1. **DEFINISI MASALAH**
2. 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.
3. **SOURCE CODE**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72 | import java.util.Scanner;  public class Modul1 {  public String nama, nim, jurusan, kode, matkul;  public double ip;  public int i, jumlahSKS, totalSKS;  public void setNama(String n) {  nama = n;  }  public void setNim(String a) {  nim = a;  }  public void setJurusan(String j) {  jurusan = j;  }  public void setKode(String k) {  kode = k;  }  public void setMatkul(String m) {  matkul = m;  }  public void setIp(double i) {  ip = i;  }  void masukkan() {  Scanner input = new Scanner(System.in);  System.out.print("Nama Mahasiswa : ");  nama = input.nextLine();  System.out.print("NIM : ");  nim = input.next();  System.out.print("Hasil IP : ");  ip = input.nextDouble();  System.out.print("Jurusan : ");  jurusan = input.next();  }  void getmatkul() {  Scanner input = new Scanner(System.in);  System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");  for (int j = 1; j < 5; j++) {  System.out.print("Masukkan kode mata kuliah anda: ");  kode = input.next();  System.out.print("Masukkan mata kuliah anda: ");  matkul = input.next();  System.out.print("Masukkan jumlah SKS anda: ");  jumlahSKS = input.nextInt();  totalSKS += jumlahSKS;  }  System.out.println("Total jumlah SKS anda: " + totalSKS);  }  void proses() {  if (ip >= 3.0) {  System.out.println("Mahasiswa lulus 5 Mata Kuliah");  System.out.println("Mahasiswa dapat mengambil 19-24 SKS");  } else if (ip >= 2.50 && ip <= 2.99) {  System.out.println("Mahasiswa lulus 5 Mata Kuliah");  System.out.println("Mahasiswa dapat mengambil 19-21 SKS");  } else if (ip >= 2.00 && ip <= 2.49) {  System.out.println("Mahasiswa lulus 4 Mata Kuliah");  System.out.println("Mahasiswa dapat mengambil 16-18 SKS");  } else if (ip >= 1.50 && ip <= 1.99) {  System.out.println("Mahasiswa lulus 3 Mata Kuliah");  System.out.println("Mahasiswa dapat mengambil 12-15 SKS");  } else if (ip <= 1.50) {  System.out.println("Mahasiswa lulus 2 Mata Kuliah");  System.out.println("Mahasiswa dapat mengambil > 12 SKS");  }  }  void displayData() {  System.out.println("========================= Tabel Data Mahasiswa =========================");  System.out.printf("%-20s%-8s%-8s%-8s%-8s%-10s%-10s%s\n", "Nama", "NIM", "Jurusan", "Ip", "Kode", "Matkul", "Jumlah SKS", "Total SKS");  System.out.printf("%-20s%-8s%-8s%-8s%-8s%-10s%-10s%s\n", nama, nim, jurusan, ip, kode, matkul, jumlahSKS, totalSKS);  }  } |

