

LAPORAN PRATIKUM PEMROGRAMAN BERORIENTASI OBJEK (G)

"Tugas 1: Konstruktor"



Dosen Pengampu:

Dyah Sulistyowati R, S.Kom.,M.Kom

Disusun oleh:

Davina Arthamevia (4524210127)

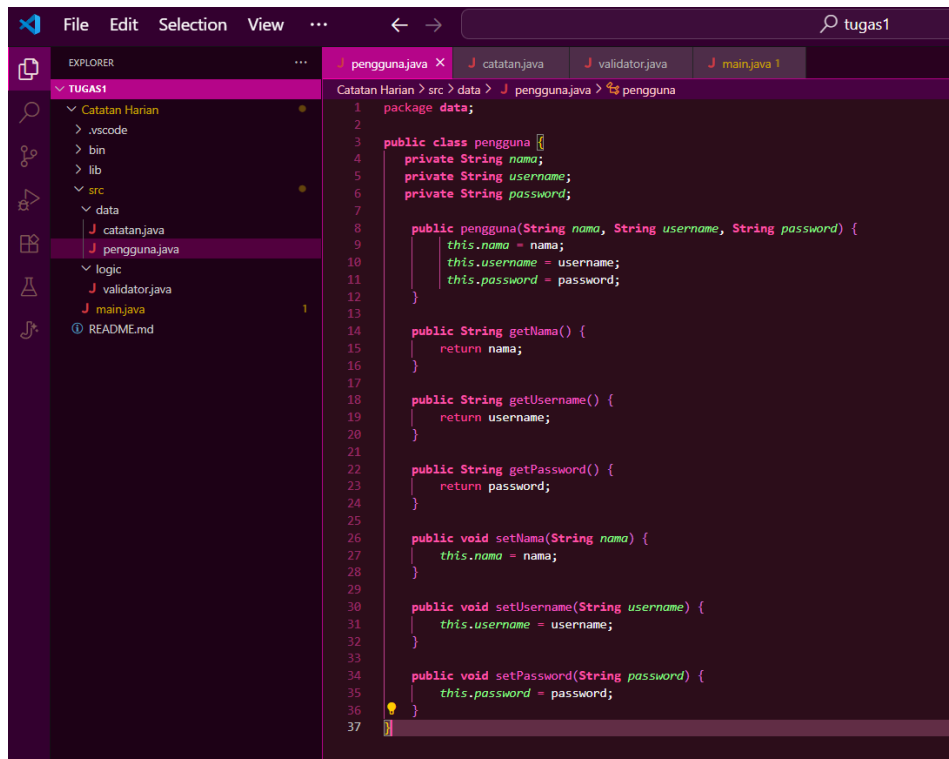
S1 – TEKNIK INFORMATIKA

FAKULTAS TEKNIK UNIVERSITAS PANCASILA

2025

➤ Source Code

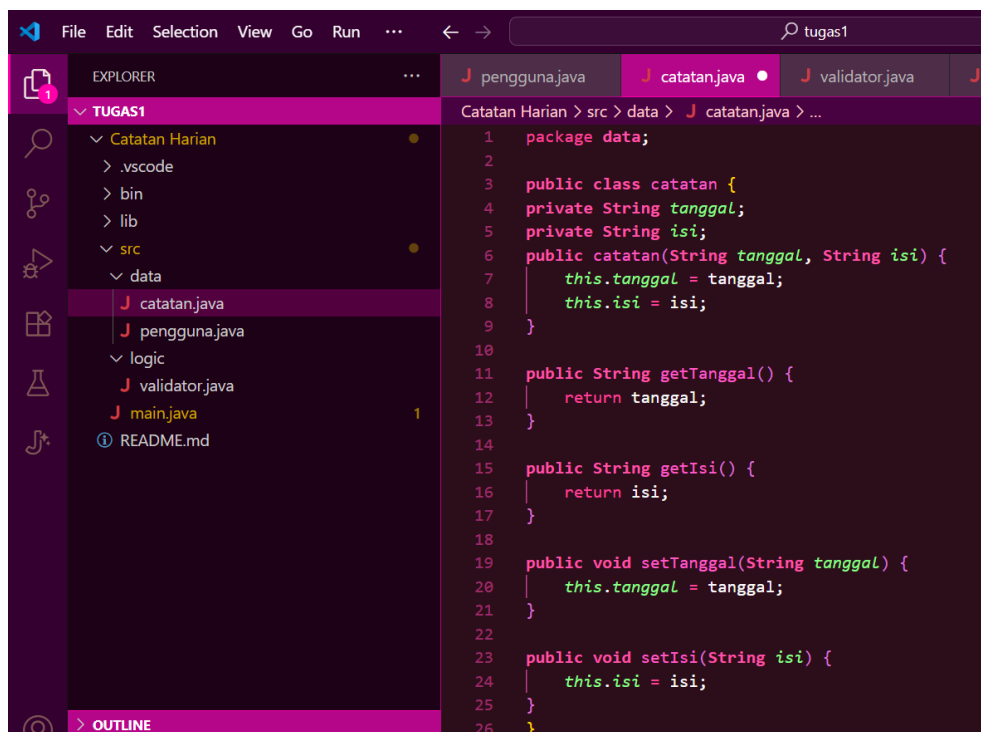
1) Class Pengguna



The screenshot shows the Visual Studio Code editor with the 'pengguna' class open. The Explorer sidebar on the left shows the project structure: 'TUGAS1' containing 'Catatan Harian' (with subfolders '.vscode', 'bin', 'lib', and 'src') and 'src' (with subfolders 'data', 'logic', and 'main.java'). The 'data' folder is expanded, showing 'catatan.java', 'pengguna.java', 'validator.java', and 'main.java'. The 'pengguna.java' file is selected and its code is displayed in the editor. The code defines a 'pengguna' class with private attributes 'nama', 'username', and 'password', a constructor, and getter/setter methods.

```
1 package data;
2
3 public class pengguna {
4     private String nama;
5     private String username;
6     private String password;
7
8     public pengguna(String nama, String username, String password) {
9         this.nama = nama;
10        this.username = username;
11        this.password = password;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public String getUsername() {
19        return username;
20    }
21
22    public String getPassword() {
23        return password;
24    }
25
26    public void setNama(String nama) {
27        this.nama = nama;
28    }
29
30    public void setUsername(String username) {
31        this.username = username;
32    }
33
34    public void setPassword(String password) {
35        this.password = password;
36    }
37 }
```

