika**  
**Fakultas Teknologi Informasi**  
**Universitas Trisakti**

**2023**

# 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 ( <b>Sub CPMK 2.4</b> ). | <b>Kriteria penilaian :</b> Rubrik deskriptif.<br><br><b>Indikator :</b> Ketepatan waktu dalam menyelesaikan <b>Laporan Praktikum 9</b> . | 85        | 1,5             |
| <b>TOTAL</b> |  |   | <b>85</b> | <b>1,5</b>      |

## 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

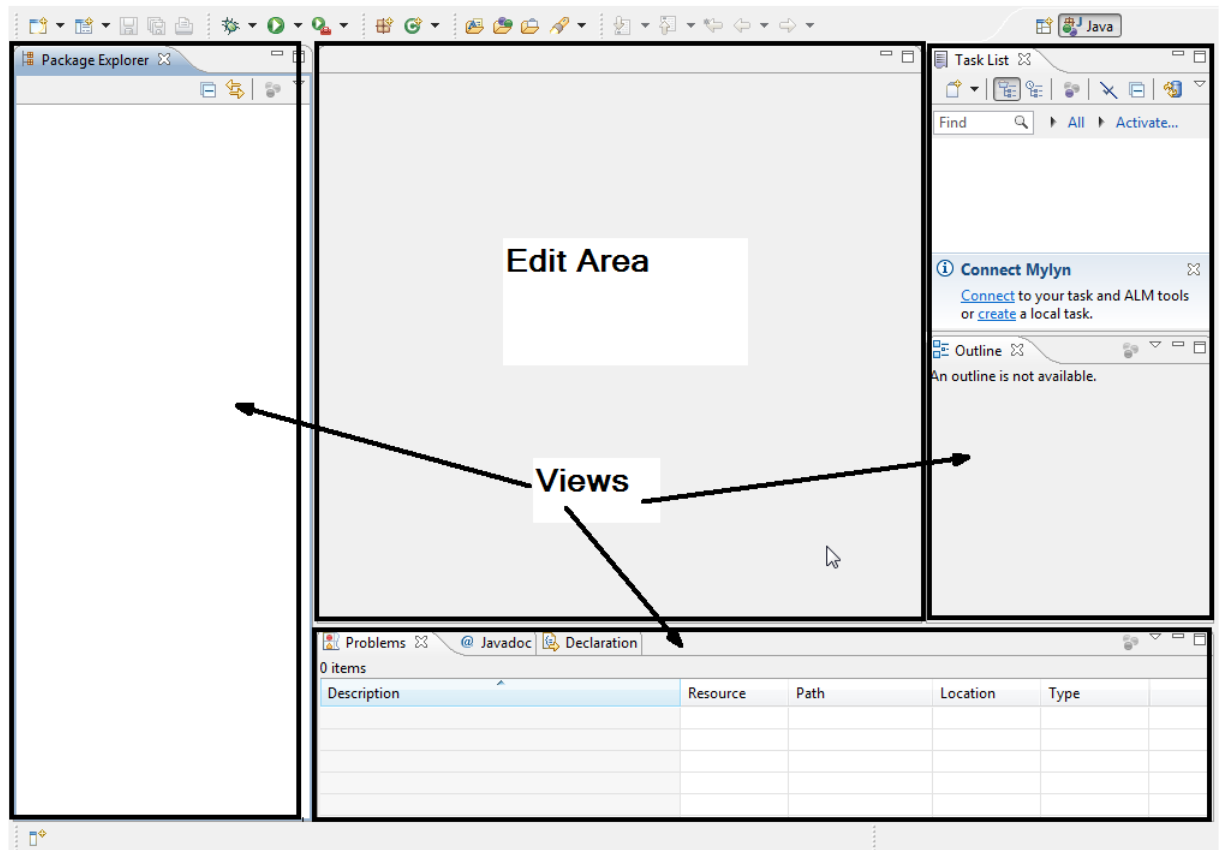
Eclipse merupakan sebuah *development environment* yang sangat *user-friendly* untuk membuat program JAVA. Eclipse sudah banyak dikenal dengan baik oleh banyak programmer JAVA. dan digunakan untuk berbagai pengembangan aplikasi. Sehingga mempelajari Eclipse dapat menambah wawasan dan pengetahuan tentang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

