

Nama : Yunan Faila Sofi

NIM : H1D024033

ShiftLama : G

ShiftBaru : C

Membuat Interface AksesSistem

```
package pert8;
```

```
public interface AksesSistem {
```

```
    // =====  
    // METHOD ABSTRAK (WAJIB DIIMPLEMENTASIKAN DI CLASS ProgrammerMagang)  
    // =====
```

```
    // DEKLARASI METHOD 1: login  
    void login(String pin);
```

```
    // DEKLARASI METHOD 2: logout  
    void logout();
```

```
    // =====  
    // DEFAULT METHOD (OPSIONAL DI-OVERRIDE)  
    // =====
```

```
    // DEKLARASI DEFAULT METHOD: getRoleAkses  
    default String getRoleAkses() {  
        return "Staff Biasa";  
    }  
}
```

Membuat Interface KaryawanKontrak

```
package pert8;
```

```
public interface KaryawanKontrak {
```

```
    // =====  
    // METHOD ABSTRAK (WAJIB DIIMPLEMENTASIKAN DI CLASS ProgrammerMagang)
```

```
// =====

// DEKLARASI METHOD 1: hitungGaji
double hitungGaji(int jamKerja);

// DEKLARASI METHOD 2: perpanjangKontrak
void perpanjangKontrak(int durasiBulan);

// =====
// DEFAULT METHOD (OPSIONAL DI-OVERRIDE)
// =====

// DEKLARASI DEFAULT METHOD: getStatusCuti
default String getStatusCuti() {
    return "Tersedia 12 hari";
}
}
```

Class ProgrammerMagang

```
package pert8;
```

```
// CLASS ProgrammerMagang harus MENIMPLEMENTASIKAN (implements)
KaryawanKontrak dan AksesSistem
public class ProgrammerMagang implements KaryawanKontrak, AksesSistem {
```

```
// =====
// ATRIBUT/STATE (Dibutuhkan untuk menyimpan data objek)
// =====
```

```
private String nama;
private double gajiPerJam;
private String pinRahasia;
private boolean sedangLogin; // default: false
```

```
// =====
// CONSTRUCTOR
// =====
```

```

public ProgrammerMagang(String nama, double gajiPerJam, String pinRahasia) {
    this.nama = nama;
    this.gajiPerJam = gajiPerJam;
    this.pinRahasia = pinRahasia;
    this.sedangLogin = false; // Inisialisasi atribut sedangLogin dengan nilai default 'false'.
}

// =====
// IMPLEMENTASI METHOD DARI KaryawanKontrak
// =====

@Override
public double hitungGaji(int jamKerja) {
    // LOGIKA: Hitung gaji (jamKerja * gajiPerJam) dan tampilkan hasilnya.
    double totalGaji = jamKerja * this.gajiPerJam;
    System.out.printf("Gaji %s (%d jam) adalah: Rp %.1f\n", this.nama, jamKerja, totalGaji);
    return totalGaji;
}

@Override
public void perpanjangKontrak(int durasiBulan) {
    // LOGIKA: Tampilkan pesan konfirmasi perpanjangan kontrak.
    System.out.println("Kontrak diperpanjang " + durasiBulan + " bulan.");
}

@Override
public String getStatusCuti() {
    // LOGIKA: Kembalikan nilai String yang spesifik untuk magang: "Tersedia 5 hari".
    return "Tersedia 5 hari";
}

// =====
// IMPLEMENTASI METHOD DARI AksesSistem
// =====

@Override
public void login(String pin) {
    // LOGIKA: Cek apakah pin yang diterima sama dengan pinRahasia.