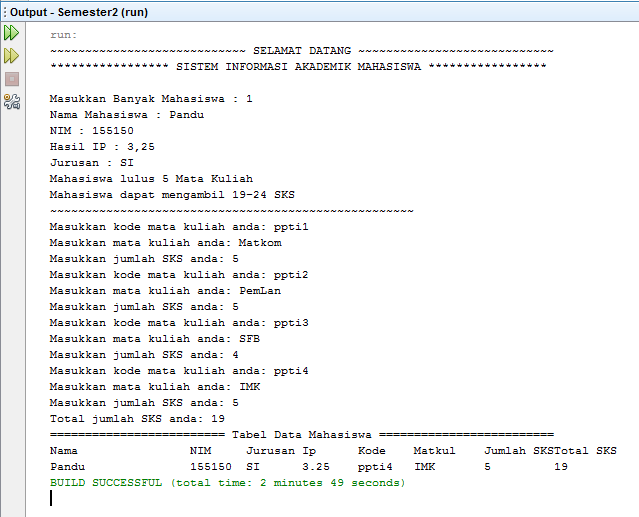
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | import java.util.Scanner;  public class MainModul1 {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  int data;  System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~ SELAMAT DATANG ~~~~~~~~~~~~~~~~~~~~~~~~~~~~");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SISTEM INFORMASI AKADEMIK MAHASISWA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println();  System.out.print("Masukkan Banyak Mahasiswa : ");  data = input.nextInt();  Modul1 first = new Modul1();  for (int i = 0; i < data; i++) {  first.masukkan();  first.proses();  first.getmatkul();  first.displayData();  }  }  } |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  9  10  12  13  15  16  18  19  21  22  24  25  26  27  28  29  30  31  32  33  35  36  37  38  39  40  41  42  43  44  47  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  67  68  69  70 | Perintah untuk mengambil masukan dari scanner  Nama class yaitu modul1  Pendeklarasian variabel global yang memiliki access modifier public yaitu bersifat dikenali oleh semua class dan bertipe data string dengan diberikan objek bernama nama, nim, jurusan, kode, matkul  Pendeklarasian variabel global yang memiliki access modifier public yaitu bersifat dikenali oleh semua class dan bertipe data double dengan diberikan objek bernama ip  Pendeklarasian variabel global memiliki access modifier public yaitu bersifat dikenali oleh semua class dan bertipe data integer dengan diberikan objek bernama i, jumlahSKS, totalSKS  Fungsi atau method void yang bernama method setNama yang memiliki parameter bernama n bertipe data string  Inisialisasi variabel nama bernama n  Fungsi atau method void yang bernama method setNim yang memiliki parameter bernama a bertipe data string  Inisialisasi variabel nim bernama a  Fungsi atau method void yang bernama method setJurusan yang memiliki parameter bernama j bertipe data string  Inisialisasi variabel jurusan bernama j  Fungsi atau method void yang bernama method setKode yang memiliki parameter bernama k bertipe data string  Inisialisasi variabel kode bernama k  Fungsi atau method void yang bernama method setMatkul yang memiliki parameter bernama m bertipe data string  Inisialisasi variabel kode bernama m  Fungsi atau method void yang bernama method setIp yang memiliki parameter bernama i bertipe data double  Inisialisasi variabel kode bernama i  Fungsi atau method void yang bernama method masukkan yang tidak memiliki parameter  Perintah untuk mengambil masukkan dari scanner  Perintah untuk menampilkan keluaran “nama mahasiswa : ”  Perintah untuk mengambil inputan yang bertipe data string lalu disimpan kedalam variabel nama  Perintah untuk menampilkan keluaran “NIM : ”  Perintah untuk mengambil inputan yang bertipe data string lalu disimpan kedalam variabel nim  Perintah untuk menampilkan keluaran “IP : “  Perintah untuk mengambil inputan yang bertipe data double lalu disimpan kedalam variabel ip  Perintah untuk menampilkan keluaran “Jurusan : “  Perintah untuk mengambil inputan yang bertipe data string lalu disimpan kedalam variabel jurusan  Fungsi atau method void yang bernama method getmatkul yang tidak memiliki parameter  Perintah untuk mengambil masukkan dari scanner  Perintah untuk menampilkan keluaran “~~~~~~~~~~~~~~~~~~~~~~~~~~~~  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~”  Perulangan dengan kondisi variabel i yang bertipe data integer dimana i harus sama dengan 1 sampai dengan i kurang dari 5 lalu variabel i dilakukan increment  Perintah untuk menampilkan keluaran “Masukkan kode mata kuliah  anda : “  Perintah untuk mengambil inputan yang bertipe data string lalu disimpan kedalam