

# **LAPORAN PRAKTIKUM**

## **METODOLOGI PENGEMBANGAN**

### **PERANGKAT LUNAK**

#### **MODUL 9**



**Versi 3.1**

**Penyusun : Tim Dosen MPPL**

|             |          |                                |
|-------------|----------|--------------------------------|
| <b>Nama</b> | <b>:</b> | <b>Sonya Ridesia Hastari</b>   |
|             |          | <b>Chaesa Namida Arumdapta</b> |
|             |          | <b>Evanda Manggani</b>         |
|             |          | <b>Tommy Andrian</b>           |
| <b>NIM</b>  | <b>:</b> | <b>064002200007</b>            |
|             |          | <b>064002200008</b>            |
|             |          | <b>064002200013</b>            |
|             |          | <b>064002200029</b>            |

**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 ( <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

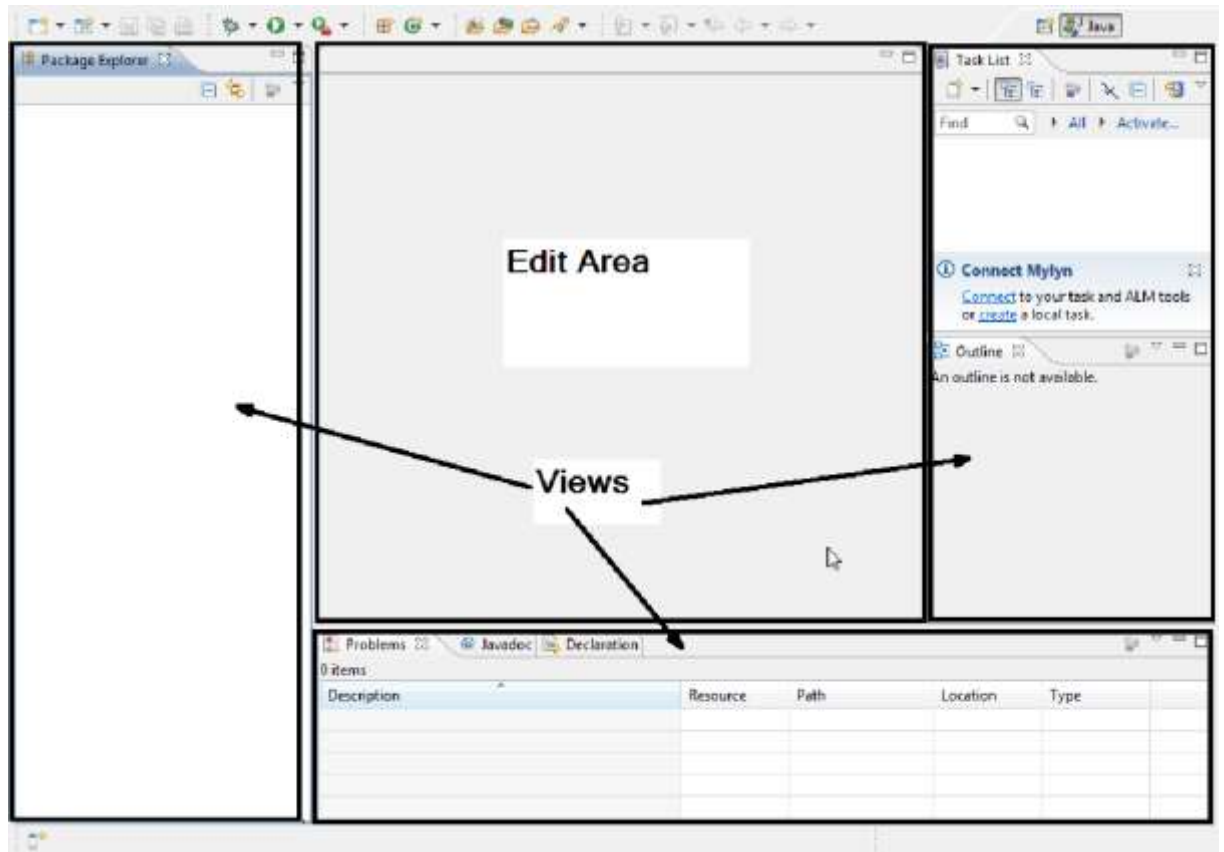
Eclipse merupakan sebuah *development environment* yang sangat *user-friendly* untuk membuat program JAVA. Eclipse sudah banyak dikenal dengan baik oleh banyak programmer JAVA. dan digunakan untuk berbagai pengembangan aplikasi. Sehingga mempelajari Eclipse dapat menambah wawasan dan pengetahuan tentang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

