1. **DEFINISI MASALAH**

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.

1. **SOURCE CODE**

|  |  |
| --- | --- |
| siam.java | |
| 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  73  74  75  76  77  78  79  80  81  82  83  84 | import java.util.Scanner;  public class siam {  Scanner input = new Scanner(System.in);  double ip;  String nama, nim, jurusan;  int sksMax, sksMasuk, jmlMatkul;  String[] matkul = new String[10];  String[] kodeMatkul = new String[10];  int[] sksInput = new int[10];  siam(String nama, String nim, String jurusan, double ip) {  this.nama = nama;  this.nim = nim;  this.jurusan = jurusan;  this.ip = ip;  }  public void jmlSKS() {  if (ip < 1.49) {  sksMax = 12;  } else if (ip >= 1.50 && ip < 1.99) {  sksMax = 15;  } else if (ip >= 2.00 && ip < 2.49) {  sksMax = 18;  } else if (ip >= 2.5 && ip < 2.99) {  sksMax = 21;  } else if (ip >= 3.00 && ip <= 4.00) {  sksMax = 24;  }  }  public void KRS() {  System.out.println("=============== KRS ==============");  System.out.println("Jumlah SKS yang bisa diambil : " + sksMax);  System.out.print("Jumlah Mata kuliah yang diambil : ");  jmlMatkul = input.nextInt();  int i = 0;  while (sksMasuk < sksMax) {  System.out.print("Kode Mata kuliah ke - " + (i + 1) + " : ");  kodeMatkul[i] = input.next();  System.out.print("Nama Mata kuliah ke - " + (i + 1) + " : ");  matkul[i] = input.next();  System.out.print("Jumlah SKS ke - " + (i + 1) + " : ");  sksInput[i] = input.nextInt();  System.out.println("");  sksMasuk += sksInput[i];  i++;  if (sksMasuk > sksMax) {  System.out.println("Maaf, Jumlah SKS yang anda ambil melebihi jumlah maksimal");  }  if (i == jmlMatkul) {  break;  }  if (sksMasuk <= sksMax) {  System.out.println("=> Sisa SKS yang dapat anda ambil adalah: " + (sksMax - sksMasuk));  }  }  }  public void display() {  if (sksMasuk > sksMax) {  System.out.println("\nMaaf, KRS tidak dapat di tampilkan");  } else {  System.out.println("======= Kartu Rencana Studi ========");  System.out.println("Nama Mahasiswa : " + this.nama);  System.out.println("NIM Mahasiswa : " + this.nim);  System.out.println("Jurusan Mahasiswa : " + this.jurusan + "\n");  System.out.println("Mata Kuliah yang telah diambil :");  for (int i = 0; i < jmlMatkul; i++) {  System.out.println((i + 1) + ". " + kodeMatkul[i] + " " + matkul[i] + " " + sksInput[i] + " SKS");  }  System.out.println("Total SKS yang diambil : " + sksMasuk);  }  }  } |

