

**LAPORAN RESMI  
PEMROGRAMAN BERORIENTASI OBYEK  
(ADVANCE CLASS DIAGRAM 2)**



**Dosen Pengampu:**

**Saniyatul Mawaddah S.ST., M.Kom**

**Disusun Oleh:**

**Muhammad Aghiitsillah**

**3124521028**

**Teknik Informatika A**

**PROGRAM STUDI D3 TEKNIK INFORMATIKA  
FAKULTAS TEKNIK INFORMATIKA DAN KOMPUTER  
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA –  
PSDKU LAMONGAN 2025**

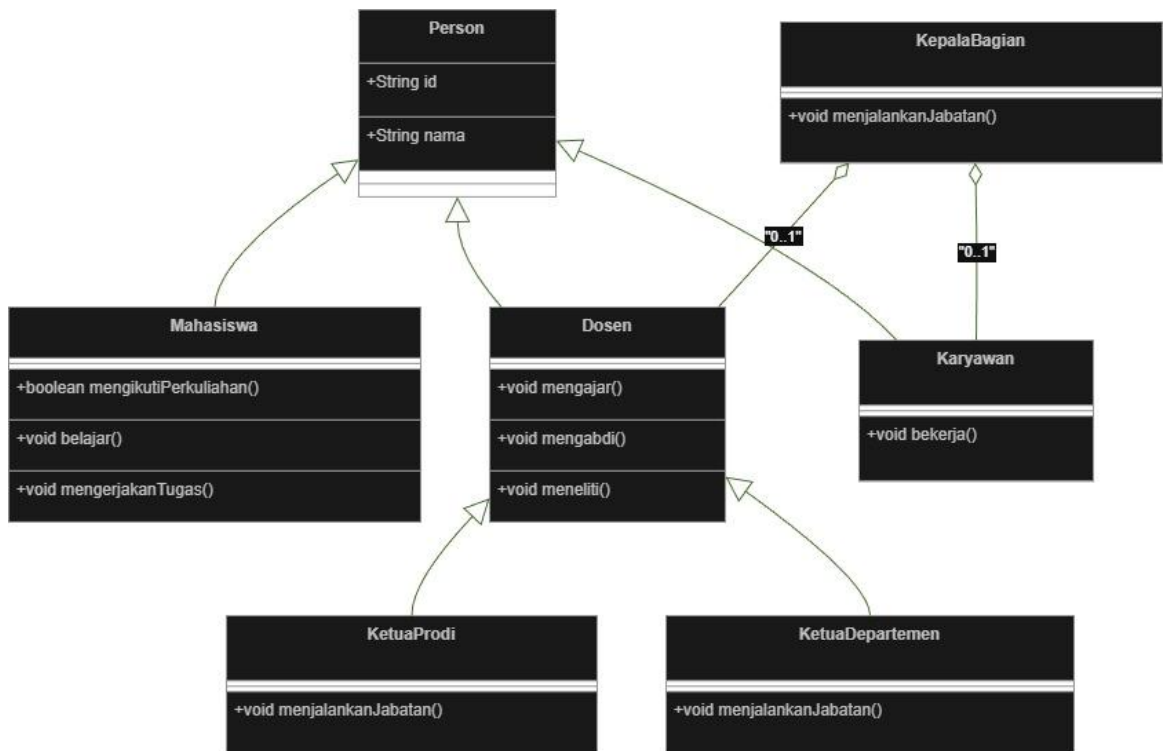
1. Sebuah program memiliki beberapa object sebagai berikut:

Mahasiswa, Dosen, Karyawan, KetuaDepartemen, KetuaProdi, dan KepalaBagian. Object-object tersebut memiliki data dan fungsi sebagai berikut:

- Mahasiswa : id, nama, dosenWali, mengikutiPerkuliahan, belajar, mengerjakanTugas
- Dosen : id, nama, mengajar, mengabdikan, meneliti
- Karyawan : id, nama, bekerja
- KetuaProdi dan KetuaDepartemen pasti adalah seorang dosen, namun memiliki fungsi tambahan yaitu menjalankanJabatan.
- KepalaBagian memiliki fungsi menjalankanJabatan.
- Dalam sistem ada Karyawan dan juga Dosen yang menjadi KepalaBagian.

Silahkan dirancang class diagram untuk kasus tersebut!

**Jawab :**



**Analisa :**

1. Inheritance digunakan karena semua objek (Mahasiswa, Dosen, Karyawan) memiliki atribut dasar yang sama (id, nama).
2. KetuaProdi dan KetuaDepartemen turunan dari Dosen, karena jabatan tersebut hanya dimiliki oleh dosen.
3. KepalaBagian dapat berasal dari Dosen atau Karyawan, maka digambarkan sebagai subclass dari keduanya (multiple inheritance konsep logis).
4. Fungsi menjalankanJabatan() menjadi ciri tambahan dari objek-objek jabatan.
5. Diagram ini menekankan hubungan hierarkis dan peran (role) dalam sistem.