```

```

    if (pin.equals(this.pinRahasia)) {
        // Jika sama, ubah sedangLogin = true dan tampilkan pesan berhasil.
        this.sedangLogin = true;
        System.out.println("Login Berhasil. Selamat datang, " + this.nama + "!");
    } else {
        // Jika tidak, tampilkan pesan gagal.
        System.out.println("Login Gagal: PIN salah.");
    }
}

@Override
public void logout() {
    // LOGIKA: Ubah sedangLogin = false dan tampilkan pesan logout.
    this.sedangLogin = false;
    System.out.println(this.nama + " berhasil logout.");
}

@Override
public String getRoleAkses() {
    // LOGIKA: Kembalikan nilai String yang spesifik untuk magang: "Magang IT".
    return "Magang IT";
}
}

```

1. Alur Kerja Program (Workflow)

Program berjalan secara sekuensial dari kelas UjiSDM. Berikut adalah langkah-langkah logisnya:

A. Inisialisasi Objek (Multiple Implementation):

- a. Program membuat objek andi dari kelas ProgrammerMagang.
- b. Kelas ini unik karena ia "menandatangani kontrak" dengan dua pihak sekaligus: KaryawanKontrak (urusan HRD) dan AksesSistem (urusan Keamanan IT).
- c. Constructor menginisialisasi Nama ("Andi"), Gaji per jam (50.000), dan PIN ("1234").

B. Perhitungan Gaji (Kontrak HRD):

- a. Program memanggil hitungGaji(160).
- b. Sistem menghitung $160 \times 50.000 = 8.000.000$ dan menampilkan hasilnya.

C. Cek Hak Cuti (Default Method Override):

- a. Program memanggil getStatusCuti().
- b. Meskipun di interface KaryawanKontrak jatah defaultnya 12 hari, kelas ProgrammerMagang telah mengubah (override) aturan tersebut menjadi "5 hari" khusus untuk anak magang.

D. Simulasi Keamanan (Kontrak Sistem):

- a. Percobaan Gagal: Program mencoba login("9999"). Logika if mendeteksi PIN salah, akses ditolak.
- b. Percobaan Sukses: Program mencoba login("1234"). PIN cocok, status sedangLogin berubah menjadi true, dan pesan selamat datang muncul.

E. Cek Role & Administrasi:

- a. Program memanggil getRoleAkses(). Hasilnya "Magang IT" (bukan "Staff Biasa" karena sudah di-override).
- b. Program memanggil perpanjangKontrak(6) untuk menampilkan info administrasi.

F. Logout: Program memanggil logout() untuk mengakhiri sesi.

2. Fungsi dan Komponen yang Digunakan

Kode ini mendemonstrasikan kemampuan Java untuk mengatasi kelemahan *Single Inheritance* dengan menggunakan Interface.

A. Konsep Multiple Interface

implements KaryawanKontrak, AksesSistem: Java tidak mengizinkan sebuah class memiliki dua "Bapak" (extends A, B), tetapi mengizinkan sebuah class memiliki banyak "Kontrak/Sertifikasi". Di sini, ProgrammerMagang memiliki fitur gabungan dari karyawan kontrak dan pengguna sistem.

B. Default Method

default keyword: Fitur Java 8 ke atas. Biasanya method di interface tidak boleh ada isinya (abstrak). Namun dengan kata kunci default, kita bisa membuat logika dasar di interface.

- *Contoh:* getStatusCuti di interface punya nilai "12 hari".
- *Override:* Di kelas ProgrammerMagang, method default ini ditimpa menjadi "5 hari". Ini menunjukkan fleksibilitas: bisa pakai yang standar, atau bikin sendiri.

C. Logika Validasi (Encapsulation sederhana)

Validasi PIN: Method login menggunakan if (pin.equals(this.pinRahasia)) untuk memastikan keamanan sederhana sebelum mengubah status login.

3. Hasil Output Program

```
=== PENGUJIAN PROGRAMMER MAGANG ===  
Gaji Andi (160 jam) adalah: Rp 8000000.0  
Status Cuti: Tersedia 5 hari  
Login Gagal: PIN salah.  
Login Berhasil. Selamat datang, Andi!  
Role Akses: Magang IT  
Kontrak diperpanjang 6 bulan.  
Andi berhasil logout.  
  
Process finished with exit code 0
```

Analisis Output:

- Gaji: Sesuai rumus $\$160 \times 50.000\$$.
- Status Cuti & Role: Output menunjukkan nilai yang spesifik untuk Magang ("5 hari" dan "Magang IT"), membuktikan bahwa *Default Method* dari interface berhasil di-override oleh class.