variabel kode  Perintah untuk menampilkan keluaran “Masukkan mata kuliah anda :”  Perintah untuk mengambil inputan yang bertipe data string lalu disimpan kedalam variabel matkul  Perintah untuk menampilkan keluaran “Masukkan jumlah SKS anda : “  Perintah untuk mengambil inputan yang bertipe data string lalu disimpan kedalam variabel jumlahSKS  totalSKS langsung ditambahkan dengan jumlahSKS lalu hasilnya langsung disimpan ke variabel totalSKS  Perintah untuk menampilkan keluaran “Total jumlah SKS anda : ” lalu langsung ditempel ke variabel totalSKS  Fungsi atau method void yang bernama proses yang tidak memiliki parameter  Percabangan yang memiliki kondisi jika ip >= 3.0  Perintah untuk menampilkan keluaran “mahasiswa lulus 5 Mata  kuliah”  Perintah untuk menampilkan keluaran “mahasiswa dapat mengambil  19-24 SKS”  Percabangan yang memiliki kondisi jika ip >= 2.50 sampai ip <= 2.99  Perintah untuk menampilkan keluaran “mahasiswa lulus 5 Mata  kuliah”  Perintah untuk menampilkan keluaran “mahasiswa dapat mengambil  19-21 SKS”  Percabangan yang memiliki kondisi jika ip >= 2.00 sampai ip <= 2.49  Perintah untuk menampilkan keluaran “mahasiswa lulus 4 Mata  kuliah”  Perintah untuk menampilkan keluaran “mahasiswa dapat mengambil  16-18 SKS”  Percabangan yang memiliki kondisi jika ip >= 1.50 sampai ip <= 1.99  Perintah untuk menampilkan keluaran “mahasiswa lulus 3 Mata  kuliah”  Perintah untuk menampilkan keluaran “mahasiswa dapat mengambil  12-15 SKS”  Percabangan yang memiliki kondisi jika ip <= 1.50  Perintah untuk menampilkan keluaran “mahasiswa lulus 2 Mata  kuliah”  Perintah untuk menampilkan keluaran “mahasiswa dapat mengambil  >12 SKS”  Fungsi atau method void yang bernama displayData yang tidak memiliki parameter  Perintah untuk menampilkan keluaran “======================== Tabel Data Mahasiswa====================”  Perintah untuk menampilkan keluaran yang diberikan jarak spasi secara otomatis yaitu 20, 8, 8, 8, 8, 10,  10 dan digunakan untuk menampilkan “Nama, NIM, Jurusan, Ip, kode,  Matkul, Jumlah SKS, Total SKS  Perintah untuk menampilkan keluaran yang diberikan jarak spasi secara otomatis yaitu 20, 8, 8, 8, 8, 10,  10 dan langsung ditempelkan pada variabel “nama, nim, jurusan, ip, kode, matkul,  jumlahSKS, totalSKS |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | Perintah untuk mengambil masukan dari scanner  Nama class yaitu MainModul1  Tempat untuk menuliskan source code atau syntax dimana PSVM ini adalah tempat pertama kali dituliskan source code atau syntax  Perintah untuk mengambil masukkan dari scanner  Pendeklarsian variabel bernama data bertipe data integer  Perintah untuk menampilkan keluaran “~~~~~~~~~~~~~~~~~~~~~~ SELAMAT DATANG ~~~~~~~~~~~~~~~~~~~~~~~”  Perintah untuk menampilkan keluaran “\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*SISTEM INFORMASI AKADEMIK MAHASISWA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”  Perintah untuk menampilkan keluaran baris kosong atau 1 kali enter  Perintah untuk menampilkan keluaran “Masukkan Banyak Mahasiswa : “  Perintah untuk mengambil inputan yang bertipe data integer lalu disimpan kedalam variabel data  Pendeklarasian pembuatan objek baru di kelas modul1 yang variabel refrensinya bernama first  Perulangan dengan kondisi variabel i yang bertipe data integer dimana i harus sama dengan 0 sampai dengan i kurang dari data lalu variabel i dilakukan increment  Memanggil method atau fungsi yang bernama masukkan dari variabel refrensi first  Memanggil method atau fungsi yang bernama proses dari variabel refrensi first  Memanggil method atau fungsi yang bernama getmatkul dari variabel refrensi first  Memanggil method atau fungsi yang bernama displayData dari variabel refrensi first |

