Tentu! Berikut adalah penjelasan yang lebih rinci untuk setiap konsep dasar OOP dalam bahasa pemrograman Java:

1. Objek:

Objek adalah instance konkret dari suatu kelas. Setiap objek memiliki atribut (variabel) yang merepresentasikan karakteristik atau data yang dimiliki oleh objek tersebut. Selain itu, objek juga memiliki metode (fungsi) yang mendefinisikan perilaku dan operasi yang dapat dilakukan oleh objek. Misalnya, dalam konteks sebuah kelas "Mobil", objek dapat menjadi "Mobil Merah" atau "Mobil Biru" yang memiliki atribut seperti warna, merk, dan nomor plat, serta metode seperti "maju", "mundur", dan "berhenti".

2. Kelas:

Kelas adalah struktur dasar yang digunakan untuk membuat objek. Kelas menyediakan blueprint atau definisi untuk objek yang akan dibuat. Kelas mendefinisikan atribut (variabel) untuk menyimpan data dan metode (fungsi) untuk melakukan operasi atau perilaku tertentu. Dalam bahasa Java, kelas didefinisikan dengan menggunakan kata kunci "class" diikuti oleh nama kelas.

3. Metode:

Metode adalah blok kode yang melakukan tugas tertentu atau mengembalikan nilai. Metode didefinisikan di dalam kelas dan digunakan untuk menerapkan perilaku objek. Metode dapat menerima parameter sebagai input, dan dapat mengembalikan nilai atau hanya melakukan tugas tertentu tanpa mengembalikan nilai. Dalam bahasa Java, metode didefinisikan dengan menggunakan tipe pengembalian (void atau tipe data lainnya), nama metode, dan parameter (jika ada).

4. Pewarisan (Inheritance):

Pewarisan adalah konsep yang memungkinkan kelas baru (subclass atau turunan) untuk mewarisi atribut dan metode dari kelas yang sudah ada (superclass atau induk). Subclass dapat memperluas atau memodifikasi perilaku yang diturunkan dari superclass. Pewarisan memungkinkan untuk menciptakan hierarki kelas dan mendorong penggunaan kembali kode. Dalam bahasa Java, pewarisan didefinisikan dengan menggunakan kata kunci "extends" diikuti oleh nama superclass.

5. Polimorfisme:

Polimorfisme adalah kemampuan suatu objek untuk mengambil banyak bentuk. Dalam bahasa Java, polimorfisme dapat dicapai melalui mekanisme overloading dan overriding. Overloading terjadi ketika ada beberapa metode dengan nama yang sama tetapi dengan parameter yang berbeda. Hal ini memungkinkan pemanggilan metode yang sesuai dengan argumen yang diberikan. Overriding terjadi ketika subclass mengimplementasikan ulang metode yang diwarisi dari superclass. Dengan overriding, objek dari superclass dapat diperlakukan sebagai objek dari subclass, dan pemanggilan metode akan menggunakan implementasi metode dari subclass.

6. Enkapsulasi:

Enkapsulasi adalah konsep untuk menyembunyikan rincian implementasi objek dari pengguna dan hanya mengekspos antarmuka publik yang diperlukan. Enkapsulasi melibatkan penggunaan akses modifier seperti public, private, dan protected untuk mengatur akses ke atribut dan

metode dalam kelas. Atribut yang dideklarasikan sebagai private hanya dapat diakses oleh metode dalam kelas itu sendiri, sedangkan metode yang dideklarasikan sebagai public dapat diakses dari luar kelas. Konsep ini membantu dalam menjaga integritas data dan memungkinkan perubahan implementasi tanpa mempengaruhi pengguna kelas tersebut.

Dengan memahami konsep dasar OOP ini, pemrogram dapat mengorganisir kode menjadi unit yang lebih kecil dan terpisah, yang disebut kelas dan objek. Hal ini memungkinkan untuk pengembangan program yang lebih terstruktur, modular, dan mudah dipelihara. OOP juga memungkinkan pemrogram untuk membagi kompleksitas menjadi bagian-bagian yang lebih kecil, mengurangi duplikasi kode, dan memfasilitasi penggunaan kembali kode.

Tentu! Berikut adalah penjelasan secara detail untuk setiap bagian dari program data mahasiswa berbasis GUI:

1. Kelas Mahasiswa:

- Kelas Mahasiswa adalah superclass yang mewakili mahasiswa umum.

- Pada kelas ini, terdapat dua atribut private: `nama` dan `nim`, yang digunakan untuk menyimpan nama dan nomor induk mahasiswa.

- Konstruktor `Mahasiswa(String nama, String nim)` digunakan untuk menginisialisasi atribut nama dan nim ketika objek Mahasiswa dibuat.

- Terdapat dua metode getter, yaitu `getNama()` dan `getNim()`, yang digunakan untuk mengakses nilai atribut nama dan nim.

- Metode `displayData()` menampilkan data mahasiswa menggunakan dialog JOptionPane.

2. Kelas MahasiswaBaru:

- Kelas MahasiswaBaru adalah subclass dari Mahasiswa yang menggunakan inheritance.

- Kelas ini memiliki atribut tambahan, yaitu `jurusan`, yang mewakili jurusan mahasiswa baru.

- Terdapat konstruktor `MahasiswaBaru(String nama, String nim, String jurusan)` yang digunakan untuk menginisialisasi atribut nama, nim, dan jurusan saat objek MahasiswaBaru dibuat.

- Pada konstruktor, digunakan keyword `super` untuk memanggil konstruktor superclass dan menginisialisasi atribut nama dan nim.

- Terdapat metode getter `getJurusan()` untuk mengakses nilai atribut jurusan.

- Melalui polimorfisme, metode `displayData()` di-overriding untuk menampilkan data mahasiswa baru beserta jurusannya menggunakan dialog JOptionPane.

3. Kelas DataMahasiswaGUI:

- Kelas DataMahasiswaGUI adalah kelas utama yang berfungsi untuk membuat jendela GUI.

- Pada method `main()`, sebuah objek JFrame dibuat dengan judul "Data Mahasiswa".

- JFrame diatur untuk menutup aplikasi ketika tombol close di jendela di klik (`setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE)`).

- Ukuran jendela diatur menggunakan `setSize(300, 200)`.

- Layout manager `FlowLayout()` digunakan untuk mengatur tata letak komponen di dalam jendela.

- Dibuat tombol "Input Data Mahasiswa Baru" menggunakan JButton.

- ActionListener ditambahkan ke tombol untuk menangani aksi pengguna saat tombol diklik.

- Ketika tombol diklik, program akan menampilkan dialog JOptionPane untuk meminta pengguna memasukkan nama, nim, dan jurusan mahasiswa baru.

- Data yang dimasukkan oleh pengguna digunakan untuk membuat objek MahasiswaBaru.

- Metode `displayData()` dari objek MahasiswaBaru dipanggil untuk menampilkan data mahasiswa baru dan jurusan menggunakan dialog JOptionPane.

Program ini memanfaatkan konsep dasar OOP seperti inheritance (pewarisan), polymorphism (polimorfisme), dan encapsulation (enkapsulasi) untuk mengatur dan memanipulasi data mahasiswa secara terstruktur.