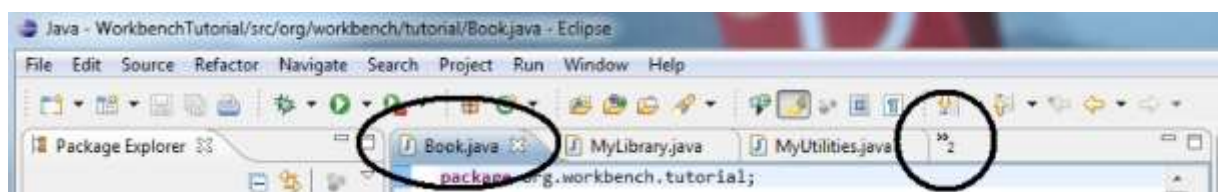
1. Pada komputer berbasis Windows, *double-click* pada *file* eclipse.exe. Sedangkan pada komputer berbasis Linux atau Mac, *double-click* pada *file* eclipse.
2. Ketika muncul *dialog box*, masukkan *pathname* untuk *workspace* (digunakan untuk menyimpan *JAVA Project*), dan tekan tombol **OK**.
3.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

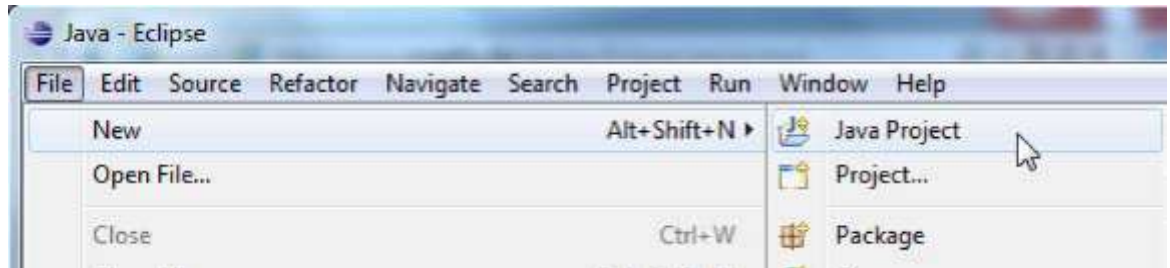
**Edit Area** dapat memiliki *multiple windows* pada *space* yang tersedia.