1. **SCREENSHOT PROGRAM**



1. **PRAKTIKUM**

A. CLASS

1. Apakah yang disebut dengan variabel instance dan lokal variabel? Jelaskan perbedaanya!
2. Variabel instance dalam bahasa pemrograman berorientasi objek, variabel instance adalah adalah atribut untuk tiap obyek dari class yang sama jadi objek-objek di dalam sebuah class kemungkinan mempunyai nilai (value) yang berbeda karena atribut diberikan kepada masing-masing objek. Objek menyimpan variabel yang tidak dideklarasikan dengan kata kunci static dalam kategori non-statis, atau dapat berubah-ubah. Nilai yang terkandung dalam variabel tak-statis ini berbeda untuk setiap objeknya. Instance variable adalah variabel yang memiliki nilai yang berbeda di setiap objek. Variabel instance sama halnya dengan variabel class, tetapi tidak menggunakan kata static dalam pendeklarasiannya.Variabel instance terhubung dengan instance dari class. Jadi kita hanya bisa menggunakannya ketika membuat instance dari class. Karena static method tidak terhubung dengan instance dari class, kita tidak bisa menggunakan variabel instance dalam static method dan memasukkannya dalam main method.
3. Variabel lokal adalah variabel yang dideklarasikan di dalam badan method. Jadi kita hanya dapat menggunakan variabel tersebut hanya di dalam method. Method lain dalam class tidak peduli akan keberadaan variabel tersebut. Dan variabel lokal hanya akan ada jika method (yang memiliki variabel lokal tersebut) dieksekusi. Dalam pendeklrasiannya kita tidak perlu menambahkan kata static (seperti yang harus dilakukan pada saat mendeklarasikan variabel class). Jika kata static digunakan dalam pendeklarasian variabel lokal, maka compiler akan menghasilkan pesan error dan menolak untuk melakukan compile terhadap program.
4. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Tidak ditemukan kesalahan.

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

|  |  |
| --- | --- |
|  | import java.util.Scanner;  public class ClassdanObject {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  mobil m1 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m1.setKecepatan(input.nextInt());  System.out.print("Masukkan Manufaktur : ");  m1.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m1.setNoPlat(input.nextLine());  m1.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m1.setWarna(input.nextLine());  System.out.println();  m1.displayMessage();  System.out.println("================");  //instan objek baru bernama m2  mobil m2 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m2.setKecepatan(input.nextInt());  System.out.print("Masukkan Manufaktur : ");  m2.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m2.setNoPlat(input.nextLine());  m2.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m2.setWarna(input.nextLine());  System.out.println();  m2.displayMessage();  System.out.println("================");  //merubah warna dari objek m1  System.out.print("mobil pada objek m1 di rubah menjadi warna : ");  m1.setWarna(input.next());  //menampilkan hasil perubahan  m1.displayMessage();  }  } |

|  |  |
| --- | --- |
|  | package classdanobject;  public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kecepatan;  public void setNoPlat(String s) {  noPlat = s;  }  public void setWarna(String s) {  warna = s;  }  public void setManufaktur(String s) {  manufaktur = s;  }  public void setKecepatan(int i) {  kecepatan = i;  }  public void displayMessage() {  System.out.println("Mobil anda adalah bermerek " + manufaktur);  System.out.println("mempunyai nomor plat " + noPlat);  System.out.println("serta memililki warna " + warna);  System.out.println("dan mampu menempuh kecepatan " + kecepatan);  }  } |