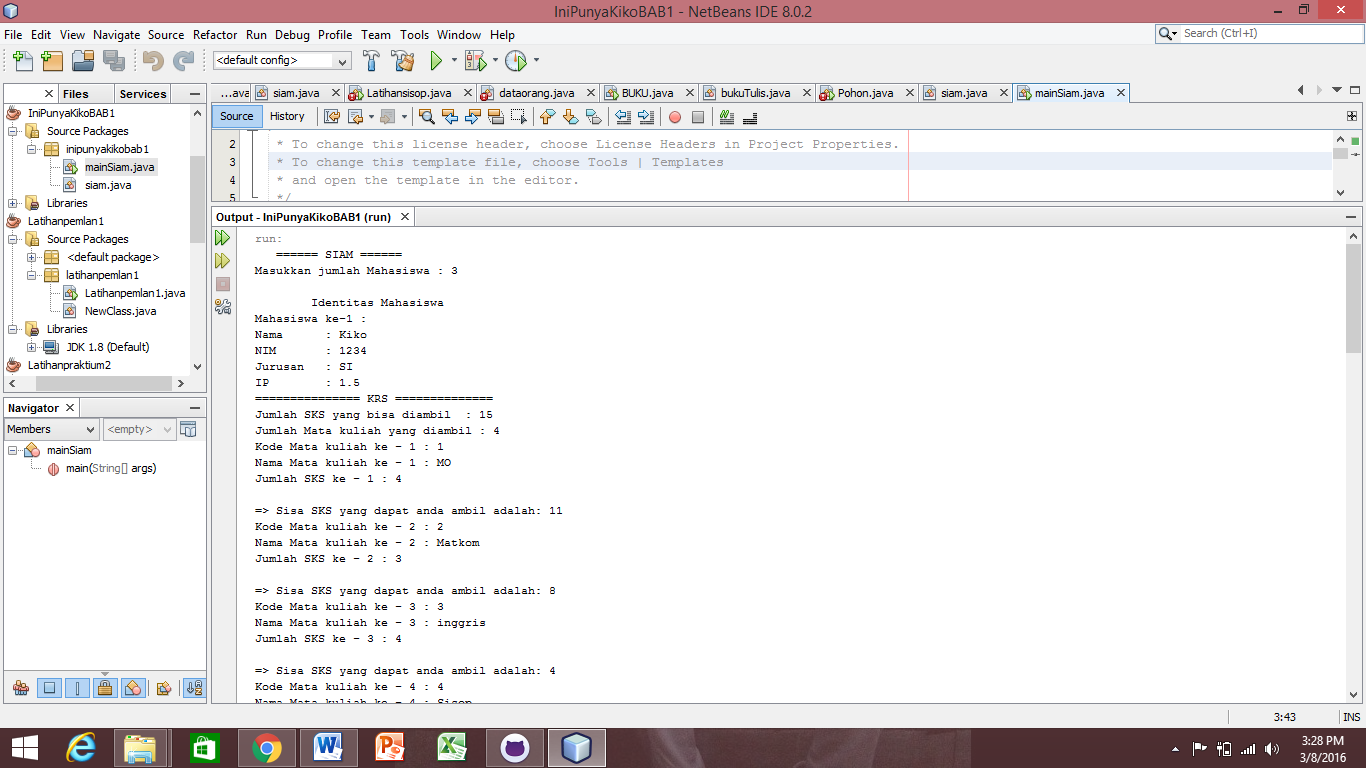
|  |  |
| --- | --- |
| mainSiam.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | import java.util.Scanner;  public class mainSiam {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  System.out.println(" ====== SIAM ====== ");  System.out.print("Masukkan jumlah Mahasiswa : ");  int jmlhMhs = input.nextInt();  double ip;  for (int i = 0; i < jmlhMhs; i++) {  System.out.println("");  System.out.println(" Identitas Mahasiswa ");  System.out.println("Mahasiswa ke-" + (i + 1) + " : ");  System.out.print("Nama : ");  String nama = input.next();  System.out.print("NIM : ");  String nim = input.next();  System.out.print("Jurusan : ");  String jurusan = input.next();  do {  System.out.print("IP : ");  ip = input.nextDouble();  if (ip > 4) {  System.out.println("IP tidak valid, Masukan kembali");  }  } while (ip > 4);  siam s1 = new siam(nama, nim, jurusan, ip);  s1.jmlSKS();  s1.KRS();  s1.display();  }  }  } |