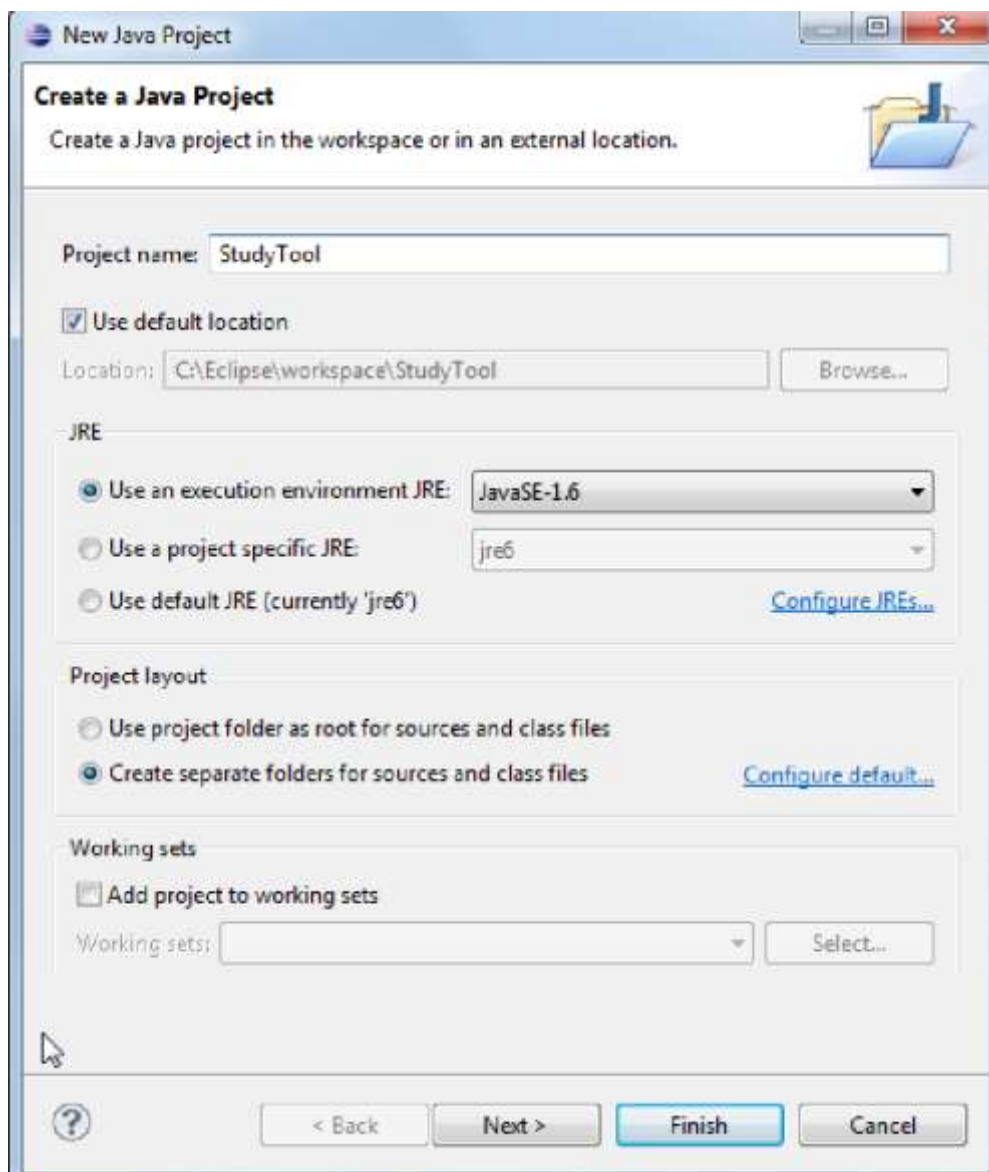
Langkah untuk membuat program JAVA :

1. **Buat Project**

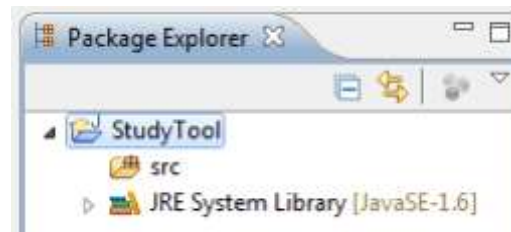
- Pilih **File** → **New** → **Java Project**



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

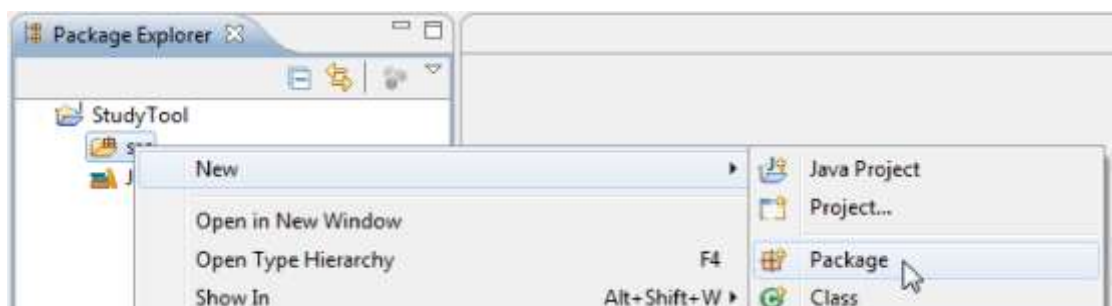


- Seluruh informasi tentang *project* di-set ke **Default** dan klik **Finish**
- Setelah itu, *project* sudah selesai dibuat dan ditampilkan sebagai folder
- **Package view** dapat dilihat di sebelah kiri dari **Edit Area**