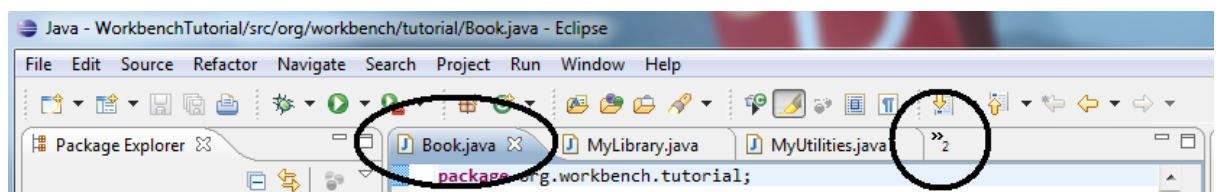
1. Pada komputer berbasis Windows, *double-click* pada *file* eclipse.exe. Sedangkan pada komputer berbasis Linux atau Mac, *double-click* pada *file* eclipse.
2. Ketika muncul *dialog box*, masukkan *pathname* untuk *workspace* (digunakan untuk menyimpan *JAVA Project*), dan tekan tombol **OK**.
3.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



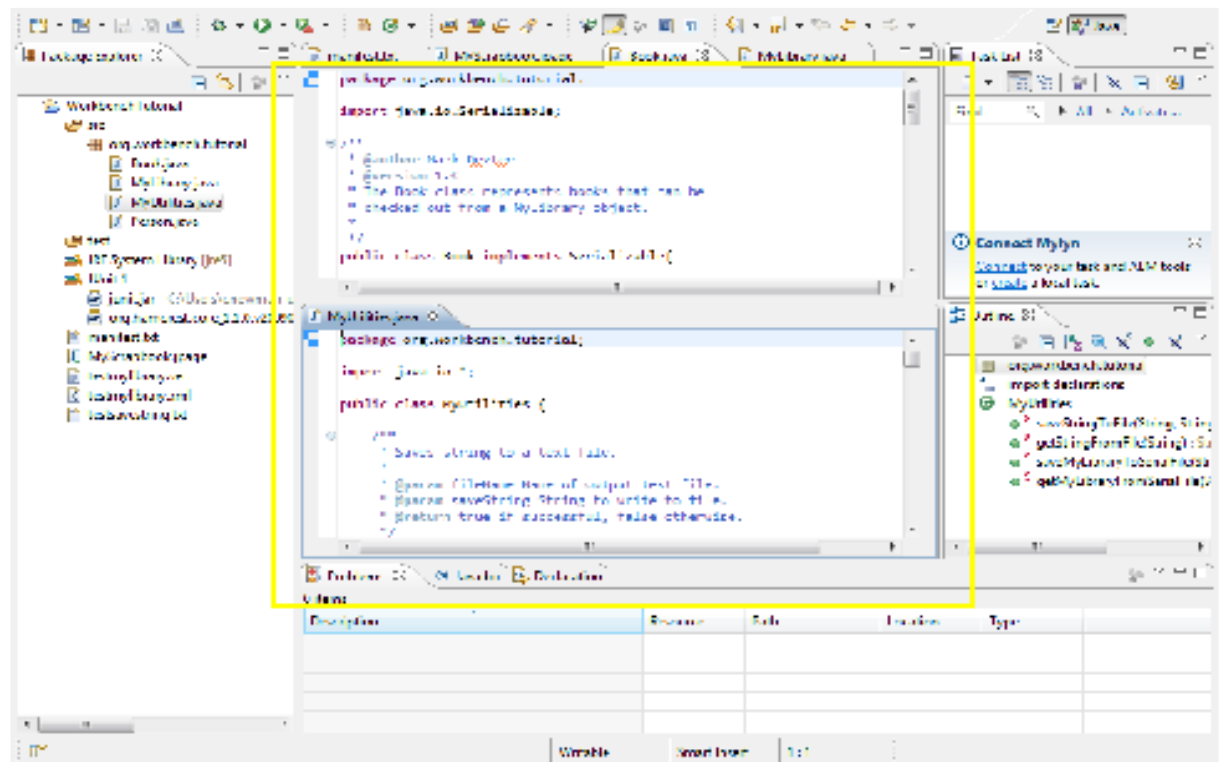
**Gambar Edit Area dan View**

**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.

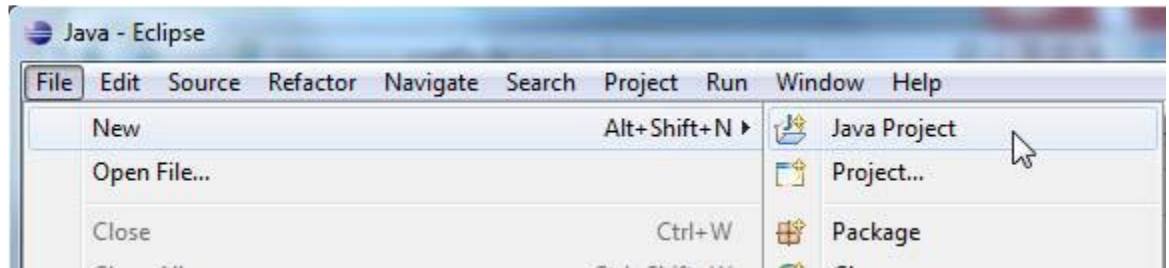


Gambar *Multiple Windows* Pada Edit Area

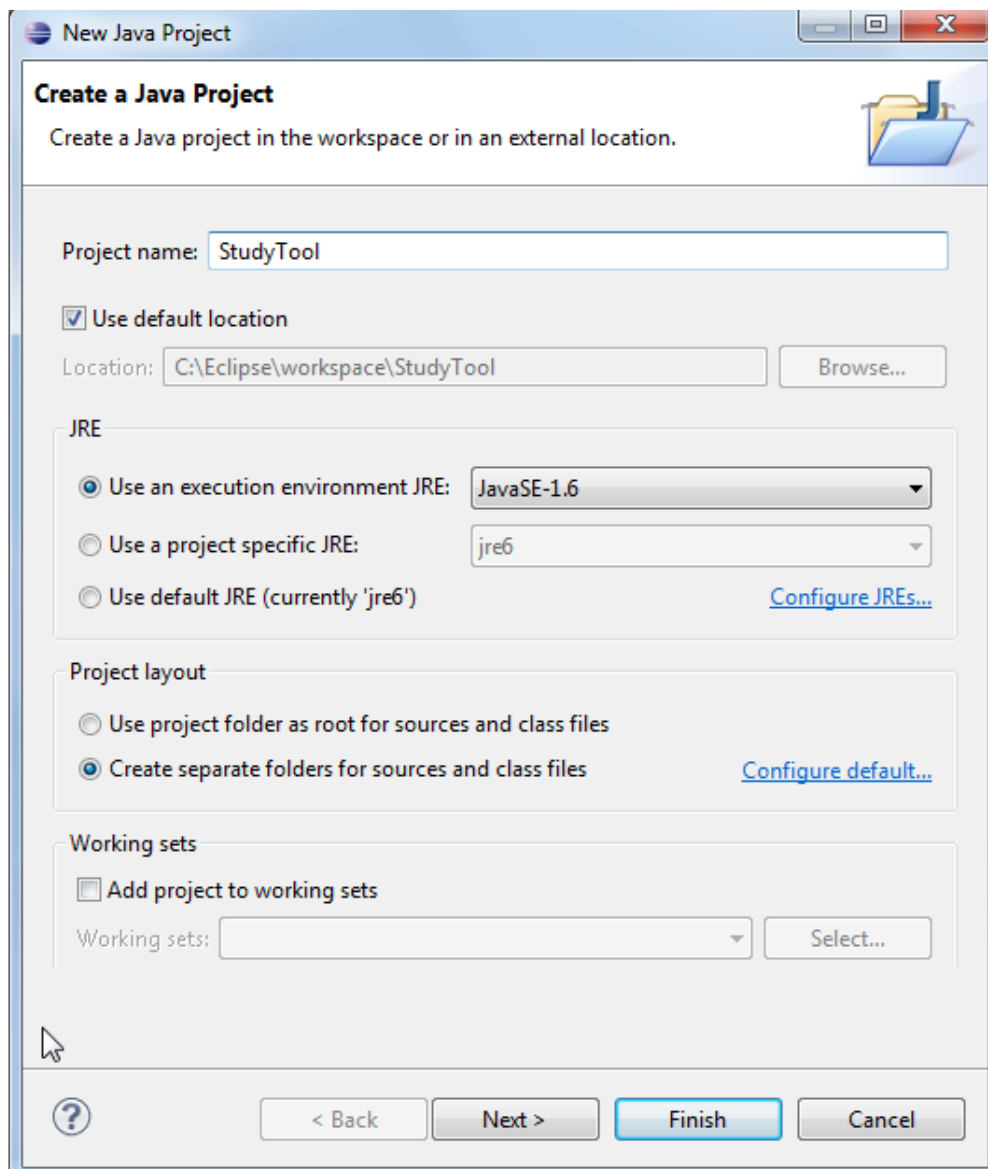
Langkah untuk membuat program JAVA :

