Nama : Yusuf Rafii Ahmad

NIM : H1D024049

Shift Awal : G

Shift Sekarang : E-F

Pertemuan8

Alurnya membuat interface 2 yait interface AksesSstem dan Karyawan Kontrak yang isinya di interface AksesSistem dengan 3 method 2 method abstract untuk loginn dan logout serta method default untuk role,

public interface AksesSistem {

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

    // DEKLARASI METHOD 1: login
    // Menerima satu parameter: String pin
    // Mengembalikan nilai: void
    public abstract  void login(String pin );


    // DEKLARASI METHOD 2: logout
    // Tidak menerima parameter
    // Mengembalikan nilai: void
    public abstract  void logout();

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

    // DEKLARASI DEFAULT METHOD: getRoleAkses
    // Tidak menerima parameter
    // Mengembalikan nilai: String "Staff Biasa"
    default String getRoleAkses() {
        return "Staff Biasa";
    }
}
```

untuk interface KaryawanKontak berisi 3 method 2 method abstract mengembalikan nilai dan default method untuk status cuti

public interface KaryawanKontrak {

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

```
    public double hitungGaji(int jamKerja);
    public void perpanjangKontrak(int durasiBulan);
    // DEKLARASI METHOD 1: hitungGaji
    // Menerima satu parameter: int jamKerja
    // Mengembalikan nilai: double (total gaji)


    // DEKLARASI METHOD 2: perpanjangKontrak
    // Menerima satu parameter: int durasiBulan
    // Mengembalikan nilai: void


    // ============================================================
    // DEFAULT METHOD (OPSIONAL DI-OVERRIDE)
    // ============================================================
    default String getStatusCuti() {
        return "Tersedia 12 hari";
    }
    // DEKLARASI DEFAULT METHOD: getStatusCuti
    // Tidak menerima parameter
    // Mengembalikan nilai: String "Tersedia 12 hari"
}
```

untuk programerMagang memilii 4 atribut private dan mengimolements kedua inrterface tadi dan mgisi isi method abstract dan defaultnya dengan overriding,

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

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

    // Deklarasikan 4 atribut private:
    // 1. String nama
    // 2. double gajiPerJam
    // 3. String pinRahasia
    // 4. boolean sedangLogin
    private String nama;
    private double gajiPerJam;
    private String pinRahasia;
    private boolean sedangLogin;

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

```
// Buat satu constructor yang menerima 3 parameter (nama, gajiPerJam, pinRahasia).
// Inisialisasi atribut sedangLogin dengan nilai default 'false'.
public ProgrammerMagang(String nama, double gajiPerJam, String pinRahasia) {
   this.nama = nama;
   this.gajiPerJam = gajiPerJam;
   this.pinRahasia = pinRahasia;
   this.sedangLogin = false; // Default belum login
}

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

// @Override: IMPLEMENTASIKAN hitungGaji(int jamKerja)
// LOGIKA: Hitung gaji (jamKerja * gajiPerJam) dan tampilkan hasilnya.
@Override
public double hitungGaji(int jamKerja) {
   double total = jamKerja * gajiPerJam;
   System.out.println("Gaji " + nama + " (" + jamKerja + " jam) adalah: Rp " + total);
   return total;
}
// @Override: IMPLEMENTASIKAN perpanjangKontrak(int durasiBulan)
// LOGIKA: Tampilkan pesan konfirmasi perpanjangan kontrak.
public void perpanjangKontrak(int durasiBulan) {
   System.out.println("Kontrak diperpanjang " + durasiBulan + " bulan.");
}
// @Override: IMPLEMENTASIKAN getStatusCuti() (Override Default Method)
// LOGIKA: Kembalikan nilai String yang spesifik untuk magang: "Tersedia 5 hari".
@Override
public String getStatusCuti() {
   return "Status Cuti: Tersedia 5 hari";
}