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

|  |  |
| --- | --- |
|  | import java.util.Scanner;  public class ClassdanObject {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  mobil m1 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m1.setKecepatan(input.nextInt());  System.out.print("Masukkan waktu (jam) : ");  m1.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m1.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m1.setNoPlat(input.nextLine());  m1.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m1.setWarna(input.nextLine());  System.out.println();  m1.displayMessage();  System.out.println("================");  //instan objek baru bernama m2  mobil m2 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m2.setKecepatan(input.nextInt());  System.out.print("Masukkan waktu (jam) : ");  m2.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m2.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m2.setNoPlat(input.nextLine());  m2.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m2.setWarna(input.nextLine());  System.out.println();  m2.displayMessage();  System.out.println("================");  //merubah warna dari objek m1  System.out.print("mobil pada objek m1 di rubah menjadi warna : ");  m1.setWarna(input.next());  //menampilkan hasil perubahan  m1.displayMessage();  }  } |

|  |  |
| --- | --- |
|  | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kecepatan;  private double waktu;  public void setNoPlat(String s) {  noPlat = s;  }  public void setWarna(String s) {  warna = s;  }  public void setManufaktur(String s) {  manufaktur = s;  }  public void setKecepatan(int i) {  kecepatan = i;  }    public void setWaktu(double w){  waktu = w;  }  public void displayMessage() {  System.out.println("Mobil anda adalah bermerek " + manufaktur);  System.out.println("mempunyai nomor plat " + noPlat);  System.out.println("serta memililki warna " + warna);  System.out.println("dan mampu menempuh kecepatan " + kecepatan);  System.out.println("dalam waktu " +waktu+ " jam");    }  } |

1. 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!

|  |  |
| --- | --- |
|  | import java.util.Scanner;  public class ClassdanObject {    public static void main(String[] args) {  Scanner input = new Scanner(System.in);  mobil m1 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m1.setKecepatan(input.nextInt());  System.out.print("Masukkan waktu (jam) : ");  m1.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m1.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m1.setNoPlat(input.nextLine());  m1.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m1.setWarna(input.nextLine());  System.out.println();  m1.displayMessage();  System.out.println("================");  //instan objek baru bernama m2  mobil m2 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m2.setKecepatan(input.nextInt());  System.out.print("Masukkan waktu (jam) : ");  m2.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m2.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m2.setNoPlat(input.nextLine());  m2.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m2.setWarna(input.nextLine());  System.out.println();  m2.displayMessage();  System.out.println("================");  //merubah warna dari objek m1  System.out.print("mobil pada objek m1 di rubah menjadi warna : ");  m1.setWarna(input.next());  //menampilkan hasil perubahan  m1.displayMessage();  }  } |

|  |  |
| --- | --- |
|  | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kecepatan;  private double waktu;  private double w;  public void setNoPlat(String n) {  noPlat = n;  }  public void setWarna(String a) {  warna = a;  }  public void setManufaktur(String m) {  manufaktur = m;  }  public void setKecepatan(int k) {  kecepatan = k;  }  public void setWaktu(double w) {  waktu = w;  setRubahsekon(w);  }  private double setRubahsekon(double r) {  w = r \* 3600;  return w;  }    public void displayMessage() {  System.out.println("Mobil anda adalah bermerek " + manufaktur);  System.out.println("mempunyai nomor plat " + noPlat);  System.out.println("serta memililki warna " + warna);  System.out.println("dan mampu menempuh kecepatan " + kecepatan);  System.out.println("dalam waktu " + waktu + " jam");  System.out.println("dirubah menjadi " +w+ " sekon");  }  } |

1. 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!

|  |  |
| --- | --- |
|  | import java.util.Scanner;  public class ClassdanObject {    public static void main(String[] args) {  Scanner input = new Scanner(System.in);  mobil m1 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m1.setKecepatan(input.nextInt());  System.out.print("Masukkan waktu (jam) : ");  m1.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m1.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m1.setNoPlat(input.nextLine());  m1.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m1.setWarna(input.nextLine());  System.out.println();  m1.displayMessage();  System.out.println("================");  //instan objek baru bernama m2  mobil m2 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m2.setKecepatan(input.nextInt());  System.out.print("Masukkan waktu (jam) : ");  m2.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m2.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m2.setNoPlat(input.nextLine());  m2.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m2.setWarna(input.nextLine());  System.out.println();  m2.displayMessage();  System.out.println("================");  //merubah warna dari objek m1  System.out.print("mobil pada objek m1 di rubah menjadi warna : ");  m1.setWarna(input.next());  //menampilkan hasil perubahan  m1.displayMessage();  }  } |