## 2. Buatlah Class Diagram dan Program untuk Register dan Login

Data yang diminta saat Login:

- username
- password

Data yang disimpan saat Register:

- name
- username
- password
- phone
- address

Tampilan awal LoginView

Alur Register => LoginView -> RegisterView -> LoginView

Alur Login => LoginView -> MainView

MainView menampilkan:

data user yang berhasil login

- name
- phone
- address

**Jawab :**



## Kode Pemrograman :

User.java :

```
public class User {
    public String name;
    public String username;
    public String password;
    public String phone;
    public String address;

    public User(String name, String username, String password, String
phone, String address) {
        this.name = name;
        this.username = username;
        this.password = password;
        this.phone = phone;
        this.address = address;
    }
}
```

UserRepository.java

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

public class UserRepository {
    private List<User> users = new ArrayList<>();

    public void save(User user) {
        users.add(user);
    }

    public User findByUsername(String username) {
        for (User u : users) {
            if (u.username.equalsIgnoreCase(username)) {
                return u;
            }
        }
        return null;
    }
}
```

AuthService.java :

```
public class AuthService {
    private UserRepository repo;

    public AuthService(UserRepository repo) {
        this.repo = repo;
    }

    public boolean register(User user) {
        if (repo.findByUsername(user.username) != null) {
            return false; // username sudah ada
        }
        repo.save(user);
    }
}
```

```

        return true;
    }

    public User login(String username, String password) {
        User u = repo.findByUsername(username);
        if (u != null && u.password.equals(password)) {
            return u;
        }
        return null;
    }
}

```

LoginView.java :

```

import java.util.Scanner;

public class LoginView {
    private AuthService authService;

    public LoginView(AuthService authService) {
        this.authService = authService;
    }

    public User show() {
        Scanner sc = new Scanner(System.in);
        System.out.println("=== LOGIN VIEW ===");
        System.out.print("Username : ");
        String username = sc.nextLine();
        System.out.print("Password : ");
        String password = sc.nextLine();

        User user = authService.login(username, password);
        if (user != null) {
            System.out.println("Login berhasil!\n");
        } else {
            System.out.println("Login gagal!\n");
        }
        return user;
    }
}

```

RegisterView.java :

```

import java.util.Scanner;

public class RegisterView {
    private AuthService authService;

    public RegisterView(AuthService authService) {
        this.authService = authService;
    }

    public void show() {
        Scanner sc = new Scanner(System.in);
        System.out.println("=== REGISTER VIEW ===");
    }
}

```

```

        System.out.print("Name: ");
        String name = sc.nextLine();
        System.out.print("Username: ");
        String username = sc.nextLine();
        System.out.print("Password: ");
        String password = sc.nextLine();
        System.out.print("Phone: ");
        String phone = sc.nextLine();
        System.out.print("Address: ");
        String address = sc.nextLine();

        User newUser = new User(name, username, password, phone, address);
        if (authService.register(newUser)) {
            System.out.println("Register berhasil!\n");
        } else {
            System.out.println("Username sudah digunakan!\n");
        }
    }
}

```

MainView.java :

```

import java.util.Scanner;

public class MainView {
    private AuthService authService;
    private LoginView loginView;
    private RegisterView registerView;

    public MainView() {
        UserRepository repo = new UserRepository();
        authService = new AuthService(repo);
        loginView = new LoginView(authService);
        registerView = new RegisterView(authService);
    }

    public void show(User user) {
        System.out.println("=== MAIN VIEW ===");
        System.out.println("Selamat datang, " + user.name);
        System.out.println("Username: " + user.username);
        System.out.println("Telepon : " + user.phone);
        System.out.println("Alamat : " + user.address);
        System.out.println();
    }

    // Jalankan menu utama di sini
    public void start() {
        Scanner sc = new Scanner(System.in);
        while (true) {
            System.out.println("=== MENU UTAMA ===");
            System.out.println("1. Login");
            System.out.println("2. Register");
            System.out.println("3. Keluar");
            System.out.print("Pilih: ");
            String pilih = sc.nextLine();

```

```

        if (pilih.equals("1")) {
            User user = loginView.show();
            if (user != null) {
                show(user);
            }
        } else if (pilih.equals("2")) {
            registerView.show();
        } else if (pilih.equals("3")) {
            System.out.println("Terima kasih!");
            break;
        } else {
            System.out.println("Pilihan tidak valid!\n");
        }
    }
}

// Menjalankan aplikasi (bisa dianggap titik awal)
public static void main(String[] args) {
    new MainView().start();
}
}

```

### Output :

```

PS C:\Kuliah\Semester 3\Praktek pemrograman berorientasi obyek\class diagram 2> java MainView
=== MENU UTAMA ===
1. Login
2. Register
3. Keluar
Pilih: 2
=== REGISTER VIEW ===
Name: agissillah
Username: agis
Password: 12345678
Phone: 081217204394
Address: +0322
Register berhasil!

=== MENU UTAMA ===
1. Login
2. Register
3. Keluar
Pilih: 1
=== LOGIN VIEW ===
Username : agis
Password : 12345678
Login berhasil!

=== MAIN VIEW ===
Selamat datang, agissillah
Username: agis
Telepon : 081217204394
Alamat : +0322

=== MENU UTAMA ===
1. Login
2. Register
3. Keluar
Pilih: 3
Terima kasih!

```

**Analisa :**

- Class User berfungsi untuk menyimpan data pengguna seperti nama, username, password, telepon, dan alamat.
- Class UserRepository bertanggung jawab untuk menyimpan dan mencari data user yang sudah terdaftar.
- Class AuthService menangani logika utama sistem seperti proses registrasi (pengecekan username ganda) dan login (verifikasi username dan password).
- Class LoginView menampilkan form login dan berinteraksi dengan AuthService untuk proses autentikasi.
- Class RegisterView menampilkan form registrasi dan berinteraksi dengan AuthService untuk menambah user baru.
- Class MainView berfungsi menampilkan data user yang berhasil login dan mengatur alur tampilan utama aplikasi.