

LAPORAN PROYEK
PEMROGRAMAN BERORIENTASI OBJEK
UMKM MAMACHAFE



Dosen Pengampu:

Muhammad Fawaz Saputra, S.T., M.Eng.

Disusun Oleh:

Nama: Khairunisa Aprilia

NIM: 2409116060

Kelas: Sistem Informasi B 2024

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN

2025

DAFTAR PUSTAKA

DAFTAR PUSTAKA	1
BAB I PENDAHULUAN.....	2
1.1 Latar Belakang	2
1.2 Tujuan	2
BAB II PEMBAHASAN.....	3
2.1 Penjelasan Source Code	3
2.2 Penjelasan Output	9
BAB III PENUTUP.....	10
3.1 Kesimpulan	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, banyak pelaku usaha kecil dan menengah (UMKM) mulai memanfaatkan teknologi untuk membantu kegiatan operasional mereka. Salah satu contohnya adalah dalam pengelolaan data karyawan, termasuk perhitungan gaji. UMKM Mamachafe, yang bergerak di bidang kuliner dengan konsep kafe bertema *matcha*, memiliki beberapa jenis karyawan, seperti karyawan kontrak dan karyawan paruh waktu. Setiap jenis karyawan memiliki sistem penggajian yang berbeda, sehingga diperlukan sistem yang efisien untuk mengelola dan menghitung total gaji mereka.

Untuk menjawab kebutuhan tersebut, penerapan konsep Pemrograman Berorientasi Objek (Object-Oriented Programming /OOP) menjadi solusi yang tepat. Dengan menggunakan prinsip Encapsulation, Inheritance, dan Polymorphism, sistem penggajian dapat dirancang lebih terstruktur, mudah dikembangkan, serta meminimalkan kesalahan dalam perhitungan. Melalui pembuatan program sederhana ini, diharapkan pengelolaan data karyawan di UMKM Mamachafe dapat dilakukan dengan lebih efisien dan profesional.

1.2 Tujuan

Tujuan dari pembuatan program berbasis OOP ini adalah sebagai berikut:

1. Menerapkan konsep OOP seperti Encapsulation, Inheritance, dan Polymorphism dalam kasus nyata pada UMKM Mamachafe.
2. Membuat sistem penggajian sederhana yang mampu menghitung total gaji berbagai tipe karyawan secara otomatis dan akurat.
3. Memudahkan manajemen UMKM dalam melihat rincian gaji karyawan baik kontrak maupun paruh waktu secara cepat.
4. Meningkatkan efisiensi pengelolaan data karyawan, sehingga proses administrasi menjadi lebih rapi dan sistematis.
5. Menunjukkan implementasi nyata dari pemrograman berorientasi objek dalam mendukung kegiatan operasional bisnis kecil dan menengah.

BAB II

PEMBAHASAN

2.1 Penjelasan Source Code

Program ini menerapkan konsep Pemrograman Berorientasi Objek (Object-Oriented Programming/OOP) dengan fokus pada tiga pilar utama, yaitu Encapsulation, Inheritance, dan Polymorphism. Berikut penjelasan struktur kodenya:

1. Class hitung (Interface)

```
package model;

public interface hitung {
    double hitungGaji();
}
```

Class ini berfungsi sebagai interface yang mendefinisikan kontrak atau aturan bahwa setiap kelas yang mengimplementasikannya harus memiliki metode `hitungGaji()`. Dengan demikian, setiap tipe karyawan wajib memiliki cara masing-masing dalam menghitung gaji.

