

LAPORAN PRAKTIKUM
METODOLOGI PENGEMBANGAN
PERANGKAT LUNAK
MODUL 9



Versi 3.1

Penyusun : Tim Dosen MPPL

Nama : Tengku Rabih Razzan
Hashemi Ralf Koizumi
Ahmad RifQi Azis

NIM : 064002200028
064002200032
064002200019

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

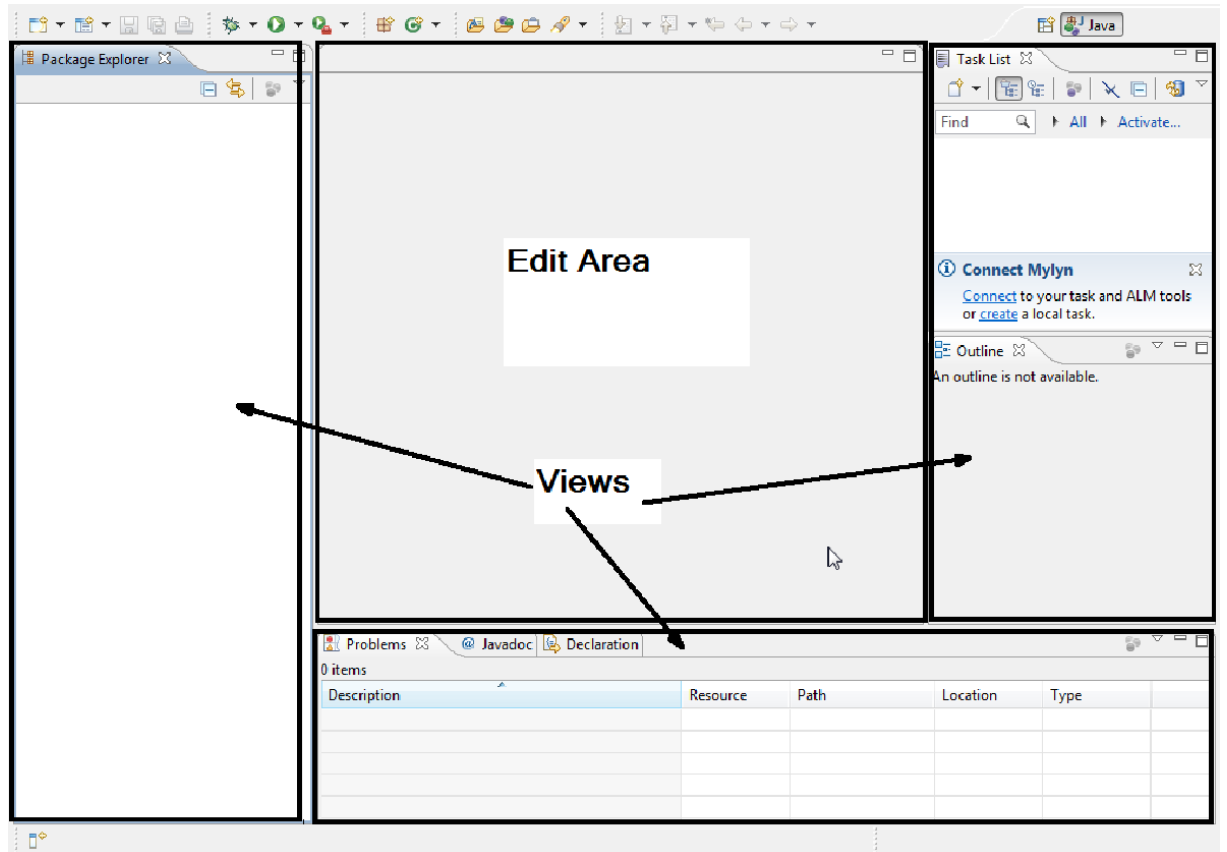
Eclipse merupakan sebuah *development environment* yang sangat *user-friendly* untuk membuat program JAVA. Eclipse sudah banyak dikenal dengan baik oleh banyak programmer JAVA. dan digunakan untuk berbagai pengembangan aplikasi. Sehingga mempelajari Eclipse dapat menambah wawasan dan pengetahuan tentang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

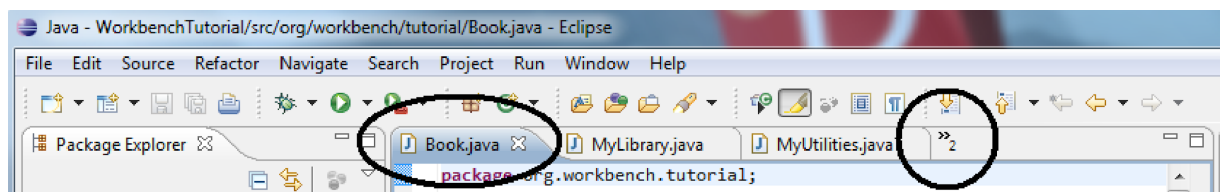
1. Pada komputer berbasis Windows, *double-click* pada *file* eclipse.exe. Sedangkan pada komputer berbasis Linux atau Mac, *double-click* pada *file* eclipse.
2. Ketika muncul *dialog box*, masukkan *pathname* untuk *workspace* (digunakan untuk menyimpan *JAVA Project*), dan tekan tombol **OK**.
3.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