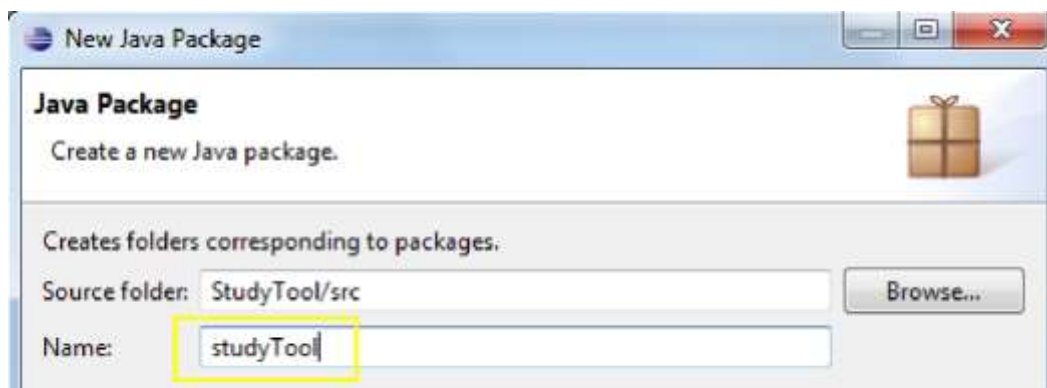


## 2. Buat **Package**

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

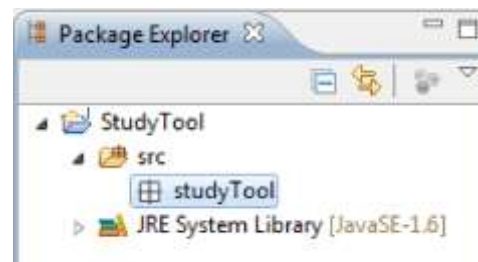


- Ketikkan nama *package* dan klik **Finish**



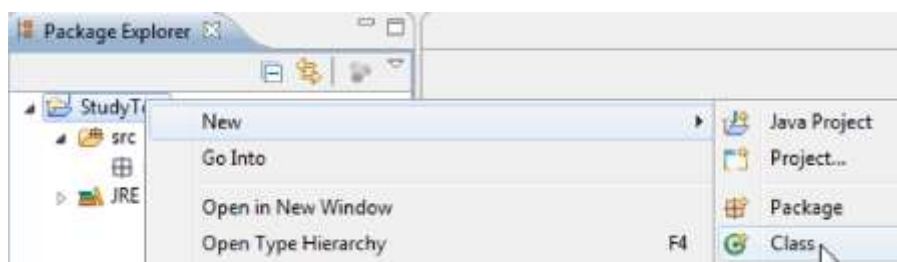


- Sehingga *package* tersebut akan tampil pada **Package Explorer**

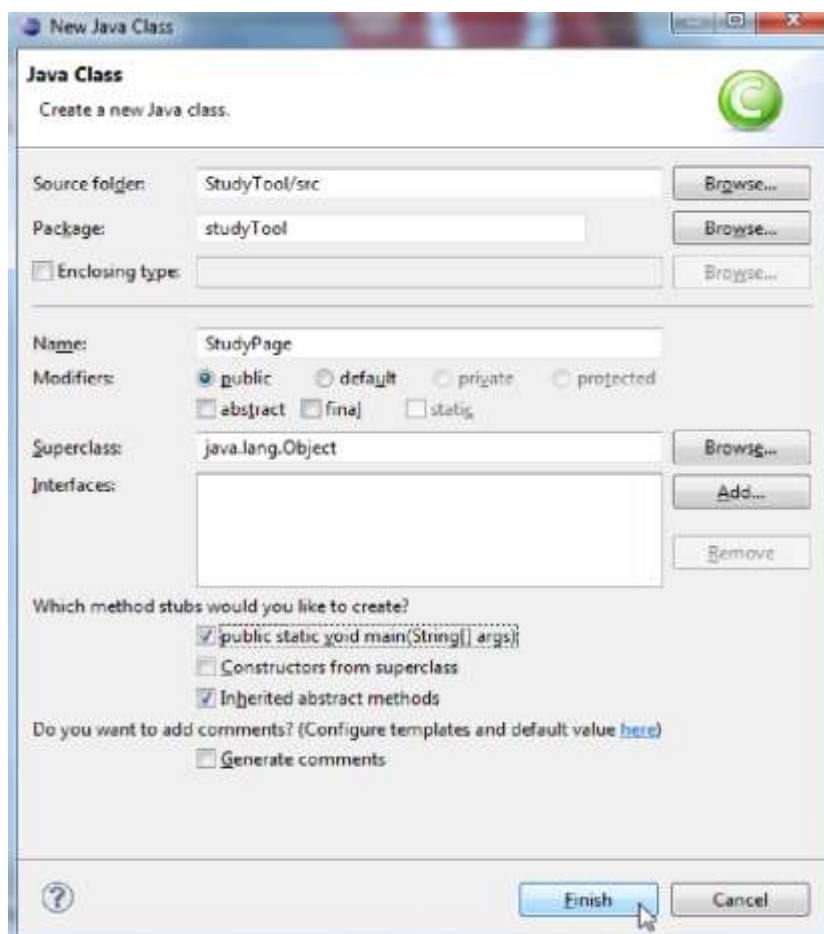


### 3. Pilih **Class**

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



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

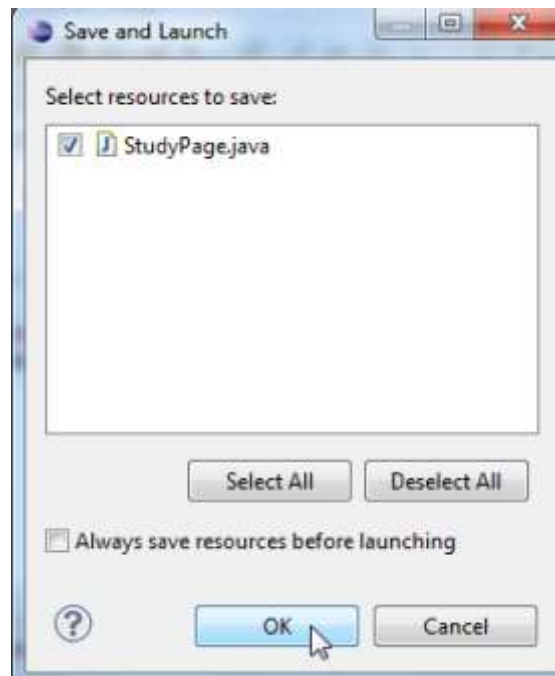


4. *Create and run JAVA code*

- 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 : < RegisterApp >.<java>

**NOTE** : Extension diisi dengan .java

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

public class RegisterApp {
    private Map<String, String> userCredentials;

    public RegisterApp() {
        userCredentials = new HashMap<>();
    }

    public void registerUser(String username, String password) {
        if (username.isEmpty()) {
            System.out.println("Username tidak boleh kosong");
            return;
        }

        if (!username.matches("[a-zA-Z]+")) {
            System.out.println("Username hanya boleh mengandung huruf");
            return;
        }

        if (password.length() < 8 || !password.matches(".*\\d.*")) {
            System.out.println("Password harus terdiri dari 8 karakter dan minimal
            satu angka");
            return;
        }

        if (!userCredentials.containsKey(username)) {
            userCredentials.put(username, password);
            System.out.println("Registrasi berhasil!");
        } else {
            System.out.println("Username sudah terdaftar. Silakan coba dengan