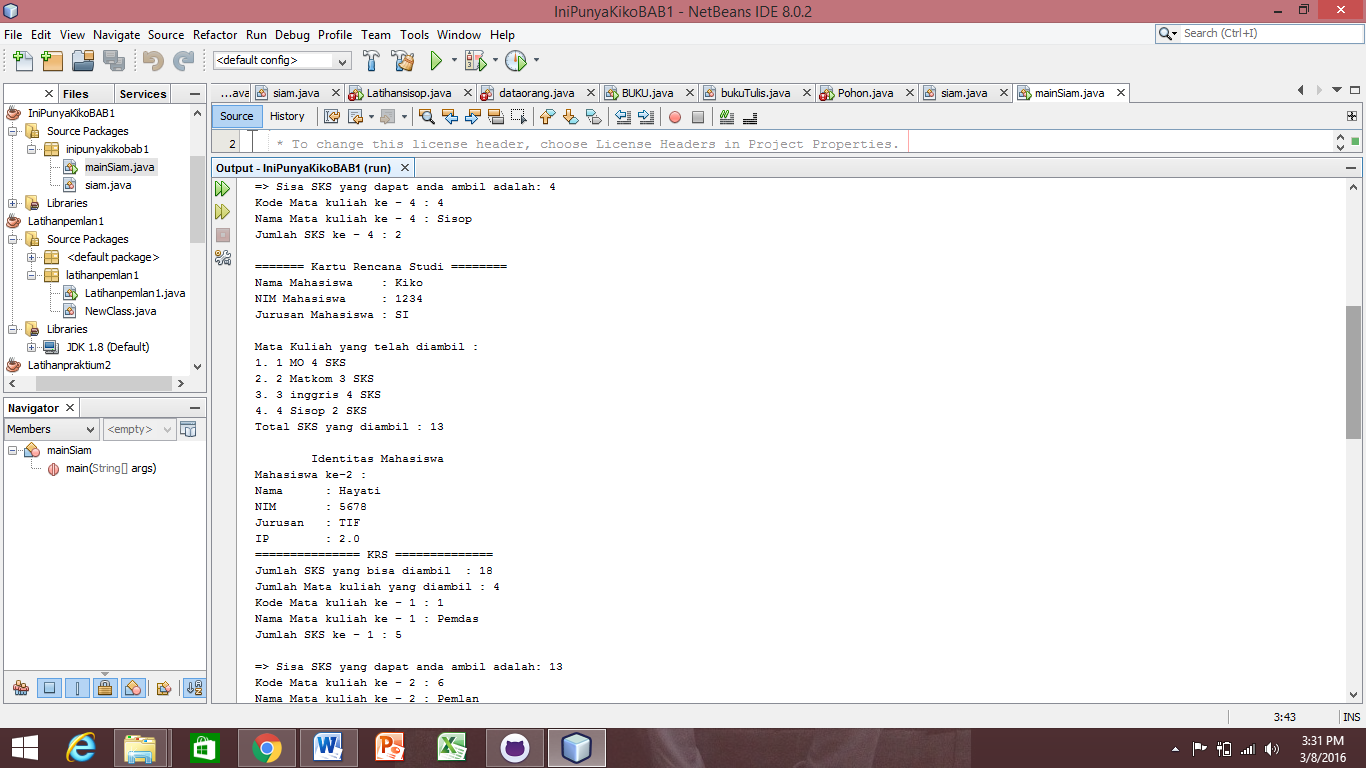
1. **PEMBAHASAN**

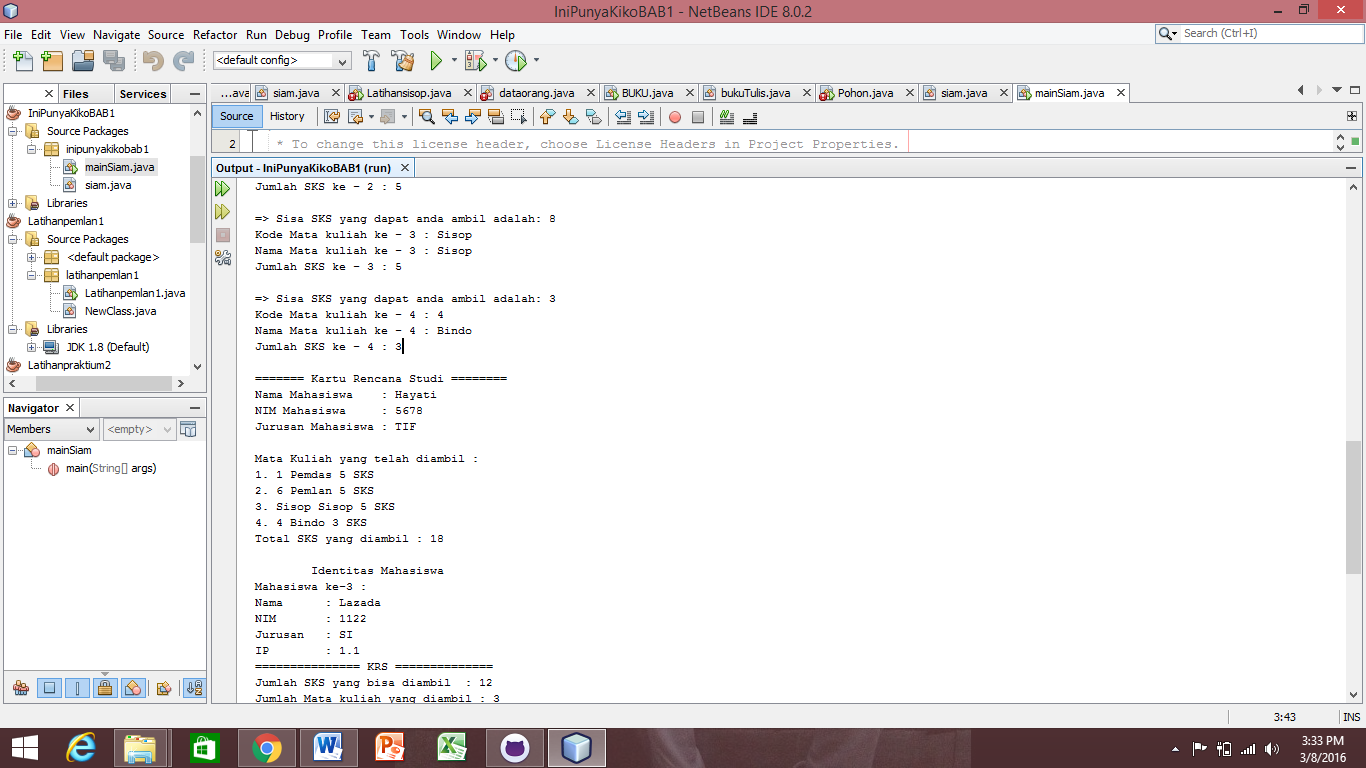
|  |  |
| --- | --- |
| siam.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  20  21  22  23  24  25  26  27  28  29  30  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  68-76 | Untuk memasukan fungsi Scanner pada program  Membuat class siam  Membuat scanner  Membuat atribut ip dengan tipe double  Membuat atribut nama,nim,jurusan dalam tipe string  Membuat atribut sksMax, sksMasuk dan jmlMatkul dalam tipe int  Membuat atribut matkul dengan array bertipe string dan alokasi 10  Membuat atribut kodeMatkul dengan array bertipe string dan alokasi 10  Membuat atribut sksInput dengan array bertipe int dan alokasi 10  Membuat konstruktor siam dengan parameter: nama, nim, jurusan bertipe String dan ip bertipe double  Memberi nilai variable instance dengan this. Untuk menghindari variabel yang sama antara variabel class dengan variabel property  Membuat method jmlSKS() yang didalamnya terdapat percabangan untuk menentukan jumlah sks yang dapat diambil sesuai IP terakhir mahasiswa, Jika IP kurang dari 1.49  maka sks max yang dapat diambil adalah 12  Jika IP kurang dari 1.99  maka sks max yang dapat diambil adalah 15  Jika IP kurang dari 2.49  maka sks max yang dapat diambil adalah 18  Jika IP kurang dari 2.99  maka sks max yang dapat diambil adalah 21  Jika IP lebih dari 3.00  maka sks max