1. **Buat Project**

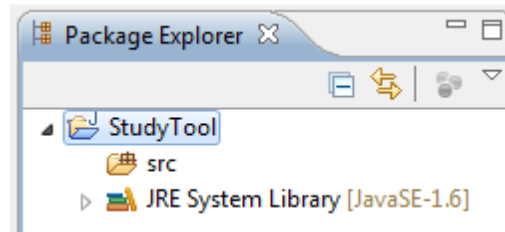
- Pilih **File** → **New** → **Java Project**



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

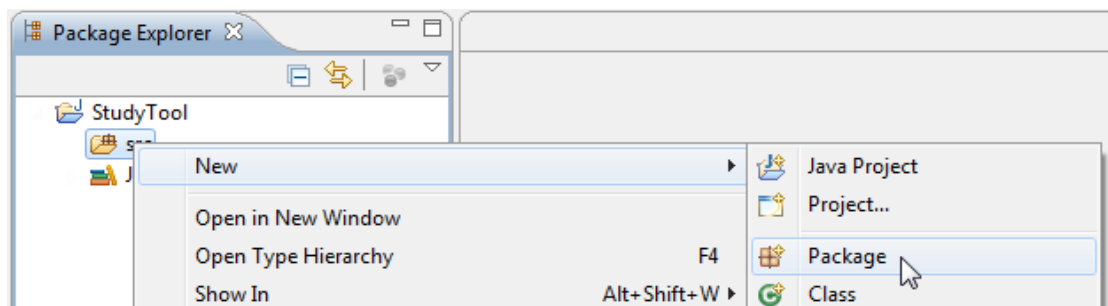


- Seluruh informasi tentang *project* di-set ke **Default** dan klik **Finish**
- Setelah itu, *project* sudah selesai dibuat dan ditampilkan sebagai folder
- **Package view** dapat dilihat di sebelah kiri dari **Edit Area**

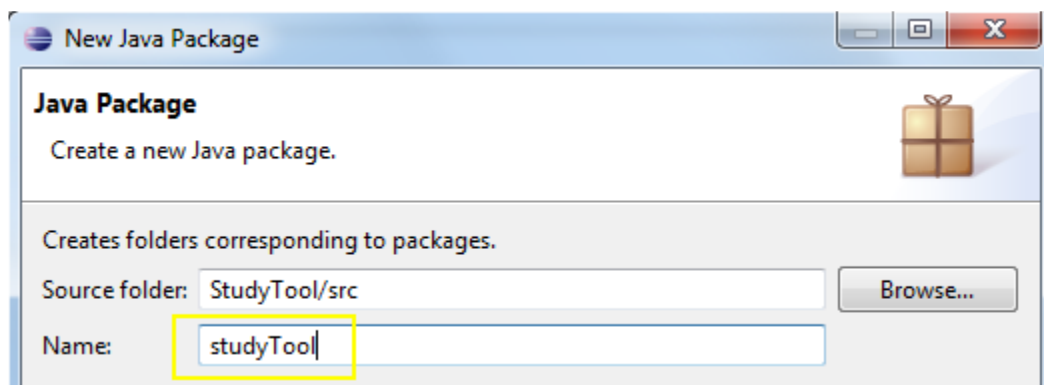


## 2. Buat Package

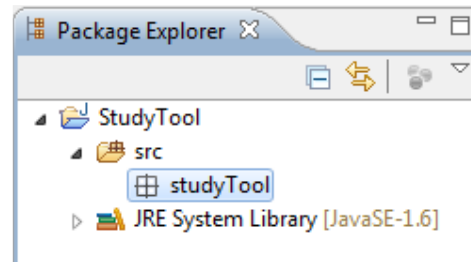
- Pada tab **Package Explorer**, pilih **<Nama Project>**, lalu klik kanan dan pilih **New → Package**