            username lain.");
        }
    }

    public boolean loginUser(String username, String password) {
        if (userCredentials.containsKey(username) &&
            userCredentials.get(username).equals(password)) {
            System.out.println("Login berhasil!");
        }
    }
}
```

```

        return true;
    } else {
        System.out.println("Login gagal. Silakan cek kembali username dan
password.");
        return false;
    }
}

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

    while (true) {
        System.out.println("Silakan pilih aksi:");
        System.out.println("1. Register");
        System.out.println("2. Login");
        System.out.println("3. Keluar");
        System.out.print("Pilihan Anda: ");
        int choice = scanner.nextInt();
        scanner.nextLine();

        switch (choice) {
            case 1:
                System.out.print("Username: ");
                String username = scanner.nextLine();
                System.out.print("Password: ");
                String password = scanner.nextLine();
                registerApp.registerUser(username, password);
                break;
            case 2:
                System.out.print("Username: ");
                username = scanner.nextLine();
                System.out.print("Password: ");
                password = scanner.nextLine();
                registerApp.loginUser(username, password);
                break;
            case 3:
                System.out.println("Terima kasih!");
                System.exit(0);
                break;
            default:
                System.out.println("Pilihan tidak valid. Silakan pilih angka 1-3.");
        }
    }
}

```