|  |  |
| --- | --- |
|  | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kecepatan, k;  private double waktu;  private double w;  public void setNoPlat(String n) {  noPlat = n;  }  public void setWarna(String a) {  warna = a;  }  public void setManufaktur(String m) {  manufaktur = m;  }  public void setKecepatan(int k) {  kecepatan = k;  setRubahkecepatan(k);  }  private int setRubahkecepatan(int t){  k = t \* 1000/3600;  return k;  }  public void setWaktu(double w) {  waktu = w;  setRubahsekon(w);  }  private double setRubahsekon(double r) {  w = r \* 3600;  return w;  }    public void displayMessage() {  System.out.println("Mobil anda adalah bermerek " + manufaktur);  System.out.println("mempunyai nomor plat " + noPlat);  System.out.println("serta memililki warna " + warna);  System.out.println("dan mampu menempuh kecepatan " + kecepatan + " km/h");  System.out.println("dirubah menjadi " +k+ " m/s");  System.out.println("dalam waktu " + waktu + " jam");  System.out.println("dirubah menjadi " +w+ " sekon");  }  } |

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

|  |  |
| --- | --- |
|  | import java.util.Scanner;  public class ClassdanObject {    public static void main(String[] args) {  Scanner input = new Scanner(System.in);  mobil m1 = new mobil();  System.out.print("Masukkan Kecepatan (km/h) : ");  m1.setKecepatan(input.nextDouble());  System.out.print("Masukkan waktu (jam) : ");  m1.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m1.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m1.setNoPlat(input.nextLine());  m1.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m1.setWarna(input.nextLine());  System.out.println();  m1.displayMessage();  System.out.println("================");  //instan objek baru bernama m2  mobil m2 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m2.setKecepatan(input.nextDouble());  System.out.print("Masukkan waktu (jam) : ");  m2.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m2.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m2.setNoPlat(input.nextLine());  m2.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m2.setWarna(input.nextLine());  System.out.println();  m2.displayMessage();  System.out.println("================");  //merubah warna dari objek m1  System.out.print("mobil pada objek m1 di rubah menjadi warna : ");  m1.setWarna(input.next());  //menampilkan hasil perubahan  m1.displayMessage();  }  } |
|  | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private double kecepatan, waktu, jarak, w, k;  public void setNoPlat(String n) {  noPlat = n;  }  public void setWarna(String a) {  warna = a;  }  public void setManufaktur(String m) {  manufaktur = m;  }  public void setKecepatan(double k) {  kecepatan = k;  getRubahkecepatan(k);  }  private double getRubahkecepatan(double t) {  k = t \* 1000 / 3600;  return k;  }    public void setWaktu(double w) {  waktu = w;  setRubahsekon(w);  }  private double setRubahsekon(double r) {  w = r \* 3600;  getHitungjarak();  return w;  }    private double getHitungjarak () {  jarak = k \* w;  return jarak;  }  public void displayMessage() {  System.out.println("Mobil anda adalah bermerek " + manufaktur);  System.out.println("mempunyai nomor plat " + noPlat);  System.out.println("serta memililki warna " + warna);  System.out.println("dan mampu menempuh kecepatan " + kecepatan + " km/h");  System.out.println("dirubah menjadi " + k + " m/s");  System.out.println("dalam waktu " + waktu + " jam");  System.out.println("dirubah menjadi " + w + " sekon");  }  } |

