

NAMA:ALGHANY SIHOMBING

KELAS:4IA28

NPM:50421118

---

LA1

## 1. Jenis-jenis Utama dari Design Pattern dalam Pengembangan Perangkat Lunak:

Design pattern dalam pengembangan perangkat lunak umumnya dibagi ke dalam tiga kategori:

- **Creational Patterns:** Mengatur cara pembuatan objek agar sesuai dengan kebutuhan tertentu, seperti:
  - **Singleton:** Memastikan hanya ada satu instance dari sebuah kelas.
  - **Factory Method:** Menghasilkan objek tanpa menentukan kelas yang tepat dari objek yang akan dibuat.
  - **Builder:** Memisahkan konstruksi objek kompleks dari representasinya sehingga objek yang sama dapat dibangun dengan cara berbeda.
- **Structural Patterns:** Mengatur bagaimana objek dan kelas dapat digabungkan menjadi struktur yang lebih besar dan kompleks:
  - **Adapter:** Mengubah antarmuka suatu kelas agar sesuai dengan antarmuka yang diharapkan klien.
  - **Composite:** Mengatur objek menjadi struktur pohon untuk mewakili hierarki bagian dan keseluruhan.
  - **Decorator:** Menambahkan fungsionalitas tambahan ke objek secara dinamis.
- **Behavioral Patterns:** Mengatur interaksi antar objek dan cara pembagian tanggung jawab di antara mereka:
  - **Observer:** Objek dapat mengirim pemberitahuan ke banyak penerima yang tertarik dalam suatu perubahan status.
  - **Strategy:** Memilih algoritma dari sebuah keluarga algoritma yang dapat digunakan secara bergantian.
  - **Command:** Mengubah permintaan menjadi objek sehingga permintaan tersebut dapat dinamis dipanggil, dijadwalkan, atau dibatalkan.

## 2. Lima Manfaat dari ERP (Enterprise Resource Planning):

- **Integrasi Data:** ERP memungkinkan integrasi seluruh informasi bisnis dalam satu sistem terpadu.

- **Efisiensi Operasi:** Mengurangi duplikasi data dan meningkatkan akurasi serta kecepatan akses informasi.
- **Pengambilan Keputusan yang Lebih Baik:** Menyediakan data yang real-time dan akurat untuk mendukung pengambilan keputusan.
- **Peningkatan Produktivitas:** Memungkinkan otomatisasi proses bisnis yang meningkatkan produktivitas dan mengurangi kesalahan manual.
- **Pengurangan Biaya Operasional:** Dengan penggunaan ERP, biaya operasional dapat dikurangi karena sistem yang lebih efisien dan terintegrasi.

### 3. Peran Teknologi Perangkat Lunak Berbasis Komponen dalam Mengurangi Waktu dan Usaha Pengembangan:

Teknologi berbasis komponen membantu mengurangi waktu dan usaha pengembangan perangkat lunak melalui:

- **Reuse (Penggunaan Ulang):** Komponen yang sudah dibuat dapat digunakan kembali di berbagai proyek, mengurangi waktu yang dibutuhkan untuk menulis ulang kode.
- **Modularitas:** Komponen dapat dikembangkan dan diuji secara terpisah, mempercepat siklus pengembangan karena bagian-bagian kecil bisa diintegrasikan ke dalam sistem yang lebih besar.
- **Efisiensi Pengembangan:** Memanfaatkan pustaka komponen yang sudah ada dapat mempercepat waktu pengembangan.
- **Pemeliharaan yang Lebih Mudah:** Komponen yang terpisah dapat diganti atau diperbarui tanpa harus mempengaruhi seluruh sistem.

### 4. Implementasi Teknologi Berbasis Komponen oleh Microsoft, OMG, dan Sun:

- **Microsoft:** Menggunakan **COM (Component Object Model)**, yang memungkinkan objek perangkat lunak untuk berkomunikasi di berbagai lingkungan dan bahasa pemrograman.
- **OMG (Object Management Group):** Menggunakan **CORBA (Common Object Request Broker Architecture)**, yang memungkinkan komunikasi antar aplikasi yang ditulis dalam bahasa pemrograman berbeda dan berjalan di platform yang berbeda.
- **Sun Microsystems:** Mengembangkan **JavaBeans**, yang merupakan komponen modular dan dapat digunakan kembali dalam lingkungan Java