```
}
```

File : < Schedule >.<java>

**NOTE** : Extension diisi dengan .java

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.*;

class Schedule {
    private Map<String, String> schedule;

    public Schedule() {
        schedule = new HashMap<>();
    }

    public void addSchedule(String date, String event) {
        if (schedule.containsKey(date)) {
            String existingEvents = schedule.get(date);
            existingEvents += ", " + event;
            schedule.put(date, existingEvents);
        } else {
            schedule.put(date, event);
        }
    }

    public void editSchedule(String date, String event, String newEvent) {
        if (schedule.containsKey(date)) {
            String existingEvents = schedule.get(date);
            if (existingEvents.contains(event)) {
                existingEvents = existingEvents.replace(event, newEvent);
            }
        }
    }
}
```

```

        schedule.put(date, existingEvents);
        System.out.println("Jadwal berhasil diubah.");
    } else {
        System.out.println("Jadwal tidak ditemukan.");
    }
} else {
    System.out.println("Tanggal tidak ditemukan.");
}
}

public void deleteSchedule(String date, String event) {
    if (schedule.containsKey(date)) {
        String existingEvents = schedule.get(date);
        if (existingEvents.contains(event)) {
            existingEvents = existingEvents.replace(event + " ", "");
            existingEvents = existingEvents.replace(" " + event, "");
            existingEvents = existingEvents.replace(event, "");
            schedule.put(date, existingEvents);
            System.out.println("Jadwal berhasil dihapus.");
        } else {
            System.out.println("Jadwal tidak ditemukan.");
        }
    } else {
        System.out.println("Tanggal tidak ditemukan.");
    }
}

public void displaySchedule() {
    if (!schedule.isEmpty()) {
        for (Map.Entry<String, String> entry : schedule.entrySet()) {
            String date = entry.getKey();
            String events = entry.getValue();
            System.out.println(date + ": " + events);
        }
    } else {
        System.out.println("Jadwal kosong.");
    }
}
}

class Reminder {
    private Map<String, Date> reminders;

    public Reminder() {
        reminders = new HashMap<>();
    }
}

```

```

    }

    public void addReminder(String event, Date date) {
        reminders.put(event, date);
    }

    public void displayReminders() {
        if (!reminders.isEmpty()) {
            for (Map.Entry<String, Date> entry : reminders.entrySet()) {
                String event = entry.getKey();
                Date date = entry.getValue();
                System.out.println("Pengingat: Acara " + event + " pada tanggal " +
date + " akan dimulai.");
            }
        } else {
            System.out.println("Tidak ada pengingat.");
        }
    }
}

public class ScheduleReminder {
    private Schedule schedule;
    private Reminder reminder;
    private Scanner scanner;

    public ScheduleReminder() {
        schedule = new Schedule();
        reminder = new Reminder();
        scanner = new Scanner(System.in);
    }

    public void addSchedule() {
        System.out.print("Masukkan tanggal (YYYY-MM-DD): ");
        String date = scanner.nextLine();
        System.out.print("Masukkan acara: ");
        String event = scanner.nextLine();
        schedule.addSchedule(date, event);
        System.out.println("Jadwal berhasil ditambahkan.");
    }

    public void editSchedule() {
        System.out.print("Masukkan tanggal (YYYY-MM-DD): ");
        String date = scanner.nextLine();
        System.out.print("Masukkan acara yang ingin diubah: ");
        String oldEvent = scanner.nextLine();
    }
}

