

Praktikum Pemrograman Berbasis Objek

Pertemuan 10

Review Time



1. Collections Framework
2. Set, List, Map
3. Iterator pada Collections

Materi Pertemuan 10

Yang akan dipelajari hari ini

Materi Pertemuan 10

01

JDBC

Aplikasi Java dengan Database

02

Proyek Akhir

Ketentuan Proyek Akhir Prak PBO

01

JDBC

JDBC

Java Database Connector adalah sebuah penghubung antara aplikasi Java dengan sebuah database.

Instalasi JDBC

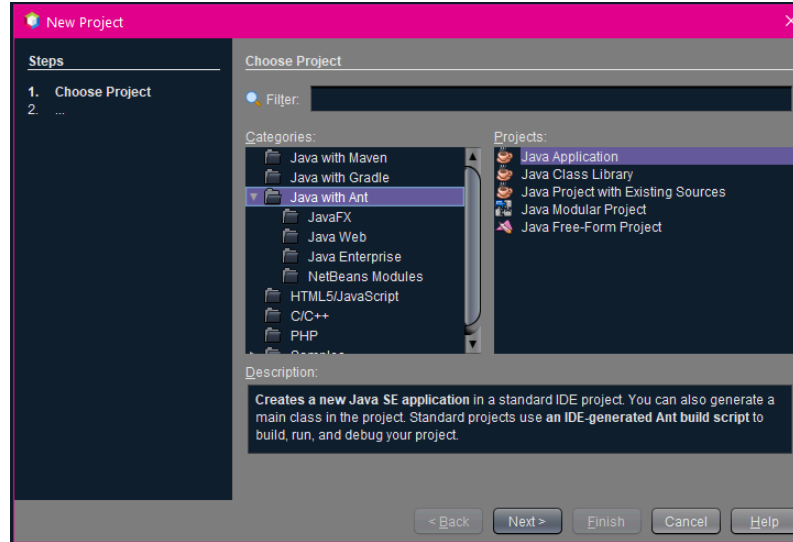
<https://dev.mysql.com/downloads/connector/j/>

Gunakan Versi 8.2.0 ke atas. Install saja yg terbaru 9.1.0

Install ZIPnya dan extract JAR file yang ada dalam ZIP

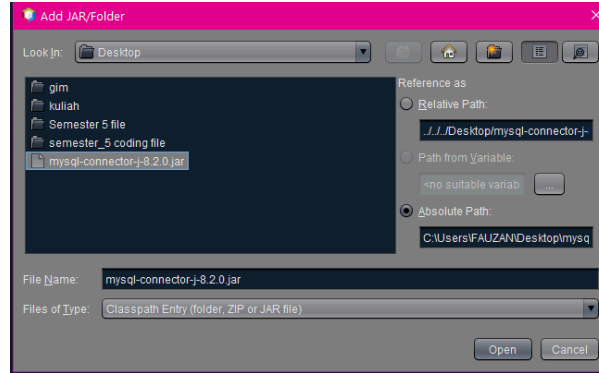
Buat Proyek Baru dalam Netbeans

1. Buat project baru dalam Netbeans
2. Pilih `Java with Ant`
dan pilih opsi Java application



Menambahkan Library dalam Project

Klik kanan folder `Libraries` dan pilih opsi `Add JAR/Folder...`. Lalu, cari JAR file yang telah dikeluarkan, pilih, dan tekan `Open`.



Saran

Disarankan untuk membuat folder khusus untuk file JAR di dalam proyek. Dengan cara ini, ketika ingin menambahkan library, cukup gunakan opsi `Add JAR/Folder ...` dan pilih file JAR yang ada di folder tersebut. Ketika proyek dikirimkan kepada orang lain, tidak perlu melakukan inisiasi ulang terhadap library yang digunakan.