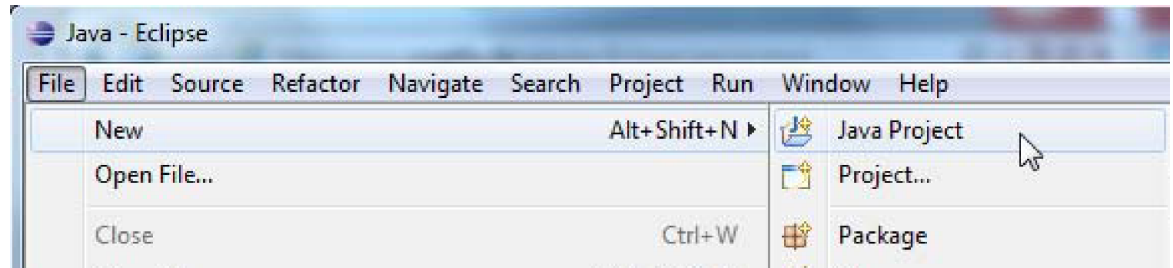
Gambar Tab Pada Edit Area

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

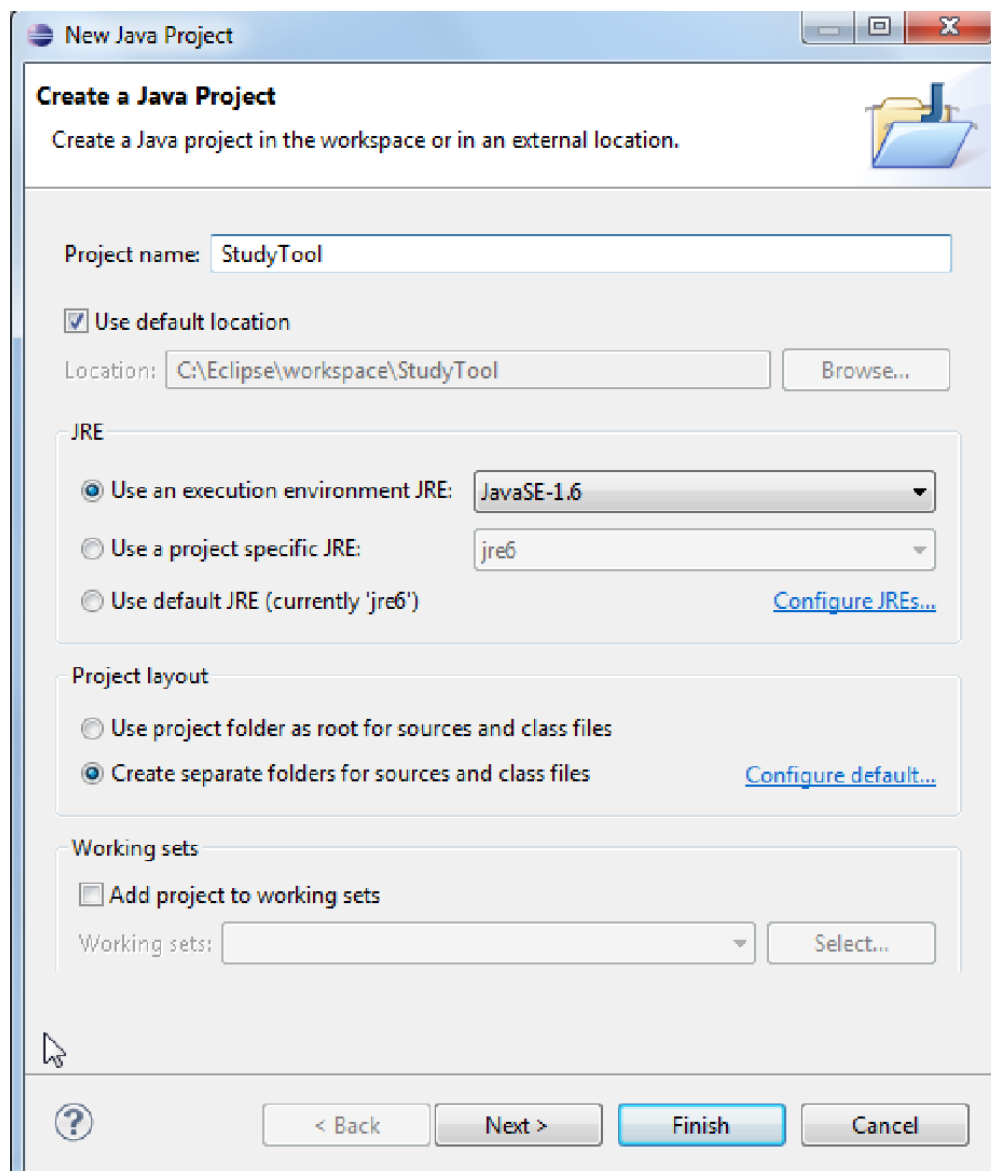
Langkah untuk membuat program JAVA :

