1. **DEFINISI MASALAH**

**Tugas Praktikum**

Sebelum mengerjakan soal di bawah ini tentukan dahulu objek, attribut, behaviour dan class.

1. Buatlah sebuah sistem sederhana yang menyerupai Sistem Informasi Akademik Mahasiswa (SIAM), dengan ketentuan user menginputkan Nama, Nim, IP serta jurusan. Selain itu mahasiswa juga dapat memasukkan kode Mata kuliah, Nama Mata kuliah dan jumlah sks mata kuliah tersebut. Jumlah sks yang di ambil harus sesuai dengan IP yang di dapat pada semester lalu. Buat skenario dengan banyak mahasiswa minimal 3 orang.

**B. SOURCE CODE**

**C. PEMBAHASAN**

**D. SCREENSHOT PROGRAM**

**E. PRAKTIKUM**

A. Class

Pertanyaan

1. Apakah yang disebut dengan variabel instance dan lokal variabel? Jelaskan perbedaanya!

Variabel Instance adalah variabel yang dapat diakses oleh semua method di dalam class,tetapi tidak menggunakan kata static dalam pendeklarasiannya.

Variabel lokal adalah variabel yang dideklarasikan di dalam badan method

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Tidak terdapat kesalahan pada program sehingga program dapat di jalankan

1. Rubah kode pada mainMobil diatas menjadi proses meminta masukan dari user dan buat menjadi interaktif!

|  |  |
| --- | --- |
| 1  5  7  9  10  11  12  13  14  15  16  17  18  20  23  25  26  27  28  29  30  32  33  34  35  37  38  39  40  41  42  43  44 | public static void main(String[] args) {  //instan objek bernama m1  Scanner in= new Scanner (System.in);Mobil m1 = new Mobil();  System.out.print("Masukkan kecepatan mobil 1 : ");  m1.setKecepatan(in.nextInt());  System.out.print("Masukkan Pabrikan mobil 1 : ");  m1.setManufaktur(in.next ());  System.out.print("Masukkan No Plat mobil 1 : ");  m1.setNoPlat(in.next());  System.out.print("Masukkan Warna mobil 1 : ");  m1.setWarna(in.next ());  m1.displayMessage();  System.out.println("================");  //instan objek baru bernama m2  Mobil m2 = new Mobil();  System.out.print("Masukkan kecepatan mobil 2 : ");  m2.setKecepatan(in.nextInt());  System.out.print("Masukkan Pabrikan mobil 2 : ");  m2.setManufaktur(in.next ());  System.out.print("Masukkan No Plat mobil 2 : ");  m2.setNoPlat(in.next ());  System.out.print("Masukkan Warna mobil 2 : ");  m2.setWarna(in.next ());  m2.displayMessage();  System.out.println("================");  //merubah warna dari objek m1  System.out.println("mobil pada objek m1 di rubah menjadi warna hijau");  m1.setWarna("Hijau");  //menampilkan hasil perubahan  m1.displayMessage();  }  } |

4. Tambahkan method pada class mobil bernama setWaktu yang berparameter double, yang kemudian disimpan pada variabel waktu! (Ketetuannya adalah user harus menginputkan dalam satuan jam)

|  |  |
| --- | --- |
| 1  5  7  9  10  11  12  13  14  15 | //class  private double waktu;  double I;  public void setWaktu() {  waktu = I;  }  //main class  System.out.print("Masukkan waktu (dalam satuan jam) : ");  m2.I=in.nextDouble(); |

5. Tambahkan method bernama rubahSekon mempunyai parameter bertipe double dan hanya dapat dipanggil pada class mobil. Method ini memiliki fungsi untuk merubah masukan user yaitu jam menjadi sekon. Method tersebut di panggil pada method setWaktu dengan nilai parameter adalah nilai dari variabel parameter method setWaktu!