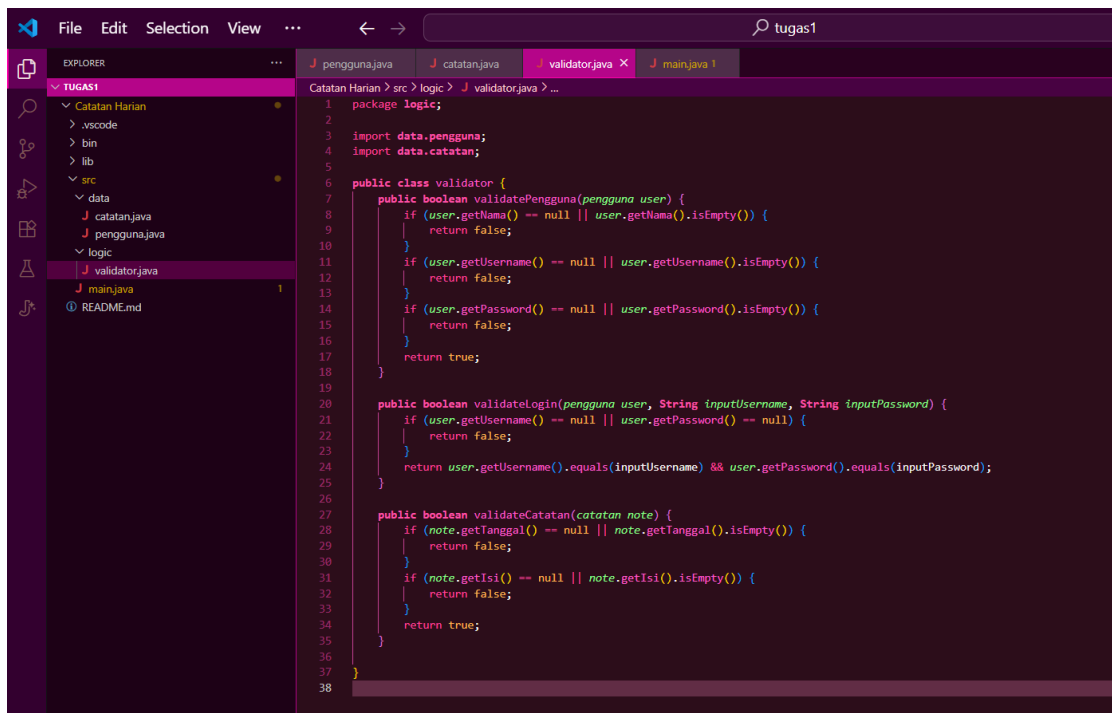
2) Class Catatan



The screenshot shows the Visual Studio Code editor with the 'catatan' class open. The Explorer sidebar on the left shows the project structure: 'TUGAS1' containing 'Catatan Harian' (with subfolders '.vscode', 'bin', 'lib', and 'src') and 'src' (with subfolders 'data', 'logic', and 'main.java'). The 'data' folder is expanded, showing 'catatan.java', 'pengguna.java', 'validator.java', and 'main.java'. The 'catatan.java' file is selected and its code is displayed in the editor. The code defines a 'catatan' class with private attributes 'tanggal' and 'isi', a constructor, and getter/setter methods.

```
1 package data;
2
3 public class catatan {
4     private String tanggal;
5     private String isi;
6     public catatan(String tanggal, String isi) {
7         this.tanggal = tanggal;
8         this.isi = isi;
9     }
10
11    public String getTanggal() {
12        return tanggal;
13    }
14
15    public String getIsi() {
16        return isi;
17    }
18
19    public void setTanggal(String tanggal) {
20        this.tanggal = tanggal;
21    }
22
23    public void setIsi(String isi) {
24        this.isi = isi;
25    }
26 }
```