1. **Buat Project**

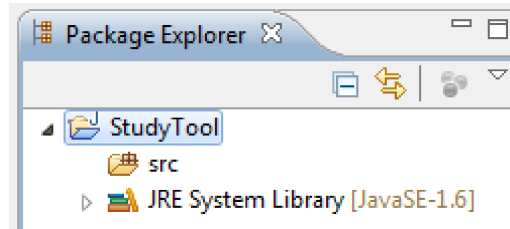
- Pilih **File** → **New** → **Java Project**



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

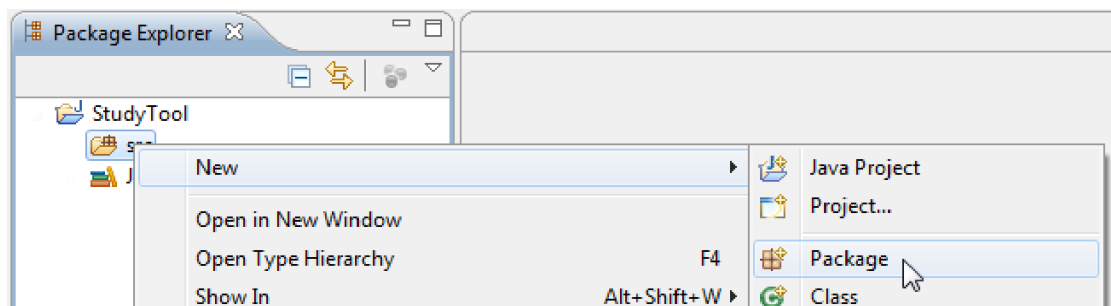


- Seluruh informasi tentang *project* di-set ke **Default** dan klik **Finish**
- Setelah itu, *project* sudah selesai dibuat dan ditampilkan sebagai folder
- **Package view** dapat dilihat di sebelah kiri dari **Edit Area**

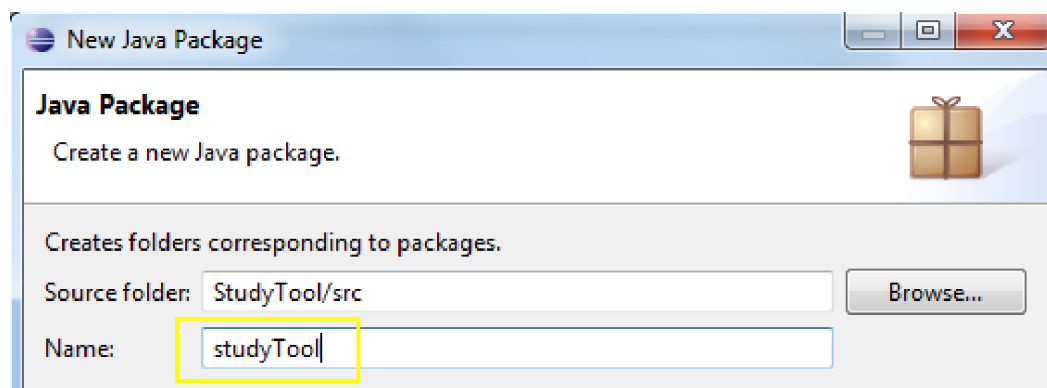


2. Buat **Package**

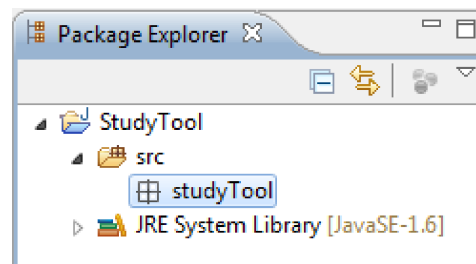
- Pada tab **Package Explorer**, pilih **<Nama Project>**, lalu klik kanan dan pilih **New → Package**