```

```

        System.out.print("Masukkan acara baru: ");
        String newEvent = scanner.nextLine();
        schedule.editSchedule(date, oldEvent, newEvent);
    }

    public void deleteSchedule() {
        System.out.print("Masukkan tanggal (YYYY-MM-DD): ");
        String date = scanner.nextLine();
        System.out.print("Masukkan acara yang ingin dihapus: ");
        String event = scanner.nextLine();
        schedule.deleteSchedule(date, event);
    }

    public void displaySchedule() {
        schedule.displaySchedule();
    }

    public void setReminder() {
        System.out.print("Masukkan tanggal (YYYY-MM-DD): ");
        String dateStr = scanner.nextLine();
        System.out.print("Masukkan acara: ");
        String event = scanner.nextLine();
        System.out.print("Masukkan jumlah jam sebelum acara dimulai: ");
        int hoursBefore = scanner.nextInt();
        scanner.nextLine(); // consume newline

        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd
HH:mm");
        try {
            Date date = dateFormat.parse(dateStr + " 00:00");
            Calendar calendar = Calendar.getInstance();
            calendar.setTime(date);
            calendar.add(Calendar.HOUR_OF_DAY, -hoursBefore);
            Date reminderDate = calendar.getTime();

            reminder.addReminder(event, reminderDate);
            System.out.println("Peningat berhasil ditambahkan.");
        } catch (ParseException e) {
            System.out.println("Format tanggal tidak valid.");
        }
    }

    public void displayReminders() {
        reminder.displayReminders();
    }
}

```



```

public void run() {
    while (true) {
        System.out.println("Menu:");
        System.out.println("1. Tambah Jadwal");
        System.out.println("2. Edit Jadwal");
        System.out.println("3. Hapus Jadwal");
        System.out.println("4. Tampilkan Jadwal");
        System.out.println("5. Set Reminder");
        System.out.println("6. Tampilkan Pengingat");
        System.out.println("0. Keluar");
        System.out.print("Pilih menu (0-6): ");
        int choice = scanner.nextInt();
        scanner.nextLine(); // consume newline

        switch (choice) {
            case 1:
                addSchedule();
                break;
            case 2:
                editSchedule();
                break;
            case 3:
                deleteSchedule();
                break;
            case 4:
                displaySchedule();
                break;
            case 5:
                setReminder();
                break;
            case 6:
                displayReminders();
                break;
            case 0:
                System.out.println("Terima kasih!");
                System.exit(0);
            default:
                System.out.println("Pilihan tidak valid.");
        }
        System.out.println();
    }
}

public static void main(String[] args) {

```

```
ScheduleReminder scheduleReminder = new ScheduleReminder();  
scheduleReminder.run();  
}  
}
```

### Latihan 9.2. Membuat *Test Case Scenario*

Project Name : Tima\_\_\_\_\_

|                                 |
|---------------------------------|
| Sub Module Name : <RegisterApp> |
|---------------------------------|

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 |
|--------------|---|---|--|--------------|
|              | Input : Masukkan username dan paasword<br><br>Action : system mengecek apakah username dan password benar | Saat username dan password benar program akan memberikan bahwa username dan password yang benar | Saat username dan password benar program akan memberikan bahwa username dan password benar | Pass         |

```
Silakan pilih aksi:  
1. Register  
2. Login  
3. Keluar  
Pilihan Anda: 1  
Username: tommy a  
Password: tommy293  
Registrasi berhasil!  
Silakan pilih aksi:  
1. Register  
2. Login  
3. Keluar  
Pilihan Anda: 2  
Username: tommy a  
Password: tommy293  
Login berhasil!  
Silakan pilih aksi:  
1. Register  
2. Login  
3. Keluar  
Pilihan Anda: 3  
Terima kasih!  
PS C:\Users\evanda>
```

Screenshot :

**Sub Module Name :** <Penjadwalan.java>

**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 |
|--------------|--|---|--|--------------|
|              | Input : memilih angka 1, untuk menambahkan jadwal baru. Memilih angka 2 untuk mengedit jadwal. Memilih angka 4 untuk menampilkan tampilan jadwal. Memilih angka 5 untuk mengatur pengingat. Memilih angka 6 untuk menampilkan pengingat. Memilih angka 3 untuk menghapus jadwal bila sudah selesai. Kemudian yang terakhir memilih angka 0 untuk keluar. | <ul style="list-style-type: none"><li>- Saat user berhasil menambah jadwal, maka system akan menampilkan jadwal berhasil ditambahkan.</li><li>- Saat user berhasil mengedit jadwal, maka system akan menampilkan jadwal berhasil di edit.</li></ul> | Saat semua fitur berjalan dengan benar. Maka system akan menjalankannya. | Pass         |