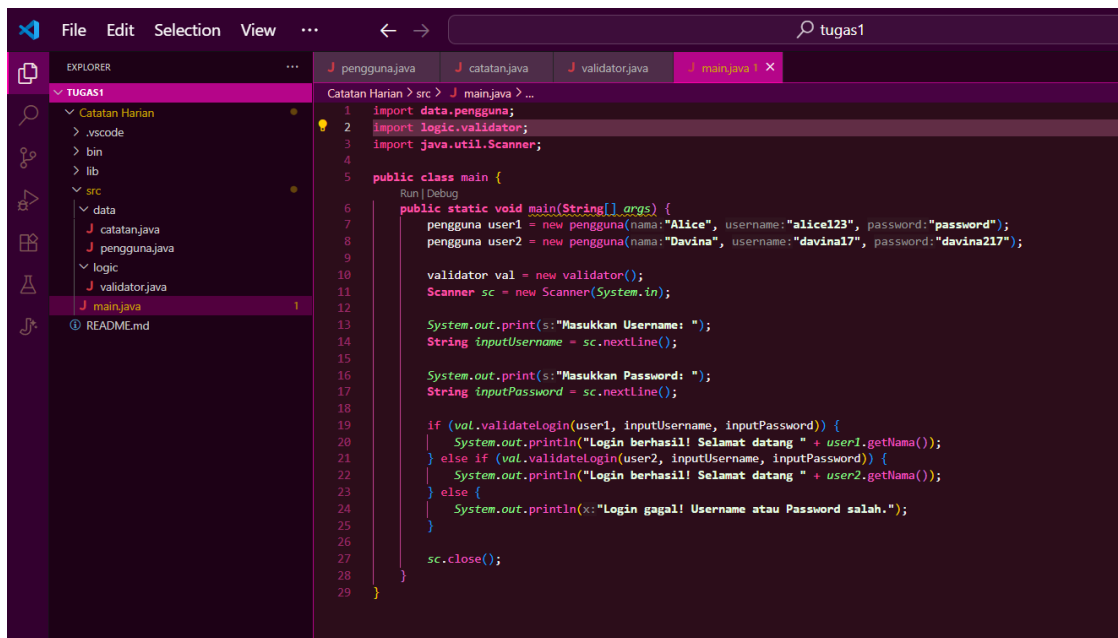
3) Class Validasi



The screenshot shows the VS Code editor with the 'validator.java' file open. The Explorer sidebar on the left shows the project structure: 'TUGAS1' > 'src' > 'logic' > 'validator.java'. The code in the editor is as follows:

```
1 package logic;
2
3 import data.pengguna;
4 import data.catatan;
5
6 public class validator {
7     public boolean validatePengguna(pengguna user) {
8         if (user.getNama() == null || user.getNama().isEmpty()) {
9             return false;
10        }
11        if (user.getUsername() == null || user.getUsername().isEmpty()) {
12            return false;
13        }
14        if (user.getPassword() == null || user.getPassword().isEmpty()) {
15            return false;
16        }
17        return true;
18    }
19
20    public boolean validateLogin(pengguna user, String inputUsername, String inputPassword) {
21        if (user.getUsername() == null || user.getPassword() == null) {
22            return false;
23        }
24        return user.getUsername().equals(inputUsername) && user.getPassword().equals(inputPassword);
25    }
26
27    public boolean validateCatatan(catatan note) {
28        if (note.getTanggal() == null || note.getTanggal().isEmpty()) {
29            return false;
30        }
31        if (note.getIsi() == null || note.getIsi().isEmpty()) {
32            return false;
33        }
34        return true;
35    }
36
37 }
38
```