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

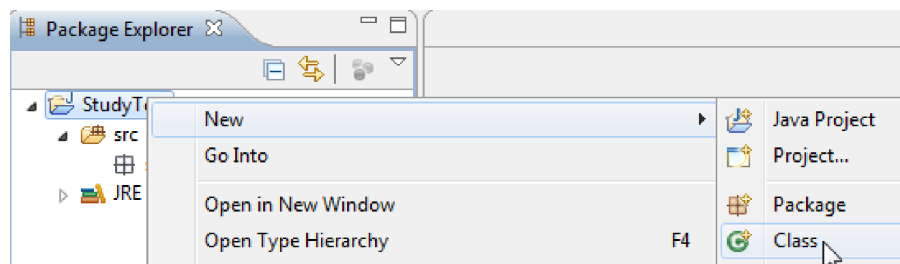


- Sehingga *package* tersebut akan tampil pada **Package Explorer**

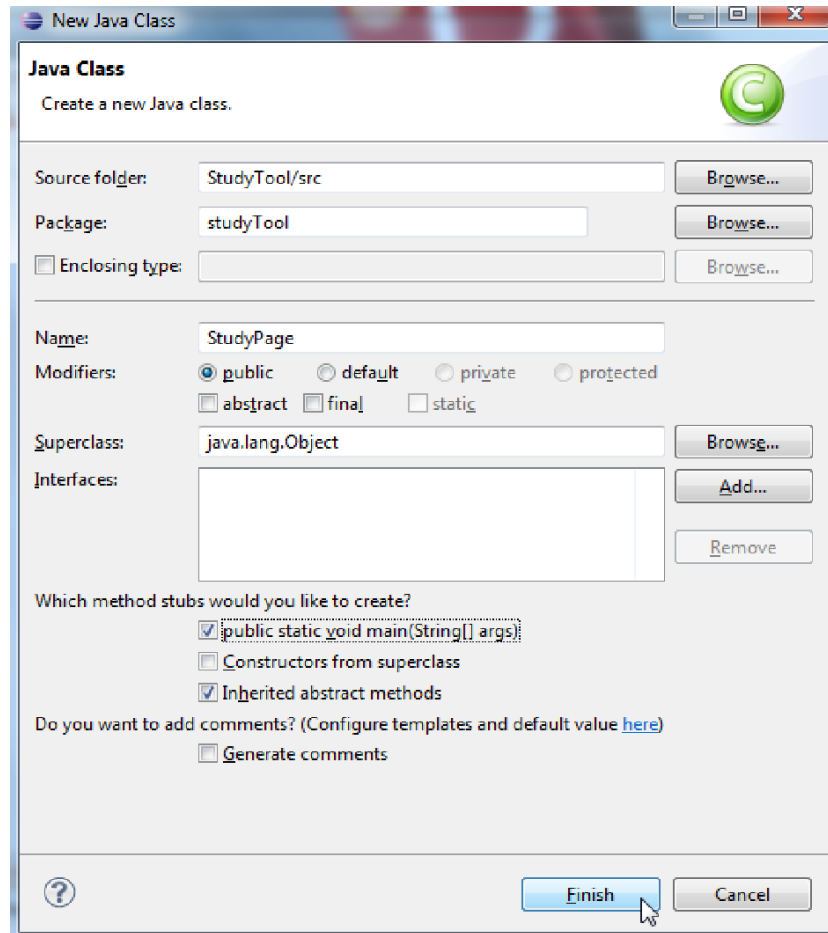


3. Pilih **Class**

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



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

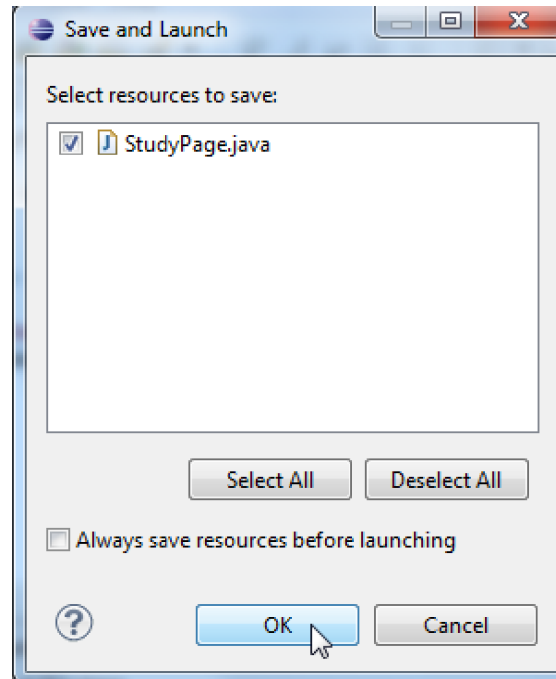


4. Create and run JAVA code

- 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 : <JaskomControllert>.<java>

NOTE : Extension diisi dengan .java

```
public class JaskomController {
    private JaskomModel model;
    private JaskomView view;

    public JaskomController(JaskomModel model, JaskomView view) {
        this.model = model;
        this.view = view;
    }

    public void jalankan() {
        int pilihan;
        do {
            pilihan = view.tampilkanMenuUtama();
            switch (pilihan) {
                case 1:
                    login();
                    break;
                case 2:
                    register();
                    break;
                case 0:
                    view.tampilkanLogoutBerhasil();
                    break;
                default:
                    view.tampilkanMenuTidakValid();
            }
        } while (pilihan != 0);
    }

    public void login() {
        view.tampilkanLogin();
        String username = view.getInputUsername();
        String password = view.getInputPassword();

        if (model.cekLogin(username, password)) {
            view.tampilkanLoginBerhasil();
            dashboard();
        } else {
            view.tampilkanLoginGagal();
            String registerChoice = view.getRegisterChoice();
            if (registerChoice.equalsIgnoreCase("ya")) {
```

```

        register();
    }
}

public void register() {
    view.tampilkanRegister();
    String newUsername = view.getNewUsername();
    String newPassword = view.getNewPassword();

    if (model.cekUsername(newUsername)) {
        view.tampilkanUsernameSudahDigunakan();
    } else {
        model.tambahUser(newUsername, newPassword);
        view.tampilkanRegistrasiBerhasil();
    }
}

public void dashboard() {
    int pilihan;
    do {
        pilihan = view.tampilkanDashboard();
        switch (pilihan) {
            case 1:
                pilihLayananPembersihan();
                break;
            case 2:
                pilihLayananPengecekan();
                break;
            case 3:
                pilihLayananPerbaikan();
                break;
            case 0:
                view.tampilkanLogoutBerhasil();
                break;
            default:
                view.tampilkanMenuTidakValid();
        }
    } while (pilihan != 0);
}

public void pilihLayananPembersihan() {
    view.tampilkanPembersihan();
    String thermalPasteChoice = view.getThermalPasteChoice();
    if (thermalPasteChoice.equalsIgnoreCase("ya")) {

