

MATA KULIAH : PEMROGRAMAN BERORIENTASI OBJEK
SESI PERTEMUAN : II (DUA)
MATERI : TUGAS CLASS & OBJECT
DOSEN : ALUN SUJJADA, S.KOM., M.T

Nama : Yulva Cintakandida
Kelas : TI21A
NIM : 20210040079

1. Jelaskan apa yang dimaksud dengan class dan object!
2. Sebutkan dan jelaskan jenis-jenis method !
3. Berdasarkan gambar berikut ini, jelaskan masing-masing bagian sesuai dengan nomor yang ada!

```
public class Komputer {  
    String jenis_komputer;  
    private String merk;  
  
    public void setDataKomputer(String jenis, String merk){  
        jenis_komputer = jenis;  
        this.merk = merk;  
    }  
  
    public String getJenis(){  
        return jenis_komputer;  
    }  
  
    public String getMerk(){  
        return merk;  
    }  
  
    public static void main(String[] args){  
        Komputer mykom = new Komputer();  
        mykom.setDataKomputer("LAPTOP", "MACBOOK");  
        System.out.println(mykom.getJenis());  
        System.out.println(mykom.getMerk());  
    }  
}
```

4. Tuliskan perbaikan pada kode program berikut ini, agar kode program tersebut dapat di compile sehingga program akan berjalan dengan benar!

```
2 public class HandPhone {  
3     String jenis_hp;  
4     int tahun_pembuatan;  
5  
6     String setDataHP(String jenis_hp, int tahun_pembuatan){  
7         jenis_hp = jenis_hp;  
8         tahun_pembuatan = tahun_pembuatan;  
9     }  
10  
11     String getJenisHP(){  
12  
13     }  
14  
15     String getTahunPembuatan(){  
16  
17     }  
18  
19     public static void main void(String args[]){  
20         HandPhone hp = new HandPhone();  
21         hp.setDataHP(jenis_hp, tahun_pembuatan);  
22         hp.getJenisHP()  
23         hp.getTahunPembuatan()  
24     }  
25 }
```

Jawab :

1. Class merupakan cetak biru atau kerangka dalam pembuatan program, sedangkan object adalah hasil instance atau penciptaan dari sebuah class. Class dan object adalah sebuah konsep yang saling berkaitan erat dan tidak dapat dipisahkan.
2.
 - Method *setter* (prosedur) digunakan untuk memberikan nilai ataupun untuk menampilkan nilai dari variabel, sehingga tidak memerlukan return value. Inisial awal menggunakan *set* dan menggunakan keyword *void*.
 - Method *getter* (fungsi) digunakan untuk mengambil nilai dari variabel, sehingga membutuhkan *return value*. Tidak memerlukan parameter dan inisial awalnya adalah *get*.
3.
 - Bagian 1
Deklarasi pembuatan *class* dengan nama *Komputer* dan menggunakan modifier *public*.
 - Bagian 2
Deklarasi *atribut* yang digunakan yang mana ada atribut *jenis_komputer* yang menggunakan tipe data *String* dan ada *merk* yang menggunakan tipe data *String* dan menggunakan modifier *private*.
 - Bagian 3
Deklarasi pembuatan *Setter* dengan nama *setDataKomputer* yang memiliki 2 parameter yaitu *String jenis* dan *String merk*. Menggunakan keyword *void* dan menggunakan modifier *public*.
 - Bagian 4
Deklarasi pembuatan *Getter* dengan nama *getJenis* bertipe data *String* dan menggunakan modifier *public* serta menggunakan *return jenis_komputer* untuk mengambil nilai.
 - Bagian 5
Deklarasi pembuatan *Getter* dengan nama *getMerk* bertipe data *String* dan menggunakan modifier *public* serta menggunakan *return merk* untuk mengambil nilai.
 - Bagian 6
Deklarasi menciptakan sebuah *object* dengan nama *mykom* dan menggunakan keyword *new*.
 - Bagian 7
Deklarasi pemanggilan *Setter* dengan 2 parameter yang berisi *"LAPTOP"* dan *"MACBOOK"*.
 - Bagian 8
Deklarasi yang berisi perintah untuk menampilkan data ke layar menggunakan perintah *System.out.println* dan memanggil *object* yang dibuat tadi yaitu *mykom* dan juga *Getter* yang dibuat tadi yaitu *getJenis* dan *getMerk*.

4.

The screenshot displays the NetBeans IDE interface. The main editor window shows the source code for a Java class named `HandPhone`. The code includes two instance variables, `jenis_hp` (String) and `tahun_pembuatan` (int). It defines three methods: `setDataHP` for setting the data, `getJenisHP` for retrieving the phone type, and `getTahunPembuatan` for retrieving the manufacturing year. A `main` method is also present, which creates a `HandPhone` object, sets its data to "SAMSUNG" and 2022, and prints the retrieved values to the console.

```
10  * @author HP 14S
11  */
12  public class HandPhone {
13      String jenis_hp;
14      int tahun_pembuatan;
15
16      public void setDataHP(String jenis_hp, int tahun_pembuatan) {
17          this.jenis_hp = jenis_hp;
18          this.tahun_pembuatan = tahun_pembuatan;
19      }
20
21      public String getJenisHP() {
22          return jenis_hp;
23      }
24
25      public int getTahunPembuatan() {
26          return tahun_pembuatan;
27      }
28
29      public static void main(String[] args) {
30          HandPhone hp = new HandPhone ();
31
32          hp.setDataHP("SAMSUNG", 2022);
33
34          System.out.println(hp.getJenisHP());
35          System.out.println(hp.getTahunPembuatan());
36      }
37  }
38
39
```

Below the editor, the 'Output - HandPhone (run)' window shows the execution results. It displays the output of the `main` method, which prints 'SAMSUNG' and '2022' on separate lines. The status bar at the bottom indicates 'BUILD SUCCESSFUL (total time: 0 seconds)'.

run:
SAMSUNG
2022
BUILD SUCCESSFUL (total time: 0 seconds)