yang dapat diambil adalah 24  Membuat method KRS  Mencetak kalimat “=============== KRS ==============”  Mencetak "Jumlah SKS yang bisa diambil : " dan nilai dari sksMax  Menccetak kalimat "Jumlah Mata kuliah yang diambil : "  Mendapat masukan nilai variable jmlMatkul dari user dengan tipe data integer  Inisialisasi nilai I sama dengan 0  Membuat perulangan menggunakan while, jika sksMasuk lebih kecil dari sksMax maka meminta pengguna memasukkan kode mata kuliah, nama mata kuliah dan jumlah SKS mata kuliah tersebut, Kemudian membuat sksMasuk += sksInput[i]; agar jumlah sksMasuk bertambah 1  Menambah nilai variable i satu nilai setiap dilakukan perulangan  Membuat percabangan jika (sksMasuk > sksMax) maka mencetak Maaf.. Jumlah SKS yang anda ambil melebihi jumlah maksimal  Mencetetak enter  Membuat sksMasuk += sksInput[i]; agar jumlah sksMasuk bertambah 1  Nilai I bertambah setiap perulangan  Jika (sksMasuk <= sksMax) maka mencetak >> Sisa SKS yang dapat anda ambil adalah: " + (sksMax - sksMasuk)  Jika nilai I sama dengan jumlah matkul,melakukan break  Membuat method display() yang didalamnya tredapat percabangan jika (sksMasuk > sksMax) maka Maaf.. KRS tidak dapat di tampilkan  Jika tidak maka menampilkan KRs berupa: nama mahasiswa, nim mahasiswa, nim mahasiswa  Membuat perulangan untuk menampilkan seluruh mata kuliah beserta kode dan jumlah sksnya serta menampilkan total sks yang diambil |