```

```

        String lokasi = view.getLokasi();
        String nomorOrder =
model.pesanPembersihanDenganThermalPaste(lokasi);
        view.tampilkanNomorOrderPembersihan(nomorOrder);
    } else if (thermalPasteChoice.equalsIgnoreCase("tidak")) {
        String lokasi = view.getLokasi();
        String nomorOrder =
model.pesanPembersihanTanpaThermalPaste(lokasi);

view.tampilkanNomorOrderPembersihanTanpaThermalPaste(nomorOrder);
    } else {
        view.tampilkanPilihanTidakValid();
    }
}

public void pilihLayananPengecekan() {
    view.tampilkanPengecekan();
    String nomorOrder = model.pesanPengecekan();
    view.tampilkanNomorOrderPengecekan(nomorOrder);
}

public void pilihLayananPerbaikan() {
    view.tampilkanPerbaikan();
    String lanjutkanPerbaikanChoice = view.getLanjutkanPerbaikanChoice();
    if (lanjutkanPerbaikanChoice.equalsIgnoreCase("ya")) {
        String nomorOrderPengecekanSebelumnya =
view.getNomorOrderPengecekanSebelumnya();
        if
(model.verifikasiPengecekanSebelumnya(nomorOrderPengecekanSebelumnya)) {
            String nomorOrder =
model.pesanPerbaikan(nomorOrderPengecekanSebelumnya);
            view.tampilkanNomorOrderPerbaikan(nomorOrder);
        } else {
            view.tampilkanVerifikasiPengecekanSebelumnyaGagal();
        }
    } else if (lanjutkanPerbaikanChoice.equalsIgnoreCase("tidak")) {
        view.tampilkanPerbaikanDibatalkan();
    } else {
        view.tampilkanPilihanTidakValid();
    }
}
}
}

```



File : <JaskomModel>.<.java>

NOTE : Extension diisi dengan .java

```
import java.util.ArrayList;
import java.util.List;

public class JaskomModel {
    private List<User> users;
    private List<String> nomorOrderPembersihan;
    private List<String> nomorOrderPengecekan;
    private List<String> nomorOrderPerbaikan;

    public JaskomModel() {
        users = new ArrayList<>();
        nomorOrderPembersihan = new ArrayList<>();
        nomorOrderPengecekan = new ArrayList<>();
        nomorOrderPerbaikan = new ArrayList<>();
    }

    public boolean cekLogin(String username, String password) {
        for (User user : users) {
            if (user.getUsername().equals(username) &&
                user.getPassword().equals(password)) {
                return true;
            }
        }
        return false;
    }

    public boolean cekUsername(String username) {
```

```

        for (User user : users) {
            if (user.getUsername().equals(username)) {
                return true;
            }
        }
        return false;
    }

    public void tambahUser(String username, String password) {
        User user = new User(username, password);
        users.add(user);
    }

    public String pesanPembersihanDenganThermalPaste(String lokasi) {
        String nomorOrder = generateNomorOrder("PO");
        nomorOrderPembersihan.add(nomorOrder);
        // ...
        return nomorOrder;
    }

    public String pesanPembersihanTanpaThermalPaste(String lokasi) {
        String nomorOrder = generateNomorOrder("PO");
        nomorOrderPembersihan.add(nomorOrder);
        // ...
        return nomorOrder;
    }

    public String pesanPengecekan() {
        String nomorOrder = generateNomorOrder("PN");
        nomorOrderPengecekan.add(nomorOrder);
        // ...
        return nomorOrder;
    }

    public boolean verifikasiPengecekanSebelumnya(String
nomorOrderPengecekanSebelumnya) {
        return
nomorOrderPengecekan.contains(nomorOrderPengecekanSebelumnya);
    }

    public String pesanPerbaikan(String nomorOrderPengecekanSebelumnya) {
        String nomorOrder = generateNomorOrder("PB");
        nomorOrderPerbaikan.add(nomorOrder);
        // ...
        return nomorOrder;
    }