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

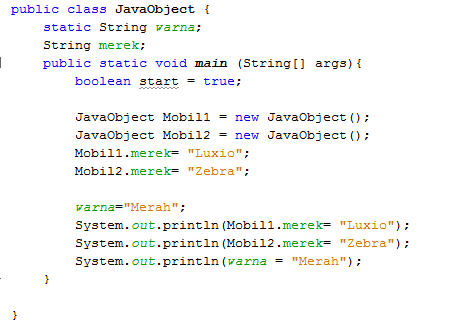
|  |  |
| --- | --- |
|  | import java.util.Scanner;  public class ClassdanObject {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  mobil m1 = new mobil();  System.out.print("Masukkan Kecepatan (km/h) : ");  m1.setKecepatan(input.nextDouble());  System.out.print("Masukkan waktu (jam) : ");  m1.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m1.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m1.setNoPlat(input.nextLine());  m1.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m1.setWarna(input.nextLine());  System.out.println();  m1.displayMessage();  System.out.println("================");  //instan objek baru bernama m2  mobil m2 = new mobil();  System.out.print("Masukkan Kecepatan : ");  m2.setKecepatan(input.nextDouble());  System.out.print("Masukkan waktu (jam) : ");  m2.setWaktu(input.nextDouble());  System.out.print("Masukkan Manufaktur : ");  m2.setManufaktur(input.next());  System.out.print("Masukkan No Plat : ");  m2.setNoPlat(input.nextLine());  m2.setNoPlat(input.nextLine());  System.out.print("Masukkan Warna : ");  m2.setWarna(input.nextLine());  System.out.println();  m2.displayMessage();  System.out.println("================");  //merubah warna dari objek m1  System.out.print("mobil pada objek m1 di rubah menjadi warna : ");  m1.setWarna(input.next());  //menampilkan hasil perubahan  m1.displayMessage();  }  } |

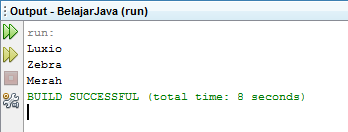
|  |  |
| --- | --- |
|  | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private double kecepatan, waktu, jarak, w, k, jarakBaru;  public void setNoPlat(String n) {  noPlat = n;  }  public void setWarna(String a) {  warna = a;  }  public void setManufaktur(String m) {  manufaktur = m;  }  public void setKecepatan(double k) {  kecepatan = k;  getRubahkecepatan(k);  }  private double getRubahkecepatan(double t) {  k = t \* 1000 / 3600;  return k;  }    public void setWaktu(double w) {  waktu = w;  setRubahsekon(w);  }  private double setRubahsekon(double r) {  w = r \* 3600;  getHitungjarak();  return w;  }    private double getHitungjarak() {  jarak = k \* w;  getRubahjarak();  return jarak;  }    private double getRubahjarak(){  jarakBaru = jarak / 1000;  return jarakBaru;  }  public void displayMessage() {  System.out.println("Mobil anda adalah bermerek " + manufaktur);  System.out.println("mempunyai nomor plat " + noPlat);  System.out.println("serta memililki warna " + warna);  System.out.println("dan mampu menempuh kecepatan " + kecepatan + " km/h");  System.out.println("dirubah menjadi " + k + " m/s");  System.out.println("dalam waktu " + waktu + " jam");  System.out.println("dirubah menjadi " + w + " sekon");  System.out.println("Jarak yang dapat ditempuh " + jarak + " meter");  System.out.println("dirubah menjadi " +jarakBaru+ " km");  }  } |

1. 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.

|  |  |
| --- | --- |
|  | public class bukuMain {  public static void main(String[] args) {  Buku1 beraksi = new Buku1();  beraksi.lembar();  beraksi.displayMessage();  }  } |
|  | public class Buku1 {  private int totalLembar;  private int hari;  private int halaman;  private int totalHalaman;  private int hitunghari;    public void lembar() {  this.totalLembar = 50;  halaman();  }    public int halaman(){  this.hari = 1;  this.halaman = 2 \* this.hari;  this.totalHalaman = this.totalLembar \* this.halaman;  hitungHari();  return this.totalHalaman;  }    public int hitungHari(){  hitunghari = this.totalHalaman \* this.halaman;  return hitunghari;  }    public void displayMessage(){  System.out.println("Lembar Buku : " +this.totalLembar+ " lembar");  System.out.println("Halaman Buku : " +this.totalHalaman+ " halaman");  System.out.println("1/2 halaman = " + this.hari+ " hari");  System.out.println("Jadi, 1 halaman = " +this.halaman+ " hari");  System.out.println("Sehingga, waktu yang diperlukan untuk menghabiskan seluruh halaman buku adalah " +hitunghari+ " hari");  }  } |