4) Class Main

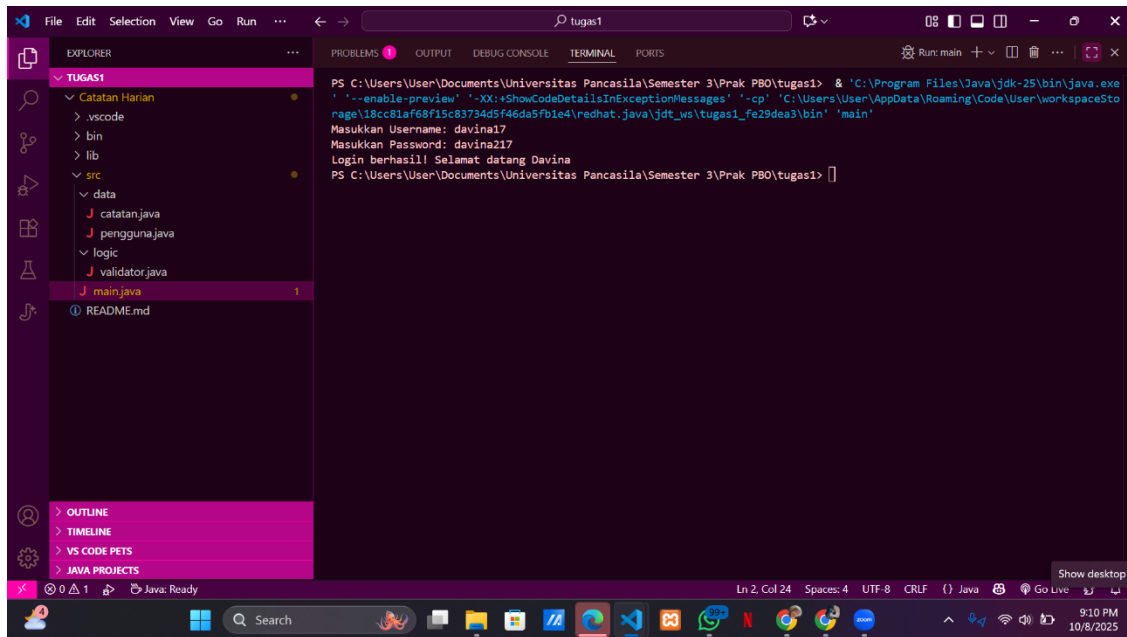


The screenshot shows the VS Code editor with the 'main.java' file open. The Explorer sidebar on the left shows the project structure: 'TUGAS1' > 'src' > 'main.java'. The code in the editor is as follows:

```
1 import data.pengguna;
2 import logic.validator;
3 import java.util.Scanner;
4
5 public class main {
6     public static void main(String[] args) {
7         pengguna user1 = new pengguna(nama:"Alice", username:"alicer123", password:"password");
8         pengguna user2 = new pengguna(nama:"Davina", username:"davinal7", password:"davina217");
9
10        validator val = new validator();
11        Scanner sc = new Scanner(System.in);
12
13        System.out.print(s:"Masukkan Username: ");
14        String inputUsername = sc.nextLine();
15
16        System.out.print(s:"Masukkan Password: ");
17        String inputPassword = sc.nextLine();
18
19        if (val.validateLogin(user1, inputUsername, inputPassword)) {
20            System.out.println("Login berhasil! Selamat datang " + user1.getNama());
21        } else if (val.validateLogin(user2, inputUsername, inputPassword)) {
22            System.out.println("Login berhasil! Selamat datang " + user2.getNama());
23        } else {
24            System.out.println(x:"Login gagal! Username atau Password salah.");
25        }
26
27        sc.close();
28    }
29 }
```