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

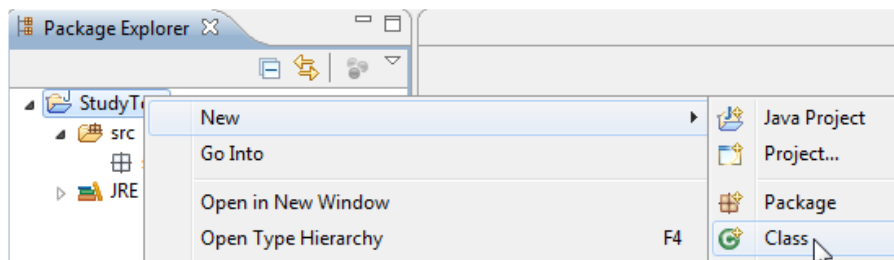


- Sehingga *package* tersebut akan tampil pada **Package Explorer**

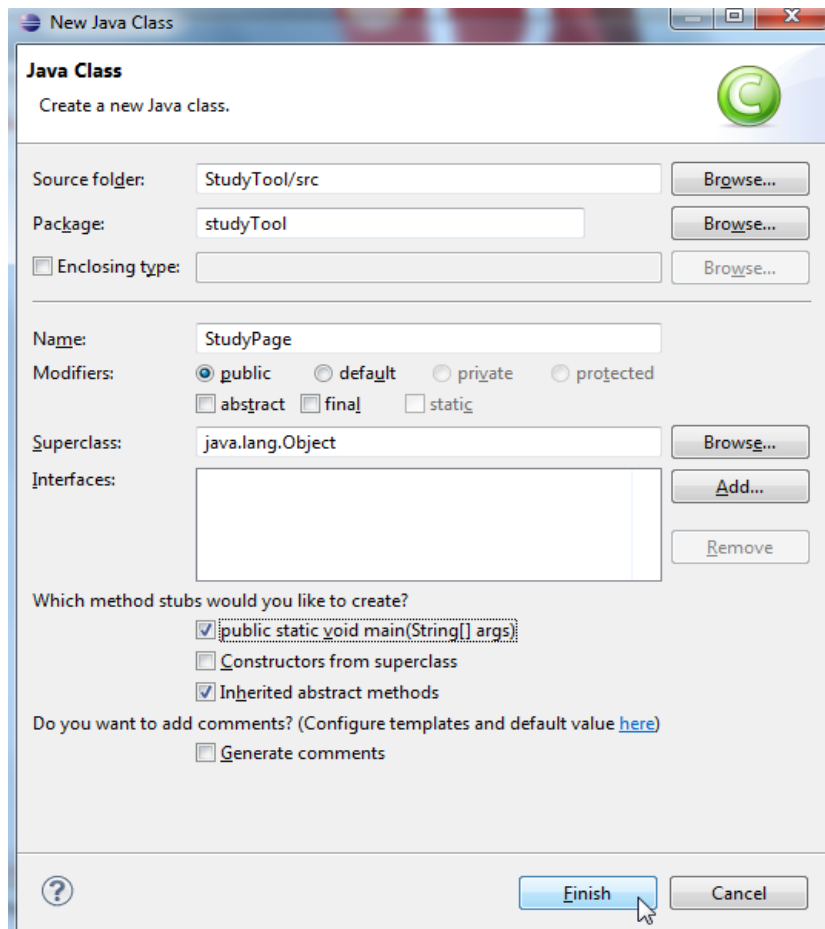


### 3. Pilih Class

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



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



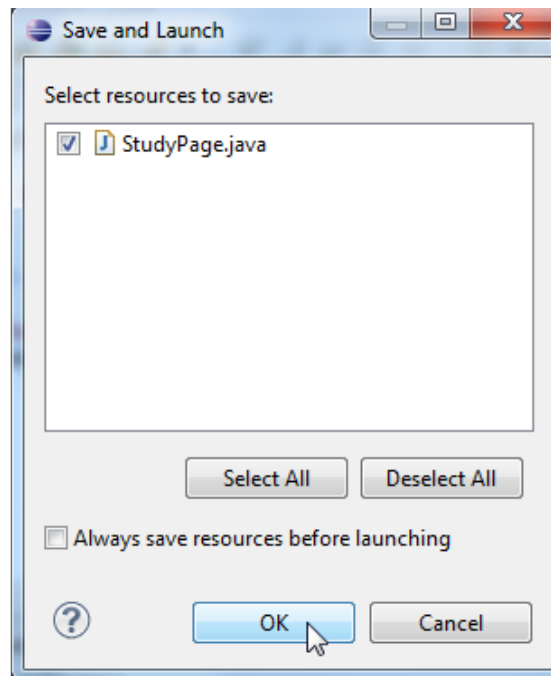


#### 4. Create and run JAVA code

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

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

- Klik kanan pada *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 : Login.java