2. Class mamaChafe (Main)

```
package main;

import model.*;

public class mamaChafe {
    public static void main(String[] args) {
        karyawanKontrak k1 = new karyawanKontrak(3000000, 500000, "Lisa",
"KK00001", "Jl. Patin No.59");
        karyawanParuhWaktu k2 = new karyawanParuhWaktu(100, 15000, "Liya",
"KP00001", "Jl. Nila No.01");
    }
}
```

```

    manajemenGaji manajemen = new manajemenGaji();
    manajemen.tambahKaryawan(k1);
    manajemen.tambahKaryawan(k2);

    manajemen.tampilkanTotalGaji();
}
}

```

Kelas `mamaChafe` berfungsi sebagai kelas utama yang menjalankan seluruh program sistem penggajian pada UMKM Mamachafe. Di dalam metode `main`, dibuat dua objek karyawan, yaitu `karyawanKontrak` dengan nama Lisa dan `karyawanParuhWaktu` dengan nama Liya. Masing-masing objek diberikan data sesuai jenis karyawan, seperti gaji pokok, bonus, jam kerja, dan tarif per jam.

Selanjutnya, dibuat objek `manajemenGaji` yang berfungsi untuk menampung dan mengelola daftar karyawan. Kedua objek karyawan tersebut kemudian ditambahkan ke daftar menggunakan metode `tambahKaryawan()`.

Terakhir, program memanggil metode `tampilkanTotalGaji()` untuk menampilkan nama, tipe karyawan, serta total gaji masing-masing di konsol, yang menjadi bukti penerapan konsep polymorphism dalam program ini.

3. Class karyawan (Encapsulation & Abstraction)

```

package model;

public abstract class karyawan implements hitung {
    private String nama;
    private String nip;
    private String alamat;

    public karyawan(String nama, String nip, String alamat) {
        this.nama = nama;
        this.nip = nip;
        this.alamat = alamat;
    }
}

```

```

    }

    public String getNama() {
        return nama;
    }

    public String getNip() {
        return nip;
    }

    public String getAlamat() {
        return alamat;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public void setNip(String nip) {
        this.nip = nip;
    }

    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    @Override
    public abstract double hitungGaji();
}

```

Class Karyawan merupakan kelas induk (parent) yang bersifat abstrak. Artinya, kelas ini tidak dapat di-instantiasi secara langsung, tetapi menjadi dasar bagi subclass-nya.

Atribut nama, nip, dan alamat dibuat private, menunjukkan penerapan Encapsulation, agar data tidak bisa diakses langsung dari luar kelas.

Metode hitungGaji() dibuat abstract, yang berarti setiap turunan dari kelas ini harus mendefinisikan cara menghitung gajinya masing-masing.

4. Class karyawanKontrak (Inheritance & Polymorphism)

```
package model;

public class karyawanKontrak extends karyawan {
    private int gajiPokok;
    private int bonusProyek;

    public karyawanKontrak(int gajiPokok, int bonusProyek, String nama, String nip, String alamat) {
        super(nama, nip, alamat);
        this.gajiPokok = gajiPokok;
        this.bonusProyek = bonusProyek;
    }

    @Override
    public double hitungGaji() {
        return gajiPokok + bonusProyek;
    }
}
```

Class ini mewarisi (extends) kelas Karyawan, yang menunjukkan konsep Inheritance. Dengan atribut gajiPokok dan bonusProyek, dan juga mendapat atribut turunan dari karyawan yaitu nama, nip, dan alamat.

Metode hitungGaji() di-override untuk menghitung gaji berdasarkan formula: $Gaji = Gaji Pokok + Bonus Proyek$.

5. Class karyawanParuhWaktu (Inheritance & Polymorphism)

```

package model;

public class karyawanParuhWaktu extends karyawan {
    private int jamKerja;
    private int tarifPerJam;

    public karyawanParuhWaktu(int jamKerja, int tarifPerJam, String nama,
String nip, String alamat) {
        super(nama, nip, alamat);
        this.jamKerja = jamKerja;
        this.tarifPerJam = tarifPerJam;
    }

    @Override
    public double hitungGaji() {
        return jamKerja * tarifPerJam;
    }
}

```

Class ini mewarisi (extends) kelas Karyawan, yang menunjukkan konsep Inheritance. Dengan atribut jamKerja dan tarifPerJam, dan juga mendapat atribut turunan dari karyawan yaitu nama, nip, dan alamat.