|  |  |
| --- | --- |
| 1  5  7  9  10  11  12  13  14  15  16  17  18 | - //class  public void setWaktu() {  waktu = I;  }  public void rubahSekon() {  waktu = I \* 3600;  }  //main  System.out.print("Masukkan waktu (dalam satuan jam) : ");  m1.I=in.nextDouble();  m1.rubahSekon();  m1.displayMessage(); |

6. Tambahkan method pada class mobil dan hanya dapat dipanggil pada class mobil bernama rubahKecepatan yang mempunyai fungsi untuk merubah format kecepatan yang awalnya km/h menjadi m/s. Dipanggil di method setKecepatan!

|  |  |
| --- | --- |
| 1  5  7  9  10  11  12  13  14  15  16  17  18 | - // class  public void setKecepatan(double x) {  kecepatan = x;  }  public void rubahKecepatan() {  kecepatan = (x \*1000)/3600;  }  //main  System.out.print("Masukkan kecepatan mobil 1 : ");  m1.x=in.nextDouble();  m1.rubahKecepatan();  m1.displayMessage(); |

7. Tambahkan method pada class mobil bernama hitungJarak yang mempunyai aksi untuk menghitung jarak yang dapat di tempuh oleh mobil dengan rumus jarak = kecepatan \* waktu!

|  |  |
| --- | --- |
| 1  5  7  9  10  11  12  13 | - //class  public void hitungJarak() {    jarak = (kecepatan \* waktu);  }  //main  m1.hitungJarak();  m1.displayMessage(); |

8. Tambahkan informasi jarak yang dapat ditempuh pada method displayMessage kemudian rubah satuannya yang awalnya m (meter) menjadi km (kilometer)!

|  |  |
| --- | --- |
| 1  5  7  9  10  11  12  13  14  15  16  17 | - // class  public void hitungJarak() {    jarak = (kecepatan \* waktu);  jarak= jarak/1000;  }  public void displayMessage() {  System.out.println("dan jarak yang bisa ditempuh " + jarak + "km");  }  //main  m1.hitungJarak();  m1.displayMessage(); |

9. Mahasiswa A ingin menulis pada sebuah buku tulis yang ingin dia miliki, isi lembar buku tersebut adalah 50 lembar. Setiap harinya ia menulis sebanyak 100 kata perhari yang cukup untuk 1/2 halaman buku. Buatlah rumus untuk menghitung berapa lama ia menghabiskan 1 buku tersebut serta identifikasilah objek, dan karakteristiknya kemudian implementasikan dalam bentuk class.

|  |  |
| --- | --- |
| 1  5  7  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | - //class  public class Book {  int isi = 100;  int day;  public void hitungHari() {  //100 kata/hari dan setengah halaman,jadi 1 lembar 4 hari  int satulembar = 4;  hari = 100\*satulembar;  System.out.println("ia menghabiskan buku tsb selama "+hari+" hari");  }  }  //main  public class MainBook {  public static void main(String[] args) {  Book Tulis = new Book();  tulis.hitungHari();  }  } |

**F. KESIMPULAN**

Kesimpulan untuk bab 1 modul 1 adalah :

1. Object Oriented Programming (OOP) adalah sebuah tata cara pembuatan program dengan menggunakan konsep yang di sebut dengan objek, objek sendiri memiliki data dan prosedur yang di kenal dengan method.
2. Cara membuat instan objek dari sebuah class dilakukan dengan cara menggunakan kata kunci new. Contoh, pada suantu kasus saya mimiliki class bernama mahasiswa dan saya ingin menginstan objek dari class mahasiswa pada class mainMahasiswa dan di beri nama mahasiswa\_1
3. Class adalah struktur dasar dari OOP(Object Oriented Programming).

Memiliki dua tipe : field dan method.

Object adalah sebuah program yang berisi sekumpulan variabel dan method yang berhubungan.

Attribut digunakan sebagai data, di definisikan oleh class, individu, berbeda satu dengan lainnya dan menggambarkan tampilan, status, dan kualitas dari object.

Behavior atau tingkah laku berlaku sebagai ,ethod dan menggambarkan bagaimana sebuah class instance beroperasi.