**NOTE** : Extension diisi dengan .java

```
package StockIT;
import java.util.Scanner;

public class Login {
    public static void main(String[] args) {
        // Data pengguna yang valid
        String username = "admin";
        String password = "password";

        // Membaca input dari pengguna
        Scanner scanner = new Scanner(System.in);
        System.out.print("Login StockIT");
        System.out.print("\nUsername: ");
        String inputUsername = scanner.nextLine();
        System.out.print("Password: ");
        String inputPassword = scanner.nextLine();

        // Memeriksa kecocokan username dan password
        if (inputUsername.equals(username) &&
            inputPassword.equals(password)) {
            System.out.println("Login berhasil!");
            System.out.println("Terimakasih");
            // Lanjutkan dengan aplikasi setelah login
        } else {
            System.out.println("Username atau password salah.");
            // Lakukan tindakan jika login gagal
        }
    }
}
```

File : Notifikasi.java

**NOTE** : Extension diisi dengan .java

```
package StockIT;
import java.util.Scanner;

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

        Scanner scanner = new Scanner(System.in);
        System.out.print("Add Notifikasi Terbaru : ");
        String reminder = scanner.nextLine();

        System.out.println("Reminder Anda : " + reminder);
    }
}
```

File : Income.java

**NOTE :** Extension diisi dengan .java

```
package StockIT;
import java.text.DecimalFormat;
import java.util.Scanner;

public class Income {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Input data pengguna
        System.out.print("Masukkan pemasukan harian: ");
        double pemasukan = input.nextDouble();

        System.out.print("Masukkan pengeluaran harian: ");
        double pengeluaran = input.nextDouble();

        // Memanggil method untuk menghitung penghasilan
        double penghasilan = hitungPenghasilan(pemasukan,
        pengeluaran);

        // Format angka penghasilan dengan pemisah ribuan
        DecimalFormat decimalFormat = new
        DecimalFormat("#,###");
        String formattedPenghasilan =
        decimalFormat.format(penghasilan);

        // Menampilkan penghasilan
        System.out.println("Penghasilan Anda adalah: " +
        formattedPenghasilan);
    }

    public static double hitungPenghasilan(double pemasukan,
    double pengeluaran) {
        double penghasilan = pemasukan - pengeluaran;
        return penghasilan;
    }
}
```

File : Data\_Stok\_Bahan.java

**NOTE :** Extension diisi dengan .java

```
package StockIT;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Data_Stok_Bahan {
    private Map<String, Integer> stokBahan;

    public Data_Stok_Bahan() {
        stokBahan = new HashMap<>();
    }
}
```

```

public void tambahStokBahan(String namaBahan, int jumlah) {
    if (stokBahan.containsKey(namaBahan)) {
        int stokLama = stokBahan.get(namaBahan);
        int stokBaru = stokLama + jumlah;
        stokBahan.put(namaBahan, stokBaru);
    } else {
        stokBahan.put(namaBahan, jumlah);
    }
}

public int cekStokBahan(String namaBahan) {
    if (stokBahan.containsKey(namaBahan)) {
        return stokBahan.get(namaBahan);
    } else {
        return 0;
    }
}

public static void main(String[] args) {
    Data_Stok_Bahan dataStok = new Data_Stok_Bahan();
    Scanner scanner = new Scanner(System.in);

    boolean running = true;

    while (running) {
        System.out.println("Menu:");
        System.out.println("1. Tambah Stok Bahan");
        System.out.println("2. Cek Stok Bahan");
        System.out.println("3. Keluar");
        System.out.print("Pilih menu: ");
        int menu = scanner.nextInt();
        scanner.nextLine(); // Membersihkan newline
character

        switch (menu) {
            case 1:
                System.out.print("Masukkan nama bahan: ");
                String namaBahan = scanner.nextLine();
                System.out.print("Masukkan jumlah stok: ");
                int jumlahStok = scanner.nextInt();
                scanner.nextLine(); // Membersihkan newline
character

                dataStok.tambahStokBahan(namaBahan,
jumlahStok);

                System.out.println("Stok " + namaBahan + ": "
+ dataStok.cekStokBahan(namaBahan));
                break;

            case 2:
                System.out.print("Masukkan nama bahan yang
ingin dicek stoknya: ");
                String bahanCek = scanner.nextLine();
                System.out.println("Stok " + bahanCek + ": "
+ dataStok.cekStokBahan(bahanCek));
                break;

            case 3:
                running = false;

```

```

        System.out.println("Terima kasih!");
        break;

        default:
            System.out.println("Pilihan menu tidak
valid.");
            break;
    }
}
}
}