Metode hitungGaji() di-override untuk menghitung gaji berdasarkan formula:
 $Gaji = \text{Jam Kerja} \times \text{Tarif Per Jam}$.

6. Class manajemenGaji (Polymorphism)

```

package model;

import java.util.*;

public class manajemenGaji {
    ArrayList<karyawan> daftarKaryawan = new ArrayList<>();
}

```


[illegible]

Kelas ManajemenGaji menyimpan daftar objek karyawan menggunakan ArrayList.

Metode tampilkanTotalGaji() melakukan iterasi polimorfik, yaitu memanggil hitungGaji() dari setiap objek Karyawan tanpa perlu tahu tipe aslinya (kontrak atau paruh waktu).

2.2 Penjelasan Output

```

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ 
| \ / | / _ \ | | \ / | / _ \ | / _ \ | | |   | |   |
| . . | / / _ \ | | . . | / / _ \ | / \ / | _ | | / / _ \ | | _ | | _
| \| / | | | _ | | \| / | | | _ | | |   | | |   | | |   |
| | | | | | | | | | | | | | | | | | | | \ _ / | | | | | | | | | |
\ _ | | / \ _ | | / \ _ | | / \ _ | | / \ _ | | / \ _ | | / \ _ |
===== Manajemen Gaji Karyawan Mama Chafe =====
Nama: Lisa
Tipe Karyawan: karyawanKontrak
Gaji yang Diterima: Rp 3.500.000
-----
Nama: Liya
Tipe Karyawan: karyawanParuhWaktu
Gaji yang Diterima: Rp 1.500.000
-----
```

Setelah program dijalankan, sistem akan menampilkan daftar gaji seluruh karyawan yang telah dimasukkan ke dalam daftar manajemen. Pada output, akan muncul informasi nama karyawan, tipe karyawan, serta total gaji yang diterima masing-masing. Misalnya, untuk karyawan kontrak bernama Lisa, total gaji yang ditampilkan merupakan hasil penjumlahan antara gaji pokok sebesar Rp3.000.000 dengan bonus proyek sebesar Rp500.000, sehingga totalnya menjadi Rp3.500.000. Sedangkan untuk karyawan paruh waktu bernama Liya, total gaji dihitung berdasarkan hasil perkalian antara jumlah jam kerja sebanyak 100 jam dengan tarif per jam sebesar Rp15.000, sehingga menghasilkan total gaji Rp1.500.000. Hasil ini menunjukkan bahwa setiap jenis karyawan memiliki cara perhitungan gaji yang berbeda, namun dapat diproses dan ditampilkan melalui satu metode yang sama, yaitu `tampilkanTotalGaji()`. Hal ini membuktikan bahwa konsep **polymorphism** berjalan dengan baik pada sistem penggajian UMKM Mamachafe.

BAB III

PENUTUP

3.1 Kesimpulan

Dari hasil perancangan dan implementasi program sistem penggajian pada UMKM Mamachafe, dapat disimpulkan bahwa konsep Pemrograman Berorientasi Objek (OOP) mampu membantu dalam membuat sistem yang lebih terstruktur, efisien, dan mudah dikembangkan. Penerapan Encapsulation terlihat dari penggunaan atribut private yang melindungi data karyawan, sementara Inheritance diterapkan melalui pewarisan kelas abstrak Karyawan oleh kelas karyawanKontrak dan karyawanParuhWaktu. Konsep Polymorphism juga terbukti berjalan dengan baik melalui metode hitungGaji() yang memiliki perilaku berbeda di setiap subclass namun dapat dipanggil dengan cara yang sama. Dengan adanya sistem ini, proses pengelolaan dan perhitungan gaji karyawan di UMKM Mamachafe menjadi lebih terotomatisasi, akurat, dan mudah untuk diperbarui sesuai kebutuhan di masa depan.