Menggunakan JDBC

Untuk membuat hubungan antara aplikasi dan database, implementasikan kode berikut : Tips : Agar lebih modular, sebaiknya snippet kode ini dijadikan kelasnya sendiri.

```
import java.sql.*;
public class Test{
    public static void main(String[] args) {
        String driver = "com.mysql.cj.jdbc.Driver";
        String db = "jdbc:mysql://localhost/NAMA_DATABASE";
        String user = "root"; // Bila username berbeda, ganti
        String password = ""; // Bila instance MySQL memiliki password
        Connection conn = null;
        Statement state = null;

        try{
            Class.forName(driver);
        } catch (Exception e){
            System.out.println("Driver Error");
        }
    }
}
```

```
try{
    conn = (Connection) DriverManager.getConnection(db, user, password);
    state = (Statement) conn.createStatement();
} catch (Exception e){
    System.out.println("Connection Error");
}

System.out.println("Database Connected");
try{
    state.executeUpdate("MYSQL QUERY");
} catch (Exception e){
    System.out.println("Error");
}
}
```

Buat objek Konektor

Contoh implementasi objek konektor :

```
import java.sql.*;
public class Konektor{
    private String driver = "com.mysql.cj.jdbc.Driver";
    private String db = "jdbc:mysql://localhost/NAMA_DATABASE";
    // Bila ada yang instance mysqlnya pindah port, tuliskan portnya
    private String user = "root"; // Bila username berbeda
    private String password = ""; // Bila instance MySQL n
    private Connection conn = null;
    private Statement state = null;
    private ResultSet rs = null;

    ...
}
```

```
// Constructor
public void Konektor() {

    try{
        Class.forName(driver);
    } catch(Exception e){
        System.out.println("Driver Error");
    }

    try{
        conn = (Connection)DriverManager.getConnection(
            db, user, password);
        state = (Statement) conn.createStatement();
    } catch(Exception e){
        System.out.println("Connection Error");
    }

    System.out.println("Database Connected");
}
```

Buat objek Konektor

Contoh implementasi objek konektor :









```
// Method untuk Create, Update, atau Delete
public void query(string stringQuery) {
    try{
        state.executeUpdate(stringQuery);
    } catch(Exception e){
        System.out.println("Error");
    }
}
```

```
// Method untuk Read
public ResultSet getData(String SQLString){
    try {
        state = (Statement) con.createStatement();
        rs = state.executeQuery(SQLString);
    } catch(Exception e){
        System.out.println("Query Error");
    }
    return rs;
}
```

Contoh CRUD

Berikut adalah contoh implementasi CRUD menggunakan JDBC :

```
public class Test {  
    public static void main(String args[]) {  
        Konektor conn = new Konektor();  
        conn.query("INSERT INTO `game`  
            (`id`, `name`, `genre`, `description`) VALUES  
            ('5', 'Huniepop', 'Puzzle', 'Candy crush + Anime');"  
    }  
}
```

	id	name	genre	description
<input type="checkbox"/>  Edit  Copy  Delete	5	Huniepop	Puzzle	Candy crush + Anime
 <input type="checkbox"/> Check all With selected:  Edit  Copy  Delete  Export				

Contoh CRUD

Berikut adalah contoh implementasi CRUD menggunakan JDBC :

```
import java.sql.ResultSet;
public class Test {
    public static void main(String args[]) {
        Konektor conn = new Konektor();
        ResultSet select = conn.getData("SELECT * FROM `game`");

        try {
            while (select.next()) {
                System.out.println("ID : " + select.getInt(1));
                System.out.println("Name : " + select.getString(2));
                System.out.println("Genre : " + select.getString(3));
                System.out.println("Desc : " + select.getString(4));
                System.out.println();
            }
        } catch (Exception e) {
            System.out.println("Err");
        }
    }
}
```

```
ID : 5
Name : Huniepop
Genre : Puzzle
Desc : Candy crush + Anime

BUILD SUCCESSFUL (total time: 0 seconds)
|
```

