**Nama : Grandina Dwiningsi Sulastrin**

**NIM : 2201010500**

**Penjelasan Konsep OOP yang Digunakan**

1. **Access Modifier (Modifier Akses):**
   1. Tujuan: Untuk mengatur visibilitas atau aksesibilitas dari variabel dan metode dalam kelas.
   2. Implementasi: Atribut nama, gaji, dan bonus menggunakan private untuk memastikan bahwa data tersebut hanya dapat diakses melalui metode dalam kelas yang bersangkutan.
   3. Manfaat: Menjaga enkapsulasi dan integritas data dengan mencegah akses atau modifikasi langsung dari luar kelas.
2. **Encapsulation (Enkapsulasi):**
   1. Tujuan: Menyembunyikan detail implementasi dari pengguna kelas dan menyediakan antarmuka yang jelas.
   2. Implementasi: Menggunakan akses modifier private untuk atribut dan menyediakan metode getter untuk mengakses nilai atribut tersebut.
   3. Manfaat: Memperbaiki modularitas kode dan memudahkan pemeliharaan serta penggunaan kembali kode.
3. **Inheritance (Pewarisan):**
   1. Tujuan: Memungkinkan pembuatan kelas baru yang mewarisi sifat dan perilaku dari kelas yang sudah ada.
   2. Implementasi: Kelas Manajer mewarisi dari kelas Karyawan dengan sintaks extends.
   3. Manfaat: Mengurangi duplikasi kode dan memungkinkan perluasan fungsionalitas dari kelas yang sudah ada.
4. **Polymorphism (Polimorfisme):**
   1. Tujuan: Memungkinkan penggunaan objek dari kelas turunan sebagai objek dari kelas induk.
   2. Implementasi: Objek Manajer digunakan sebagai tipe Karyawan dalam demonstrasi polimorfisme.
   3. Manfaat: Memungkinkan fleksibilitas dalam penggunaan objek dan meningkatkan kemampuan ekstensi serta pemeliharaan kode.
5. **Kesimpulan:**

Dengan menggunakan konsep OOP ini, aplikasi yang dibangun menjadi lebih terstruktur, modular, dan mudah untuk dikembangkan serta dipelihara. Enkapsulasi melindungi data dan memastikan integritasnya, pewarisan memungkinkan penggunaan kembali kode, dan polimorfisme memberikan fleksibilitas dalam pemrograman.