```

```

    }

    private String generateNomorOrder(String prefix) {
        int count = 1;
        switch (prefix) {
            case "PO":
                count = nomorOrderPembersihan.size() + 1;
                break;
            case "PN":
                count = nomorOrderPengecekan.size() + 1;
                break;
            case "PB":
                count = nomorOrderPerbaikan.size() + 1;
                break;
        }
        return prefix + String.format("%03d", count);
    }

    private class User {
        private String username;
        private String password;

        public User(String username, String password) {
            this.username = username;
            this.password = password;
        }

        public String getUsername() {
            return username;
        }

        public String getPassword() {
            return password;
        }
    }
}

```


File : <JaskomView>.<java>

NOTE : Extension diisi dengan .java

```
import java.util.Scanner;

public class JaskomView {
    private Scanner scanner;

    public JaskomView() {
        scanner = new Scanner(System.in);
    }

    public int tampilkanMenuUtama() {
        System.out.println("===== MENU UTAMA =====");
        System.out.println("1. Login");
        System.out.println("2. Register");
        System.out.println("0. Keluar");
        System.out.print("Masukkan pilihan Anda: ");
        return scanner.nextInt();
    }

    public void tampilkanLogin() {
        System.out.println("\n===== LOGIN =====");
    }

    public String getInputUsername() {
        System.out.print("Masukkan username: ");
        return scanner.next();
    }

    public String getInputPassword() {
        System.out.print("Masukkan password: ");
        return scanner.next();
    }
}
```

```

}

public void tampilkanLoginBerhasil() {
    System.out.println("Login berhasil.");
}

public void tampilkanLoginGagal() {
    System.out.println("Login gagal.");
}

public String getRegisterChoice() {
    System.out.print("Apakah Anda ingin melakukan registrasi? (ya/tidak): ");
    return scanner.next();
}

public void tampilkanRegister() {
    System.out.println("\n===== REGISTRASI =====");
}

public String getNewUsername() {
    System.out.print("Masukkan username baru: ");
    return scanner.next();
}

public String getNewPassword() {
    System.out.print("Masukkan password baru: ");
    return scanner.next();
}

public void tampilkanUsernameSudahDigunakan() {
    System.out.println("Username sudah digunakan. Registrasi gagal.");
}

public void tampilkanRegistrasiBerhasil() {
    System.out.println("Registrasi berhasil.");
}

public void tampilkanLogoutBerhasil() {
    System.out.println("Logout berhasil.");
}

public void tampilkanMenuTidakValid() {
    System.out.println("Menu tidak valid. Silakan coba lagi.");
}

```

```

public int tampilkanDashboard() {
    System.out.println("\n===== DASHBOARD =====");
    System.out.println("1. Layanan Pembersihan");
    System.out.println("2. Layanan Pengecekan");
    System.out.println("3. Layanan Perbaikan");
    System.out.println("0. Logout");
    System.out.print("Masukkan pilihan Anda: ");
    return scanner.nextInt();
}

public void tampilkanPembersihan() {
    System.out.println("\n===== PEMBERSIHAN =====");
}

public String getThermalPasteChoice() {
    System.out.print("Apakah Anda membutuhkan penggantian thermal paste?
(ya/tidak): ");
    scanner.nextLine(); // Consume the newline character
    return scanner.nextLine();
}

public String getLokasi() {
    System.out.print("Masukkan lokasi Anda untuk mencari toko terdekat: ");
    return scanner.nextLine();
}

public void tampilkanNomorOrderPembersihan(String nomorOrder) {
    System.out.println("Pembersihan dengan penggantian thermal paste telah
dipesan dengan nomor order: " + nomorOrder);
}

public void tampilkanNomorOrderPembersihanTanpaThermalPaste(String
nomorOrder) {
    System.out.println("Pembersihan tanpa penggantian thermal paste telah
dipesan dengan nomor order: " + nomorOrder);
}

public void tampilkanPengecekan() {
    System.out.println("\n===== PENGECEKAN =====");
}

public void tampilkanNomorOrderPengecekan(String nomorOrder) {
    System.out.println("Pengecekan telah dipesan dengan nomor order: " +
nomorOrder);
}