1. **KESIMPULAN**
2. OOP (Object Oriented Programing ) merupakan paradigma pemrograman yang popular saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programing yang berarti pula Pemrograman Beorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970. OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya. Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan Object yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau statement. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari.
3. Instansiasi adalah sebuah istilah yang mempunyai arti perwujudan, umumnya istilah ini biasanya digunakan untuk objek-objek yang menginisiasi dari sebuah class, sedangkan sebuah class dapat mempunyai banyak object. Untuk dapat membuat sebuah objek, kita perlu melakukan instansiasi. Instansiasi berfungsi untuk mengubah class menjadi objek (class adalah kumpulan dari objek-objek) ciri-ciri penulisan Instansiasi : **new**. Instansiasi membutuhkan operator **new**untuk mempersiapkan memory sesuai dengan isi kelas.





a. Class

Kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh ‘class of dog‘ adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh orang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

b.Object

Object merupakan perwujudan dari class, setiap object akan mempunyai attribute dan method yang dimiliki oleh class-nya, contohnya: amir, ahmad, yani merupakan object dari class manusia. Setiap object dapat berinteraksi dengan object lainnya meskipun berasal dari class yang berbeda. Object merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun object lain dan mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya. Pada dasarnya semua benda yang ada di dunia dapat dianggap sebagai sebuah object. Rumah, mobil, sepeda motor, meja, dan komputer merupakan contoh-contoh object yang ada di dunia nyata. Object adalah implementasi dari class. Secara sederhananya, dapat dikatakan terdiri dari properti (atribut) dan method.  
Pemahaman singkatnya :  
• Objek merupakan sebuah objek khusus yang digunakan untuk mengakses anggota objek (atribut atau method) dari dalam objek itu sendiri.  
• Apabila terdapat variabel lain yang memiliki nama yang sama dengan nama atribut di dalam objek, “this” dapat digunakan untuk merujuk ke atribut objek.

1. Attribute

Attribute/Variabel merupakan data yang membedakan antara objek satu dengan objek yang lain.Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas, Atribut dapat memiliki hak akses private, public maupun protected.Sebuah atribut yang dinyatakan sebagai private hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas lainnya tidak dapat mengakses atribut ini secara langsung.Sebuah atribut yang dinyatakan sebagai public dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya.Sebuah atribut yang dinyatakan sebagai protected tidak dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.  
       Atribut juga biasa disebut state/ properties/ field merupakan bagian yang dimiliki oleh sebuah class yang tidak melakukan operasi, tetapi kepadanya dilakukan operasi sehingga dapat merubah nilai dari attribute tersebut.Dalam class, atribut disebut sebagai variabel. Atribut dapat membedakan antara satu object dengan object yang lain. Sebagai contoh, pada class : mahasiswa, terdapat object mahasiswa si A, dan object mahasiswa si B.Yang membedakan antara object si A dan si B adalah NPM-nya (Nomor Pokok Mahasiswa) yang merupakan atribut dari object tersebut. Pada atribut, terdapat pula dua istilah variabel, yaitu Instance Variabel dan Class Variable dimana pada tiap variabel memiliki karakteristik tersendiri.

1. Behavior

Behavior adalah tingkah laku atau apa yang dapat dilakukan oleh objek itu atau disebut juga sebagai method.Biasanya dapat digunakan untuk mengubah nilai attribue suatu objek, menerima informasi dari objek lain dan mengirim informasi ke objek lain untuk melakukan task. Behavior adalah kelakuan atau sifat dari Object seperti mobil tadi bisa melaju, membelok, membunyikan klakson dan lain sebagainya. Dalam class, behavior disebut juga sebagai methods. Methods sendiri adalah serangkaian statements dalam suatu class yang menghandle suatu task tertentu. Cara objek berkomunikasi dengan objek yang lain adalah dengan menggunakan method.