➤ Hasil Running

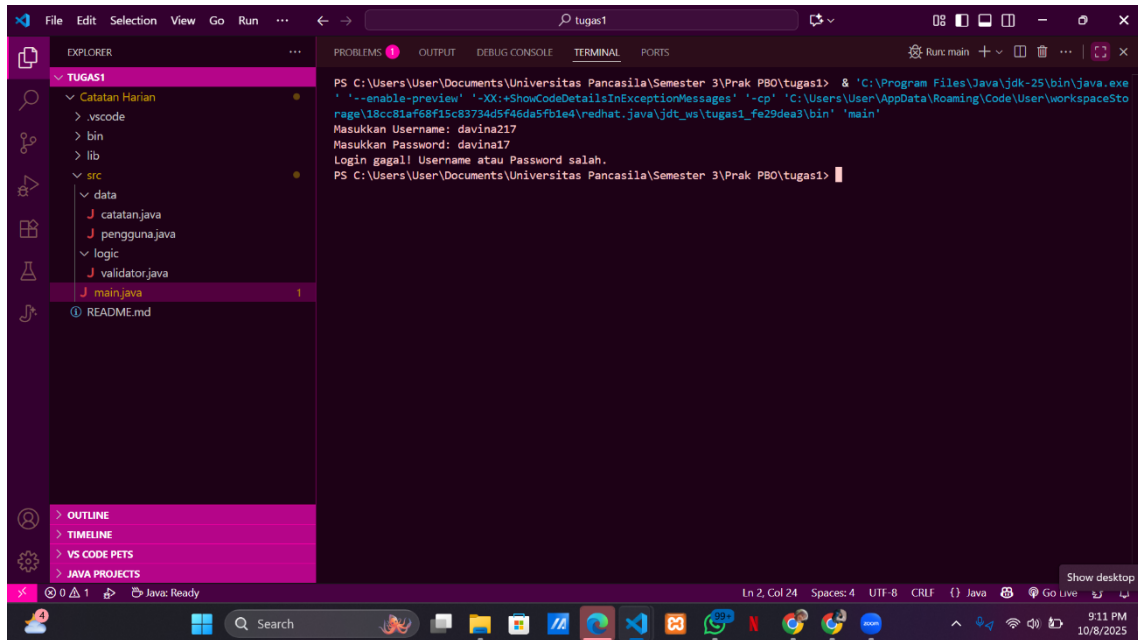
1) Login Berhasil



The screenshot shows the Visual Studio Code interface with a project named 'TUGAS1'. The Explorer panel on the left shows the file structure: 'Catatan Harian', '.vscode', 'bin', 'lib', 'src' (containing 'data', 'catatan.java', 'pengguna.java', 'logic', 'validasi.java'), and 'main.java'. The Terminal panel on the right shows the command prompt output for running the Java application. The output indicates a successful login for the user 'davina17'.

```
PS C:\Users\User\Documents\Universitas Pancasila\Semester 3\Prak PBO\tugas1> & 'C:\Program Files\Java\jdk-25\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\18cc81af68f15c83734d5f46da5fb1e4\redhat.java\jdt_ws\tugas1_fe29dea3\bin' 'main'
Masukkan Username: davina17
Masukkan Password: davina217
Login berhasil! Selamat datang Davina
PS C:\Users\User\Documents\Universitas Pancasila\Semester 3\Prak PBO\tugas1>
```

2) Login Gagal



The screenshot shows the Visual Studio Code interface with the same project 'TUGAS1'. The Explorer panel on the left shows the same file structure as the previous screenshot. The Terminal panel on the right shows the command prompt output for running the Java application. The output indicates a failed login because the password 'davina17' does not match the stored password 'davina217'.

```
PS C:\Users\User\Documents\Universitas Pancasila\Semester 3\Prak PBO\tugas1> & 'C:\Program Files\Java\jdk-25\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\User\AppData\Roaming\Code\User\workspaceStorage\18cc81af68f15c83734d5f46da5fb1e4\redhat.java\jdt_ws\tugas1_fe29dea3\bin' 'main'
Masukkan Username: davina17
Masukkan Password: davina17
Login gagal! Username atau Password salah.
PS C:\Users\User\Documents\Universitas Pancasila\Semester 3\Prak PBO\tugas1>
```

➤ **Penjelasan**

Program ini terdiri dari 3 file utama:

- **Package Data**

Dalam package data ini terdiri dari 2 file

- 1) **Pengguna**

- ✓ Berisi data dari pengguna yang terdiri dari nama, username, dan password.
 - ✓ Untuk menyimpan data pengguna
 - ✓ Terdapat konstruktor supaya dapat membuat objek baru
 - ✓ Terdapat getter dan setter agar dapat mengakses dan mengubah data dengan aman

- 2) **Catatan**

- ✓ Berisi data catatan dari pengguna
 - ✓ Untuk menyimpan catatan dari pengguna
 - ✓ Terdapat konstruktor untuk membuat catatan baru
 - ✓ Terdapat getter dan setter agar dapat mengakses dan mengubah catatan dengan aman

- **Package Logic**

Dalam package logic ini terdiri dari 1 file

- 1) **Validator:**

- ✓ Untuk mengecek apakah data pengguna sudah valid atau belum
 - ✓ Untuk memastikan agar data pengguna tidak kosong

- **Class Main**

- ✓ Memiliki fungsi untuk menjalankan program
 - ✓ Untuk meminta pengguna memasukkan username dan password melalui scanner
 - ✓ Untuk mengecek login dari pengguna