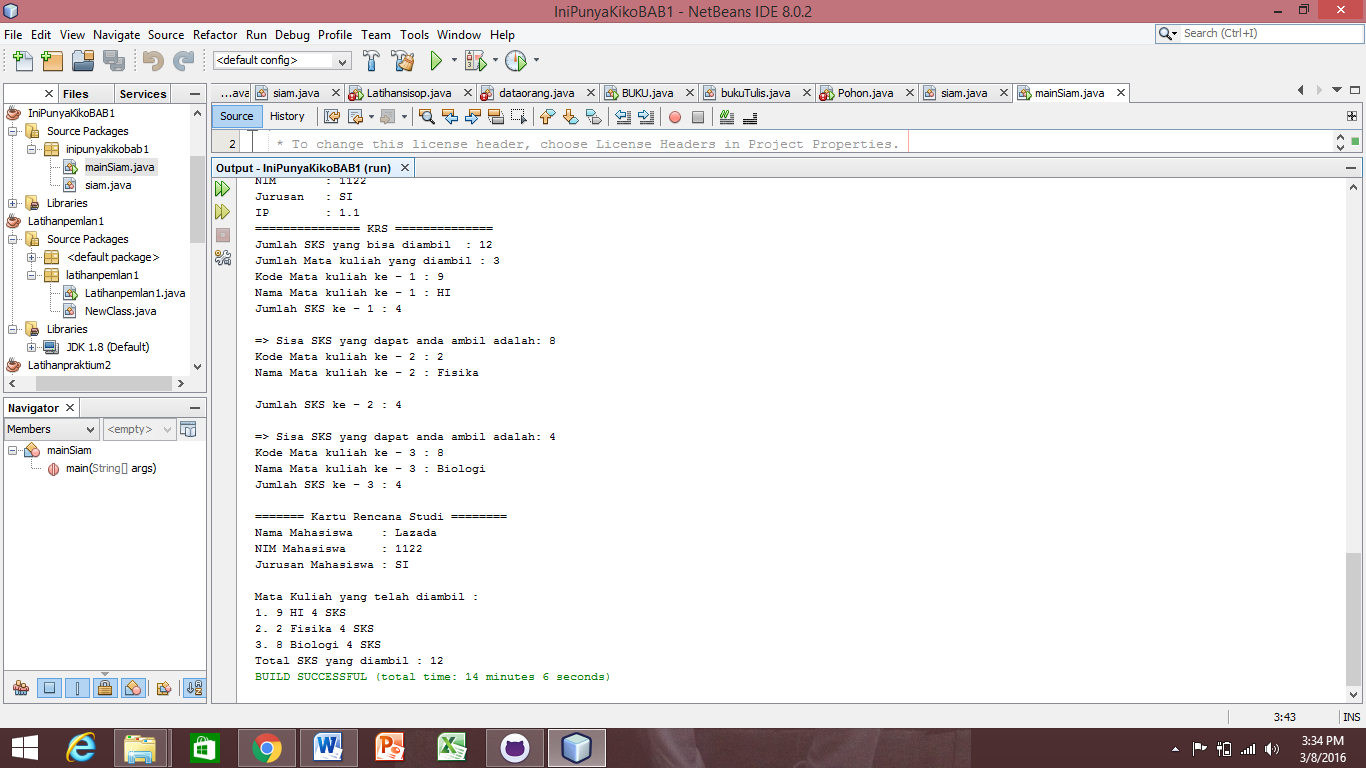
|  |  |
| --- | --- |
| mainSiam.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | Untuk memasukan fungsi Scanner pada program  Membuat kelas mainSiam untuk memanggil method dari kelas siam  fungsi main dimana public pada bagian ini berarti bahwa metode-metode dapat dipanggil didalam dan juga diluar kelas. static sendiri berarti sama untuk seluruh instant dari kelas dan void sendiri berarti bahwa metode tidak akan mengirim apapun setelah selesainya  Membuat fungsi scanner dengan nama input  Mencetak ====SIAM====  Mencetak Masukkan jumlah Mahasiswa :  Memasukan jumlah mahasiswa  Membuat atribut ip dengan tipe double  Membuat perulangan i=0 dan i lebih kecil dari jmlhmhs  Mencetak enter  Mencetak Identitas Mahasiswa  Mencetak “Mahasiswa ke-“ dan menampilkan nilai i  Mencetak “Nama :”  Memasukkan nama mahasiswa  Mencetak “NIM :”  Memasukkan NIM  Mencetak “Jurusan :”  Memasukkan Jurusan  Membuat perulangan dengan do  Mencetak “IP :”  Memasukkan jumlah IP mahasiswa  Membuat percabangan dengan if jika memasukkan ip > 4  Maka mencetak “IP tidak valid, Masukan kembali.”  Jika tidak maka menginstansiasi objek dengan nama objek s1  Ketika ip diatas 4  Membuat class baru bernama s1  Memanggil method jmlSKS  Memanggil method KRS  Memanggil method display |