## 1.Input

```
11 // Superclass Burung
12 class Burung {
13     String nama;
14     String warna;
15
16     public Burung(String nama, String warna) {
17         this.nama = nama;
18         this.warna = warna;
19     }
20
21     public void berkicau() {
22         System.out.println(nama + " berkicau dengan merdu!");
23     }
24
25     public void terbang() {
26         System.out.println(nama + " terbang di langit.");
27     }
28 }
29
30 // Subclass Burung Beo
31 class Beo extends Burung {
32     public Beo(String nama, String warna) {
33         super(nama, warna);
34     }
35
36     public void menirukan() {
37         System.out.println(nama + " bisa menirukan suara manusia!");
38     }
39 }
40
41 // Subclass Burung Merak
42 class Merak extends Burung {
43     public Merak(String nama, String warna) {
44         super(nama, warna);
45     }
46
47     public void memamerkanBulu() {
48         System.out.println(nama + " membuka bulu ekornya yang indah.");
49     }
50 }
```

```

52 // Subclass Burung Kenari
53 class Kenari extends Burung {
54     public Kenari(String nama, String warna) {
55         super(nama, warna);
56     }
57
58     public void bernyanyi() {
59         System.out.println(nama + " bernyanyi dengan merdu!");
60     }
61 }
62
63 // Main class untuk menjalankan program
64 public class Praktikum {
65     public static void main(String[] args) {
66         Beo beo = new Beo("Beo", "Hitam");
67         Merak merak = new Merak("Merak", "Biru");
68         Kenari kenari = new Kenari("Kenari", "Kuning");
69
70         // Memanggil metode dari masing-masing objek
71         beo.berkicau();
72         beo.menirukan();
73
74         merak.berkicau();
75         merak.memamerkanBulu();
76
77         kenari.berkicau();
78         kenari.bernyanyi();
79     }
80 }
81

```

## Output

```

Output - Belajar Java (run)

run:
Beo berkicau dengan merdu!
Beo bisa menirukan suara manusia!
Merak berkicau dengan merdu!
Merak membuka bulu ekornya yang indah.
Kenari berkicau dengan merdu!
Kenari bernyanyi dengan merdu!
BUILD SUCCESSFUL (total time: 0 seconds)

```

## 2. Penjelasan kode dalam Java:

- **Superclass Burung:** Kelas Burung adalah kelas dasar yang memiliki dua atribut, yaitu nama dan warna. Ada dua metode umum yaitu berkicau dan terbang.
- **Subclass Beo, Merak, dan Kenari:** Mereka mewarisi atribut dan metode dari kelas Burung, tapi menambahkan metode khusus untuk setiap jenis burung. Beo memiliki metode menirukan, Merak memiliki memamerkanBulu, dan Kenari memiliki bernyanyi.

- **super(nama, warna):** Digunakan dalam konstruktor setiap subclass untuk memanggil konstruktor dari superclass Burung, agar atribut nama dan warna dapat diinisialisasi

### 3. Kelebihan dari OOP:

- **Modularitas:** Kode yang ditulis dalam paradigma OOP lebih modular karena dibagi menjadi kelas-kelas yang bisa digunakan kembali.
- **Abstraksi:** OOP memungkinkan kita untuk menyembunyikan detail kompleks dari pengguna dan hanya menampilkan antarmuka yang diperlukan.
- **Pewarisan (Inheritance):** Dengan OOP, kita bisa membuat kelas baru dengan mengambil sifat dari kelas yang ada, sehingga kode lebih singkat dan efisien.
- **Polimorfisme:** OOP mendukung konsep polimorfisme, di mana kita bisa menggunakan satu antarmuka untuk bekerja dengan berbagai jenis objek.

### 4. Perbedaan antara Inheritance dan Polymorphism:

- **Inheritance (Pewarisan):** Adalah konsep di mana suatu kelas dapat mewarisi atribut dan metode dari kelas lain. Ini memungkinkan kita membuat hierarki antar kelas dan menurunkan sifat-sifat dari kelas induk (superclass) ke kelas anak (subclass).
- **Polymorphism (Polimorfisme):** Adalah kemampuan dari suatu fungsi atau metode untuk bekerja dengan berbagai jenis objek. Dalam OOP, polimorfisme memungkinkan metode yang sama dipanggil pada objek-objek yang berbeda, dan objek tersebut akan merespons sesuai dengan implementasinya masing-masing.