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

// @Override: IMPLEMENTASIKAN login(String pin)
// LOGIKA: Cek apakah pin yang diterima sama dengan pinRahasia.
//         Jika sama, ubah sedangLogin = true dan tampilkan pesan berhasil.
//         Jika tidak, tampilkan pesan gagal.
@Override
public void login(String pin) {
   if (pin.equals(pinRahasia)) {
      sedangLogin = true;
      System.out.println("Login Berhasil. Selamat datang, " + nama + "!");
   } else {
```

```
            System.out.println("Login Gagal: PIN salah.");
        }
    }
    // @Override: IMPLEMENTASIKAN logout()
    // LOGIKA: Ubah sedangLogin = false dan tampilkan pesan logout.
    @Override
    public void logout() {
        sedangLogin = false;
        System.out.println(nama + " berhasil logout.");
    }
    // @Override: IMPLEMENTASIKAN getRoleAkses() (Override Default Method)
    // LOGIKA: Kembalikan nilai String yang spesifik untuk magang: "Magang IT".
    @Override
    public String getRoleAkses() {
        return "Role Akses: Magang IT";
    }
}
```

Membuat objek andi lalu memnaggil method hitungGaji(),status cuti, method login dan logout, perpanjangan kontrak serta login salah

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

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

    // Deklarasikan 4 atribut private:
    // 1. String nama
    // 2. double gajiPerJam
    // 3. String pinRahasia
    // 4. boolean sedangLogin
    private String nama;
    private double gajiPerJam;
    private String pinRahasia;
    private boolean sedangLogin;

    // ============================================================
    // CONSTRUCTOR
    // ============================================================

    // Buat satu constructor yang menerima 3 parameter (nama, gajiPerJam, pinRahasia).
    // Inisialisasi atribut sedangLogin dengan nilai default 'false'.
    public ProgrammerMagang(String nama, double gajiPerJam, String pinRahasia) {
        this.nama = nama;
```

```
    this.gajiPerJam = gajiPerJam;
    this.pinRahasia = pinRahasia;
    this.sedangLogin = false; // Default belum login
}

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

// @Override: IMPLEMENTASIKAN hitungGaji(int jamKerja)
// LOGIKA: Hitung gaji (jamKerja * gajiPerJam) dan tampilkan hasilnya.
@Override
public double hitungGaji(int jamKerja) {
    double total = jamKerja * gajiPerJam;
    System.out.println("Gaji " + nama + " (" + jamKerja + " jam) adalah: Rp " + total);
    return total;
}
// @Override: IMPLEMENTASIKAN perpanjangKontrak(int durasiBulan)
// LOGIKA: Tampilkan pesan konfirmasi perpanjangan kontrak.
public void perpanjangKontrak(int durasiBulan) {
    System.out.println("Kontrak diperpanjang " + durasiBulan + " bulan.");
}
// @Override: IMPLEMENTASIKAN getStatusCuti() (Override Default Method)
// LOGIKA: Kembalikan nilai String yang spesifik untuk magang: "Tersedia 5 hari".
@Override
public String getStatusCuti() {
    return "Status Cuti: Tersedia 5 hari";
}

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

// @Override: IMPLEMENTASIKAN login(String pin)
// LOGIKA: Cek apakah pin yang diterima sama dengan pinRahasia.
//         Jika sama, ubah sedangLogin = true dan tampilkan pesan berhasil.
//         Jika tidak, tampilkan pesan gagal.
@Override
public void login(String pin) {
    if (pin.equals(pinRahasia)) {
        sedangLogin = true;
        System.out.println("Login Berhasil. Selamat datang, " + nama + "!");
    } else {
        System.out.println("Login Gagal: PIN salah.");
    }
}
// @Override: IMPLEMENTASIKAN logout()
// LOGIKA: Ubah sedangLogin = false dan tampilkan pesan logout.
```

```
    @Override
    public void logout() {
        sedangLogin = false;
        System.out.println(nama + " berhasil logout.");
    }
    // @Override: IMPLEMENTASIKAN getRoleAkses() (Override Default Method)
    // LOGIKA: Kembalikan nilai String yang spesifik untuk magang: "Magang IT".
    @Override
    public String getRoleAkses() {
        return "Role Akses: Magang IT";
    }
}
```

```
=== 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.
```