|  |   |  |  |  |
|--|---|--|--|--|
|  | Action : system mengecek. Kemudian menjalankan sesuai angka yang user masukan | <ul style="list-style-type: none"> <li>- Saat user berhasil menampilkan jadwal, maka system akan menampilkan jadwal berhasil di tampilkan.</li> <li>- Saat user berhasil mengatur pengingat jadwal, maka system akan menampilkan mengatur pengingat jadwal berhasil ditambahkan.</li> <li>- Saat user berhasil menampilkan pengingat jadwal, maka system akan menampilkan jadwal berhasil ditampilkan.</li> <li>- Saat user berhasil menghapus jadwal, maka system akan menampilkan</li> </ul> |  |  |
|--|---|--|--|--|

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | jadwal berhasil<br>dihapus.<br>- Saat user menekan<br>0 maka system<br>akan menampilkan<br>Terima Kasih |  |  |
|--|--|---|--|--|

```

Menu:
1. Tambah Jadwal
2. Edit Jadwal
3. Hapus Jadwal
4. Tampilkan Jadwal
5. Set Reminder
6. Tampilkan Pengingat
0. Keluar
Pilih menu (0-6): 1
Masukkan tanggal (YYYY-MM-DD): 2023-06-06
Masukkan acara: tommy kencan dengan putry
Jadwal berhasil ditambahkan.

Menu:
1. Tambah Jadwal
2. Edit Jadwal
3. Hapus Jadwal
4. Tampilkan Jadwal
5. Set Reminder
6. Tampilkan Pengingat
0. Keluar
Pilih menu (0-6): 2
Masukkan tanggal (YYYY-MM-DD): 2023-06-06
Masukkan acara yang ingin diubah: tommy kencan dengan putry
Masukkan acara baru: tommy ngelamar putry
Jadwal berhasil diubah.

Menu:
1. Tambah Jadwal
2. Edit Jadwal
3. Hapus Jadwal
4. Tampilkan Jadwal
5. Set Reminder
6. Tampilkan Pengingat
0. Keluar
Pilih menu (0-6): 4
2023-06-06: tommy ngelamar putry

Menu:
1. Tambah Jadwal
2. Edit Jadwal
3. Hapus Jadwal
4. Tampilkan Jadwal
5. Set Reminder
6. Tampilkan Pengingat

```

Screenshot :



Menu:

1. Tambah Jadwal
2. Edit Jadwal
3. Hapus Jadwal
4. Tampilkan Jadwal
5. Set Reminder
6. Tampilkan Pengingat
0. Keluar

Pilih menu (0-6): 5

Masukkan tanggal (YYYY-MM-DD): 2023-06-06

Masukkan acara: tonmy ngelamar putry

Masukkan jumlah jam sebelum acara dimulai: 10

Pengingat berhasil ditambahkan.

Menu:

1. Tambah Jadwal
2. Edit Jadwal
3. Hapus Jadwal
4. Tampilkan Jadwal
5. Set Reminder
6. Tampilkan Pengingat
0. Keluar

Pilih menu (0-6): 6

Pengingat: Acara 'tonmy ngelamar putry' pada tanggal Mon Jun 05 14:00:00 WIB 2023 akan dimulai.

Menu:

1. Tambah Jadwal
2. Edit Jadwal
3. Hapus Jadwal
4. Tampilkan Jadwal
5. Set Reminder
6. Tampilkan Pengingat
0. Keluar

Pilih menu (0-6): 3

Masukkan tanggal (YYYY-MM-DD): 2023-06-06

Masukkan acara yang ingin dihapus: tonmy ngelamar putry

Jadwal berhasil dihapus.

Menu:

1. Tambah Jadwal
2. Edit Jadwal
3. Hapus Jadwal
4. Tampilkan Jadwal
5. Set Reminder
6. Tampilkan Pengingat

Menu:

1. Tambah Jadwal
  2. Edit Jadwal
  3. Hapus Jadwal
  4. Tampilkan Jadwal
  5. Set Reminder
  6. Tampilkan Pengingat
  0. Keluar
- Pilih menu (0-6): 0
- Terima kasih!