03

Proyek Akhir

Scope Proyek Akhir

1. Buat aplikasi dengan tema bebas, misal Perpustakaan, Akademik, Toko, Dll (Kecuali tema yang sudah dilakukan di dalam matkul Pak Akmal)
2. Memakai Netbeans untuk melakukan project.
3. Mengimplementasikan GUI dalam aplikasi.
4. Data dalam aplikasi dapat tersimpan dalam sebuah database.
5. Database memiliki 3 tabel atau lebih.
6. Mengimplementasikan Autentikasi.
7. Aplikasi mengimplementasikan CRUD.
8. Gunakan modularitas pada struktur kode, misalkan class Mahasiswa dan Dosen dipisahkan.
9. Interface yang menarik menjadi nilai tambahan
10. Anggota maksimum 3 orang dalam 1 kelompok. 2 diperbolehkan

03 - A

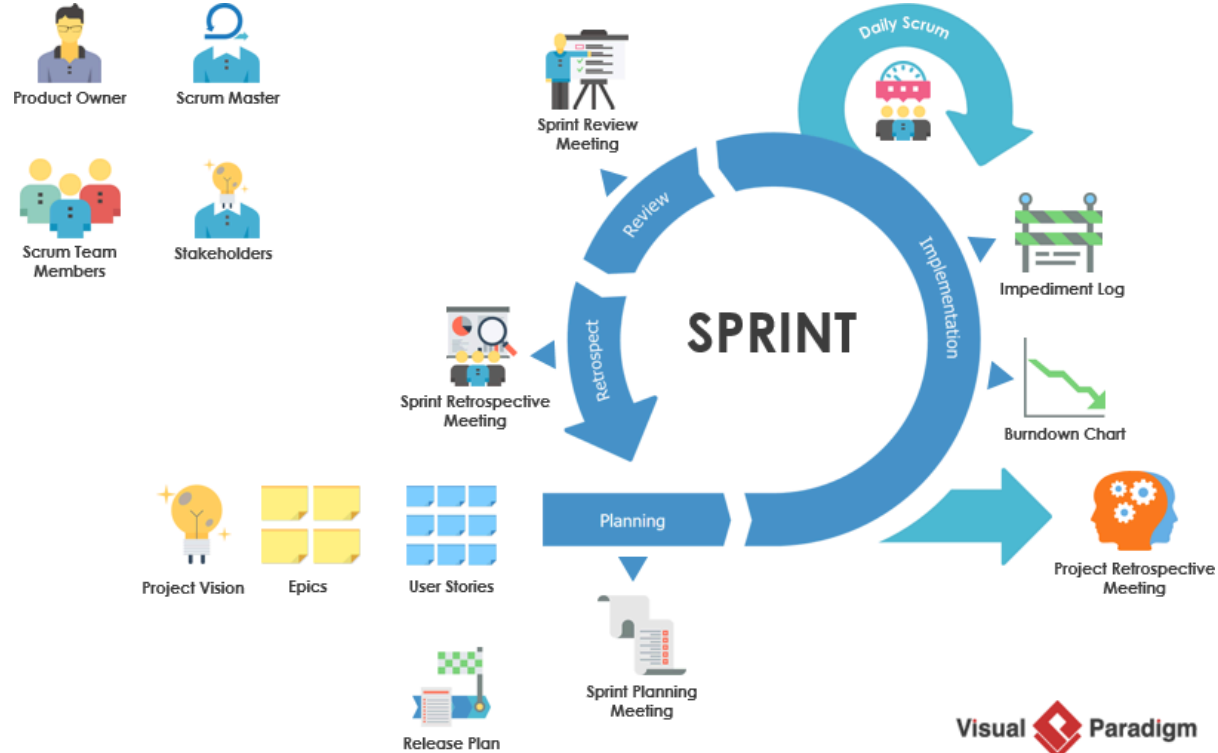
Scrum Framework

Scrum Framework

Scrum adalah sebuah framework untuk proses development software.

- Fast paced development
- Simple to understand
- Difficult to master

Scrum Framework



Scrum Framework

1. Product Backlog
2. Sprint Planning
3. Sprint Backlog
4. Daily Scrum Meeting
5. Sprint Review + Sprint Retrospective



Product Backlog

Product backlog adalah rencana keseluruhan produk secara general.