```

```

public void tampilkanPerbaikan() {
    System.out.println("\n===== PERBAIKAN =====");
}

public String getLanjutkanPerbaikanChoice() {
    System.out.print("Apakah Anda ingin melanjutkan perbaikan? (ya/tidak: ");
    scanner.nextLine(); // Consume the newline character
    return scanner.nextLine();
}

public String getNomorOrderPengecekanSebelumnya() {
    System.out.print("Masukkan nomor order pengecekan sebelumnya: ");
    return scanner.next();
}

public void tampilkanNomorOrderPerbaikan(String nomorOrder) {
    System.out.println("Perbaikan telah dipesan dengan nomor order: " +
nomorOrder);
}

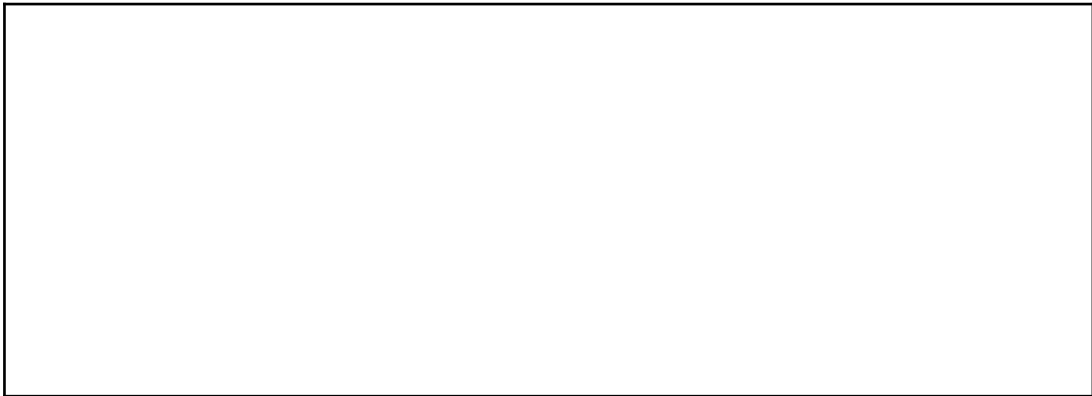
public void tampilkanPerbaikanDibatalkan() {
    System.out.println("Perbaikan dibatalkan.");
}

public void tampilkanPengecekanSebelumnyaTidakDitemukan() {
    System.out.println("Pengecekan sebelumnya tidak ditemukan.");
}

public void tampilkanVerifikasiPengecekanSebelumnyaGagal() {
    System.out.println("Verifikasi nomor order pengecekan sebelumnya gagal.");
}

public void tampilkanPilihanTidakValid() {
    System.out.println("Pilihan tidak valid. Silakan coba lagi.");
}
}

```



File : <Main>.<java>

NOTE : Extension diisi dengan .java

```
public class Main {  
    public static void main(String[] args) {  
        JaskomModel model = new JaskomModel();  
        JaskomView view = new JaskomView();  
        JaskomController controller = new JaskomController(model, view);  
        controller.jalankan();  
    }  
}
```

Latihan 9.2. Membuat *Test Case Scenario*

Project Name : _____Jaskom_____

Sub Module Name : cekUsernameTersedia

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
1	Input :Username Action : memberikan true atau false setelah mengecek username apakah username telah digunakan	memberikan true / false dengan input username oleh user apakah sudah ada atau belum	memberikan true / false dengan input username oleh user apakah sudah ada atau belum	Pass

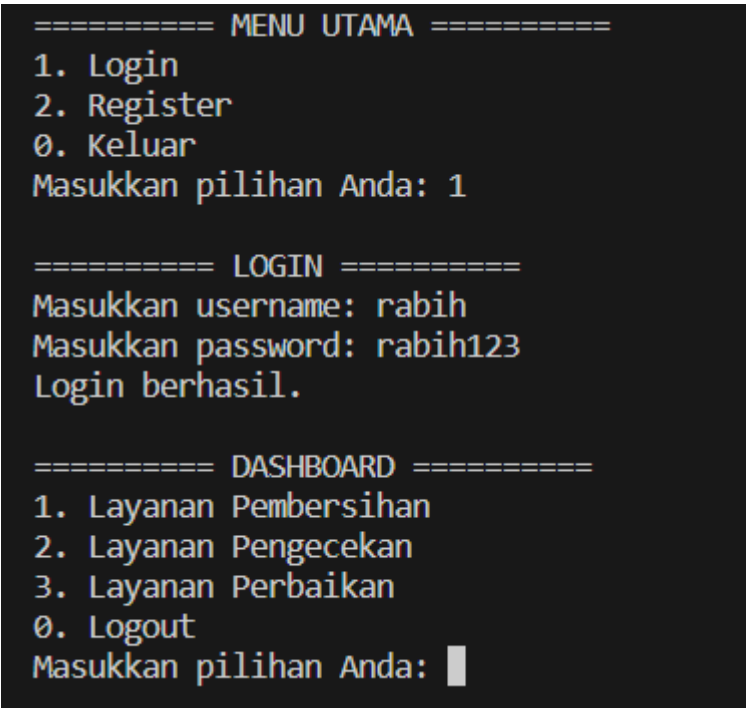
```
===== REGISTRASI =====  
Masukkan username baru: rabih  
Masukkan password baru: rabih123  
Registrasi berhasil.  
===== MENU UTAMA =====  
1. Login  
2. Register  
0. Keluar  
Masukkan pilihan Anda: 2  
  
===== REGISTRASI =====  
Masukkan username baru: rabih  
Masukkan password baru: rabih123  
Username sudah digunakan. Registrasi gagal.
```

Screenshot :

Sub Module Name : prosesLogin

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
2	Input :username , password Action :jika username dan password benar masuk memberikan true	jika benar masuk ke dashboard jika salah meminta registrasi	jika benar masuk ke dashboard jika salah meminta registrasi	Pass
<div> <div>Screenshot :</div>  <pre> ===== MENU UTAMA ===== 1. Login 2. Register 0. Keluar Masukkan pilihan Anda: 1 ===== LOGIN ===== Masukkan username: rabih Masukkan password: rabih123 Login berhasil. ===== DASHBOARD ===== 1. Layanan Pembersihan 2. Layanan Pengecekan 3. Layanan Perbaikan 0. Logout Masukkan pilihan Anda: █ </pre> </div>				