```

File : Tracking\_Pengiriman.java

**NOTE** : Extension diisi dengan .java

```

package StockIT;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Tracking_Pengiriman {
    private Map<String, String> pengiriman;
    private Map<String, Integer> stokBarang;

    public Tracking_Pengiriman() {
        pengiriman = new HashMap<>();
        stokBarang = new HashMap<>();
    }

    public void tambahPengiriman(String nomorResi, String
status) {
        pengiriman.put(nomorResi, status);
    }

    public void cekStatusPengiriman(String nomorResi) {
        String status = pengiriman.get(nomorResi);
        if (status != null) {
            System.out.println("Status pengiriman untuk nomor
resi " + nomorResi + ": " + status);
        } else {
            System.out.println("Nomor resi tidak ditemukan.");
        }
    }

    public void tambahStokBarang(String namaBarang, int jumlah)
{
        int stokLama = stokBarang.getOrDefault(namaBarang, 0);
        int stokBaru = stokLama + jumlah;
        stokBarang.put(namaBarang, stokBaru);
    }

    public void cekStokBarang(String namaBarang) {
        int stok = stokBarang.getOrDefault(namaBarang, 0);
        System.out.println("Stok barang " + namaBarang + ": " +
stok);
    }
}

```

```

    }

    public static void main(String[] args) {
        Tracking_Pengiriman tracking = new
Tracking_Pengiriman();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("=== Aplikasi Tracking Pengiriman
===");
            System.out.println("1. Tambah pengiriman");
            System.out.println("2. Cek status pengiriman");
            System.out.println("3. Tambah stok barang");
            System.out.println("4. Cek stok barang");
            System.out.println("5. Keluar");
            System.out.print("Pilih menu: ");

            int pilihan = scanner.nextInt();
            scanner.nextLine(); // Membersihkan newline
character

            switch (pilihan) {
                case 1:
                    System.out.print("Masukkan nomor resi: ");
                    String nomorResi = scanner.nextLine();
                    System.out.print("Masukkan status
pengiriman: ");
                    String status = scanner.nextLine();
                    tracking.tambahPengiriman(nomorResi,
status);
                    System.out.println("Pengiriman berhasil
ditambahkan.");
                    break;
                case 2:
                    System.out.print("Masukkan nomor resi yang
ingin dicek: ");
                    nomorResi = scanner.nextLine();
                    tracking.cekStatusPengiriman(nomorResi);
                    break;
                case 3:
                    System.out.print("Masukkan nama barang: ");
                    String namaBarang = scanner.nextLine();
                    System.out.print("Masukkan jumlah stok yang
ditambahkan: ");
                    int jumlah = scanner.nextInt();
                    scanner.nextLine(); // Membersihkan newline
character
                    tracking.tambahStokBarang(namaBarang,
jumlah);
                    System.out.println("Stok barang berhasil
ditambahkan.");
                    break;
                case 4:
                    System.out.print("Masukkan nama barang yang
ingin dicek stoknya: ");
                    namaBarang = scanner.nextLine();
                    tracking.cekStokBarang(namaBarang);
                    break;
                case 5:

```

```

        System.out.println("Terima kasih telah
menggunakan aplikasi.");
        System.exit(0);
        default:
            System.out.println("Pilihan tidak valid.");
    }

    System.out.println();
}
}
}
}

```

File : Menu.java

**NOTE** : Extension diisi dengan .java

```

package StockIT;
import java.util.Scanner;

public class Menu {
    public static void main(String[] args) {
        System.out.println("-----StockIT-----");
        System.out.println("-----");
        System.out.println("| 1. Data Stok Bahan |");
        System.out.println("| 2. Income |");
        System.out.println("| 3. Tracking |");
        System.out.println("| 4. Notifikasi |");
        System.out.println("-----");
    }
}

```

## Latihan 9.2. Membuat *Test Case Scenario*

**Project Name :** StockIT

|                                |
|--------------------------------|
| <b>Sub Module Name : Login</b> |
|--------------------------------|

**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 |
|--------------|---|--|--|--------------|
| Login        | Input : masukan username dan password<br><br>Action : system mengecek apakah username dan passwors benar. | Saat memasukan username dan password maka jika benar akan masuk kedalam aplikasi | Saat username dan password benar program akan memberikan bahwa username dan password benar | Pass         |
| Screenshot : |   |  |  |              |



```
Login StockIT
Username: admin
Password: password
Login berhasil!
Terimakasih
```