Sprint Planning

Biasanya rentang 1-2 minggu. Praktikan akan mengimplementasikan Sprint rentang 1 minggu.

Diskusi memecah product backlog menjadi lebih spesifik

Menghasilkan “**Sprint Backlog**”

Sprint Backlog

Setiap anggota tim secara mandiri mengambil backlog yang akan dikerjakan dalam 1 sprint kedepan.

Tempat manajemen sprint:

- Trello (Gratis)
- GitHub Project (Gratis)
- Atlassian (Berbayar)

Scrum Meeting

Original : 15 menit setiap hari

Praktikum : durasi bebas, 1x seminggu

3 Poin Bahasan:

1. Apa yang telah dikerjakan selama 1 minggu kebelakang
2. Kendala dalam pengerjaan
3. Apa yang akan dikerjakan selama 1 minggu kedepan

Setiap anggota tim membahas satu persatu. Fungsinya sebagai forum koordinasi tim.

Scrum Review + Retrospective

Diadakan di akhir Sprint.

Membahas apa kendala dan apa yang bisa ditingkatkan di sprint berikutnya.

Rencana pengambilan Sprint Backlog untuk Sprint selanjutnya.

Iterasi kembali ke sprint planning...

Snippets

Snippets yang sangat penting untuk belajar GUI Netbeans (Berisi Modul dari dosen serta program GUI yang siap dicoba)

snippets-PBO-10

Jangan dipakai buat ngumpulin tugas, Buat referensi boleh

Silahkan untuk kalian mencoba menjalankan dan mempelajari snippets yang disediakan pada repository tersebut. Selamat belajar!

03 - B

Teknis Proyek Akhir

Teknis Proyek

Link Daftar Anggota Kelompok:

 Daftar Anggota Kelompok

Harap mengisi data di sheet ini paling lambat Minggu, 2 November 2024, karena akan segera diserahkan kepada Dosen.

Link Assignment A:

https://classroom.github.com/a/Fum_TD-D

Assignment B:

<https://classroom.github.com/a/7mX5QgGR>

Untuk nama kelompok, gunakan format “**Kelompok KELAS-XX**”, misal “Kelompok B-02”

- ``changelog`` berisi dokumen mingguan.
- ``challenge-guideline.md`` copy dari Project Backlog yang diberikan Asisten Praktikum.
- ``README.md`` detail tentang project

Praktikum PBO 2021

Accept the group assignment —

OOP Final

Before you can accept this assignment, you must create or join a team. Be sure to select the correct team as you won't be able to change this later.


Create a new team


Create a new team

Create team

 changelog

 LICENSE

 README.md

 challenge-guideline.md

Teknis Mingguan

- H-1 Praktikum
 - Scrum Meeting kelompok.
 - Pembuatan Scrum Report Mingguan.
- H+0 Praktikum
 - Semua PJ Kelompok mengadakan Scrum Meeting dengan asprak.
 - Diskusi tentang Scrum Report Mingguan keseluruhan
 - Pemberian saran dari Asisten Praktikum

Note: PJ tim berubah setiap minggu.

Assignment!

Tugas yang ini beda nih

Assignment 10 - 1

1. Accept assignment pada link sebelumnya.
2. Adakan Sprint planning dengan Tim. Kemudian buat **Sprint Backlog** dari awal hingga akhir.

Assignment 10 - 2

1. Implementasikan Sprint Backlog untuk Sprint 1 selama 1 minggu.
2. Buat laporan mingguan dengan mengubah file dalam folder changelog pada repository.
3. Target minggu depan: file `sprint-planning` (tugas 1) dan `sprint-1` (tugas 2) sudah beres.

Deadline Pengumpulan

Kelas A:

20 November 2024, 23:59 WIB

Kelas B:

19 November 2024, 23:59 WIB

Waktu yang dilihat adalah waktu last commit.

Jika ada yang commit melewati deadline walaupun sudah commit sebelumnya akan dianggap telat

Terima Kasih!

Do you have any questions? Please use respective class discussion channel on Discord.

Selalu ada waktu untuk terus belajar 📖👉