Sub Module Name : Registrasi

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
3	Input : Username, Password Action : Data username dan password masuk	Username dan password terdata didatabase	Username dan password terdata didatabase	Pass / Fail

```
===== MENU UTAMA =====  
1. Login  
2. Register  
0. Keluar  
Masukkan pilihan Anda: 2  
  
===== REGISTRASI =====  
Masukkan username baru: Shemi  
Masukkan password baru: amogus123  
Registrasi berhasil.
```

Screenshot :

Sub Module Name : prosesPilihanPembersihan

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
4	Input : location Action : membuat nomor order setelah mengecek lokasi	mengeluarkan nomor order setelah memasukkan lokasi	memberi opsi memakai thermal paste atau tidak, mengeluarkan nomor order setelah memasukkan lokasi	Pass / Fail

Screenshot :

```
===== DASHBOARD =====  
1. Layanan Pembersihan  
2. Layanan Pengecekan  
3. Layanan Perbaikan  
0. Logout  
Masukkan pilihan Anda: 1  
  
===== PEMBERSIHAN =====  
Apakah Anda membutuhkan penggantian thermal paste? (ya/tidak): ya  
Masukkan lokasi Anda untuk mencari toko terdekat: Bekasi  
Pembersihan dengan penggantian thermal paste telah dipesan dengan nomor order: P0001
```

Sub Module Name : prosesPilihanPengecekan

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
5	Input : location Action :membuat nomor order setelah mengecek lokasi	mengeluarkan nomor order setelah memasukkan lokasi	mengeluarkan nomor order setelah memasukkan lokasi	Pass
<div>Screenshot :</div> <div><pre>===== MENU UTAMA ===== 1. Login 2. Register 1. Layanan Pembersihan 2. Layanan Pengecekan 3. Layanan Perbaikan 0. Logout Masukkan pilihan Anda: 2 ===== PENGECEKAN ===== Pengecekan telah dipesan dengan nomor order: PN001</pre></div>				

Sub Module Name : prosesPilihanPerbaikan

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
6	Input:continueRepairChoice, previousOrderNumber Action :membuat nomor order setelah masuk previousOrderNumber dancontinueRepairChoice	membuat nomor order perbaikan	membuat nomor order perbaikan	Pass

```
===== DASHBOARD =====
1. Layanan Pembersihan
2. Layanan Pengecekan
3. Layanan Perbaikan
0. Logout
Masukkan pilihan Anda: 3

===== PERBAIKAN =====
Apakah Anda ingin melanjutkan perbaikan? (ya/tidak): ya
Masukkan nomor order pengecekan sebelumnya: PN001
Perbaikan telah dipesan dengan nomor order: PB001
```

Screenshot :

Sub Module Name : tampilkanPesanPembersihanDipesan/tampilkanPesanPengecekanDipesan/tampilkanPesanPerbaikanDipesan

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
7	Input : - Action : menampilkan nomor pesan	mengeluarkan nomor pesanan sesuai dengan opsi	mengeluarkan nomor pesanan sesuai dengan opsi	Pass

===== PEMBERSIHAN =====

Apakah Anda membutuhkan penggantian thermal paste? (ya/tidak): ya

Masukkan lokasi Anda untuk mencari toko terdekat: Bekasi

Pembersihan dengan penggantian thermal paste telah dipesan dengan nomor order: P0001

===== DASHBOARD =====

1. Layanan Pembersihan

2. Layanan Pengecekan

3. Layanan Perbaikan

Masukkan lokasi Anda untuk mencari toko terdekat: jakarta

Pembersihan tanpa penggantian thermal paste telah dipesan dengan nomor order: P0002

===== DASHBOARD =====

1. Layanan Pembersihan

2. Layanan Pengecekan

3. Layanan Perbaikan

0. Logout

Masukkan pilihan Anda: 2

===== PENGECEKAN =====

Pengecekan telah dipesan dengan nomor order: PN001

===== DASHBOARD =====

1. Layanan Pembersihan

2. Layanan Pengecekan

3. Layanan Perbaikan

0. Logout

Masukkan pilihan Anda: 3

===== PERBAIKAN =====

Apakah Anda ingin melanjutkan perbaikan? (ya/tidak): ya

Masukkan nomor order pengecekan sebelumnya: PN001

Perbaikan telah dipesan dengan nomor order: PB001

Sub Module Name : tampilkanPesanLogout

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
8	Input : - Action :mengeluarkan pesan logout	menampilkan pesan logout	menampilkan pesan logout	Pass

Screenshot :

```
===== DASHBOARD =====  
1. Layanan Pembersihan  
2. Layanan Pengecekan  
3. Layanan Perbaikan  
0. Logout  
Masukkan pilihan Anda: 0  
Logout berhasil.
```