**Sub Module Name : Income**

**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 |
|---------------------|--|--|---|--------------|
| Income              | Input : memasukan pemasukan pada hari itu,<br>dan pengeluaran yang ada pada hari itu<br><br>Action : system akan memberikan proses<br>penghasilan harian | Dapat menghitung<br>income yang diterima | Program menampilkan<br>penghasilan harian | Pass         |
| <b>Screenshot :</b> |  |  |   |              |

```
terminasi, maka para pengguna, di bagian  
Masukkan pemasukan harian: 1000000  
Masukkan pengeluaran harian: 50000  
Penghasilan Anda adalah: 950,000  
|
```

**Sub Module Name : Notifikasi**

**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 |
|--------------|--|--|--|--------------|
| Notifikasi   | Input : menambahkan notifikasi kedalam sistem<br><br>Action : menyimpan notifikasi di sistem | Dapat mengeluarkan pengingat dalam bentuk notifikasi | Menambahkan notifikasi pada system dan memberikan pemberitahuan notifikasi berhasil dibuat | Pass         |

**Screenshot :**

```
Add Notifikasi Terbaru : tagihan listrik
Reminder Anda : tagihan listrik
```

|  |
|--|
| <b>Sub Module Name : Tracking Pengiriman</b> |
|--|

**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 |
|---------------------|---|--|---|--------------|
| Tracking_Pengiriman | Input : memasukan penambahan pengiriman<br><br>Action : menyimpan status pengiriman dengan resi | Dapat memberikan update status pengiriman, dan melacak | Menambahkan barang yang akan dikirim dan memberikan status pengiriman | Pass         |
| Screenshot :        |   |  |   |              |

```
=== Aplikasi Tracking Pengiriman ===
1. Tambah pengiriman
2. Cek status pengiriman
3. Tambah stok barang
4. Cek stok barang
5. Keluar
Pilih menu: 1
Masukkan nomor resi: 112
Masukkan status pengiriman: dijalan
Pengiriman berhasil ditambahkan.

=== Aplikasi Tracking Pengiriman ===
1. Tambah pengiriman
2. Cek status pengiriman
3. Tambah stok barang
4. Cek stok barang
5. Keluar
Pilih menu: 2
Masukkan nomor resi yang ingin dicek: 112
Status pengiriman untuk nomor resi 112: dijalan

=== Aplikasi Tracking Pengiriman ===
1. Tambah pengiriman
2. Cek status pengiriman
3. Tambah stok barang
4. Cek stok barang
5. Keluar
Pilih menu:
```

|  |
|--|
| <b>Sub Module Name : Data Stok Bahan</b> |
|--|

**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 |
|------------------------|---|---|---|--------------|
| <b>Data_Stok_Bahan</b> | Input : memasukan nama barang dan jumlahnya<br><br>Action : system menyimpan data bahan dan jumlahnya | Memberikan jumlah stok yang tersedia dan bahan apa saja | Menambahkan bahan dan jumlah bahan tersebut lalu memunculkan jumlah bahan | Pass         |
| <b>Screenshot :</b>    |   |   |   |              |

```
Menu:
1. Tambah Stok Bahan
2. Cek Stok Bahan
3. Keluar
Pilih menu: 1
Masukkan nama bahan: Tepung
Masukkan jumlah stok: 20
Stok Tepung: 20
Menu:
1. Tambah Stok Bahan
2. Cek Stok Bahan
3. Keluar
Pilih menu: 1
Masukkan nama bahan: Gula
Masukkan jumlah stok: 15
Stok Gula: 15
Menu:
1. Tambah Stok Bahan
2. Cek Stok Bahan
3. Keluar
Pilih menu: 2
Masukkan nama bahan yang ingin dicek stoknya: Gula
Stok Gula: 15
Menu:
1. Tambah Stok Bahan
2. Cek Stok Bahan
3. Keluar
Pilih menu: 3
Terima kasih!
```

|                               |
|-------------------------------|
| <b>Sub Module Name : Menu</b> |
|-------------------------------|

**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 |
|--------------|--|------------------------------------|---|--------------|
| Menu         | Input : tidak ada yang harus di lakukan user<br><br>Action : system akan meberikan tampilan menu | Memberikan tampilan menu dashboard | Saat di run akan menampilkan dashboard menu | Pass         |
| Screenshot : |  |                                    |   |              |



```
-----StockIT-----  
-----  
| 1. Data Stok Bahan |  
| 2. Income           |  
| 3. Tracking         |  
| 4. Notifikasi       |  
-----
```