Nama : Nadia Sari

NIM : 20220040067

Kelas : TI22J

Tugas : Sesi 2 PBO

Dosen : Alun Sujjada, S.Kom, MT

1. **Jelaskan apa yang dimaksud dengan class dan object!**

**Class** dalam pemrograman berorientasi objek (OOP) adalah **cetak biru atau blueprint dari sebuah objek**. Class digunakan untuk **membuat kerangka dasar**. Analoginya, kita bisa membayangkan class seperti **laptop atau notebook**, yaitu gambaran umum tentang suatu benda. Contoh class dalam bahasa Java:

class Laptop {

// Isi dari class Laptop...

// Isi dari class Laptop...

}

Di sini, Laptop adalah nama class yang menggambarkan kerangka dasar laptop.

**Object**, di sisi lain, adalah **instansi dari sebuah class**. Ketika kita membuat object, kita menggunakan cetakan dari class tersebut. Misalnya, kita bisa membuat object laptop1, laptop2, dan seterusnya berdasarkan class Laptop. Setiap object memiliki karakteristik dan perilaku yang sama seperti yang didefinisikan dalam class. Contoh property (atau atribut) dari class Laptop bisa berupa pemilik, merk, dan ukuran layar.

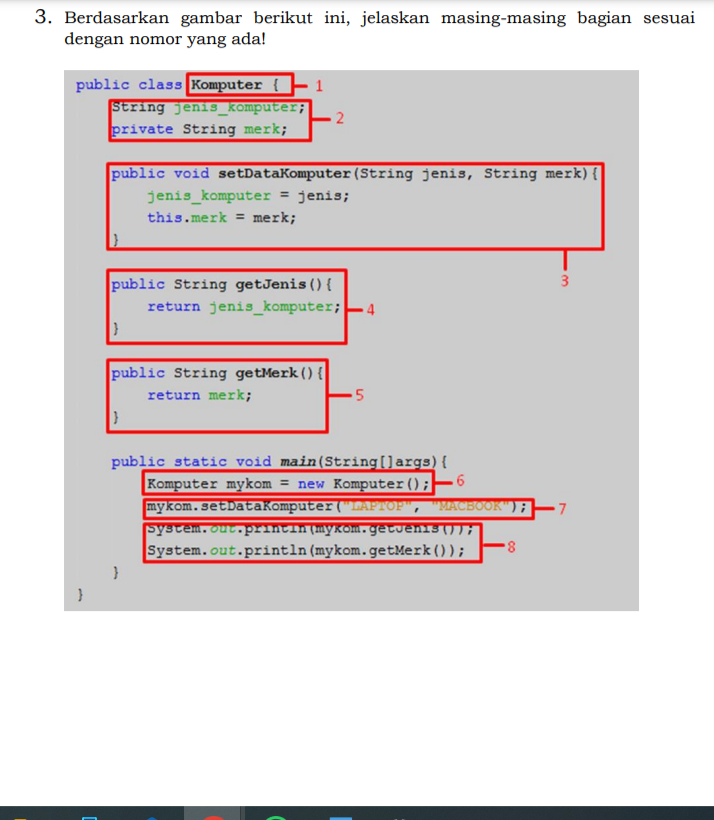
Dengan kata lain:

* **Class** adalah cetak biru yang mendefinisikan apa yang sebuah objek adalah, apa atributnya, dan bagaimana perilakunya.
* [**Object** adalah instansi konkret dari sebuah class yang selalu mengikuti template dari class tersebut](https://www.duniailkom.com/tutorial-oop-java-pengertian-class-object-property-dan-method/)

1. **Sebutkan dan jelaskan jenis-jenis method !**

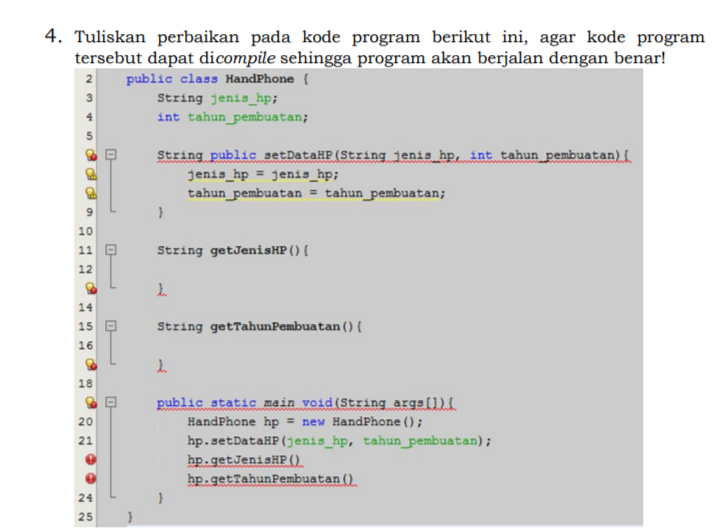
Berikut adalah beberapa jenis metode dalam pemrograman:

1. **Predefined Method**:
   * Metode yang telah ditentukan sebelumnya oleh bahasa pemrograman atau pustaka standar. Contohnya, metode System.out.println() untuk mencetak teks ke konsol.
2. **User-Defined Method**:
   * Metode yang kita buat sendiri untuk melakukan tugas tertentu. Kita dapat menggunakan kembali kode dengan mendefinisikan metode kita sendiri.
3. **Metode Static**:
   * Metode yang terkait dengan kelas itu sendiri, bukan dengan objek yang dibuat dari kelas tersebut. Metode ini dapat dipanggil tanpa membuat objek.
4. **Metode Instance**:
   * Metode yang terkait dengan objek yang dibuat dari suatu kelas. Metode ini memerlukan objek sebagai pemanggilnya.
     + **Accessor Method**: Metode yang digunakan untuk mengakses nilai dari atribut objek.
     + **Mutator Method**: Metode yang digunakan untuk mengubah nilai dari atribut objek.
5. **Abstract Method**:
   * Metode yang hanya dideklarasikan di dalam kelas (tanpa implementasi). Kelas turunan harus mengimplementasikan metode ini.
6. **Factory Method**:
   * Metode yang digunakan untuk membuat objek dari suatu kelas. Ini memisahkan proses pembuatan objek dari klien yang menggunakannya.
7. **Metode Overloading**:
   * Menggunakan nama metode yang sama dengan parameter berbeda. Contoh: metode print dengan parameter tipe data yang berbeda.
8. **Variable Arguments (Var-Args)**:
   * Metode yang dapat menerima jumlah argumen yang bervariasi. Contoh: public void calculate(int... numbers).
9. **Metode finalize()**:
   * Metode yang dipanggil sebelum objek dihapus oleh pemulung sampah (garbage collector).
10. **Berdasarkan gambar berikut ini, jelaskan masing-masing bagian sesuai dengan nomor yang ada!**

****

**JAWAB**

Berikut adalah penjelasan untuk masing-masing bagian sesuai dengan nomor yang ada:

1. **Deklarasi Kelas “Komputer” (Class Declaration)**:
   * Pada bagian ini, kita mendefinisikan kelas dengan nama “Komputer”. Ini adalah kerangka dasar yang akan digunakan untuk membuat objek-objek komputer.
   * Kelas ini akan memiliki beberapa variabel dan metode yang akan kita definisikan lebih lanjut.
2. **Variabel Instance (Instance Variables)**:
   * Variabel “jenis\_komputer” dan “merk” dideklarasikan sebagai variabel instance dalam kelas “Komputer”.
   * Variabel ini akan menyimpan informasi tentang jenis komputer dan merk komputer yang akan digunakan dalam objek-objek yang dibuat dari kelas ini.
3. **Metode setDataKomputer**:
   * Metode ini digunakan untuk mengatur nilai variabel “jenis\_komputer” dan “merk”.
   * Parameter yang diterima adalah nilai untuk kedua variabel tersebut.
   * Dalam contoh ini, metode ini akan mengatur nilai “jenis\_komputer” dan “merk” berdasarkan input yang diberikan.
4. **Metode getJenis**:
   * Metode ini digunakan untuk mendapatkan nilai dari variabel “jenis\_komputer”.
   * Ketika metode ini dipanggil pada objek, ia akan mengembalikan nilai “jenis\_komputer” yang telah diatur sebelumnya.
5. **Metode getMerk**:
   * Metode ini digunakan untuk mendapatkan nilai dari variabel “merk”.
   * Ketika metode ini dipanggil pada objek, ia akan mengembalikan nilai “merk” yang telah diatur sebelumnya.
6. **Metode main**:
   * Metode ini adalah tempat eksekusi program dimulai.
   * Pada contoh ini, kita membuat objek “mykom” dari kelas “Komputer” dan memanggil metode setDataKomputer untuk mengatur nilai variabel.
   * ****Kemudian, kita mencetak nilai dari variabel “merk” menggunakan metode getMerk.

public class Laptop {

String jenis\_Laptop;

int tahun\_pembuatan;

public void setDataLaptop(String jenis, int tahun){

jenis\_hp = jenis;

tahun\_pembuatan = tahun;

}

public String getLaptop(){

return jenis\_hp;

}

public int getTahunPembuatan(){

return tahun\_pembuatan;

}

public static void main(String[] args) {

HandPhone hp = new Laptop();

hp.setDataHandPhone("Iphone 15 Pro Max", 2023);

System.out.println(hp.getLaptop());

System.out.println(hp.getTahunPembuatan());

}

}