1. **SCREENSHOT PROGRAM**









1. **PRAKTIKUM**

Pertanyaan

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

Instance Variable adalah variable yang ada di dalam class tetapi berada di luar method. Instance variable ini merupakan attribute dari suatu class dan mempunyai default value yang tidak perlu diinisialisasi.

Local Variabel adalah variabel yang dideklarasikan di dalam badan method. Jadi kita hanya dapat menggunakan variabel tersebut hanya di dalam method.

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

Tidak terdapat kesalahan saat menjalankan source code tersebut.

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

|  |  |
| --- | --- |
| MainMobil.java | |
| 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 | import java.util.Scanner;  public class mainmobil {  public static void main(String[] args) {    Scanner input = new Scanner(System.in);    String noPlat, warna, manufaktur;  int kecepatan;    mobil m1 = new mobil();  //instan objek bernama m1  System.out.println("Silahkan memasukkan identitas mobil anda");  System.out.print("No plat : ");  noPlat = input.next();  System.out.print("Manufaktur : ");  manufaktur = input.next();  System.out.print("Kecepatan : ");  kecepatan = input.nextInt();  System.out.print("warna : ");  warna = input.next();    m1.setNoPlat(noPlat);  m1.setManufaktur(manufaktur);  m1.setKecepatan(kecepatan);  m1.setWarna(warna);  m1.displayMessage();  System.out.println("================");    mobil m2 = new mobil();  //instan objek baru bernama m2  System.out.println("Silahkan memasukkan identitas mobil anda");  System.out.print("No plat : ");  noPlat = input.next();  System.out.print("Manufaktur : ");  manufaktur = input.next();  System.out.print("Kecepatan : ");  kecepatan = input.nextInt();  System.out.print("warna : ");  warna = input.next();    m2.setNoPlat(noPlat);  m2.setManufaktur(manufaktur);  m2.setKecepatan(kecepatan);  m2.setWarna(warna);  m2.displayMessage();  System.out.println("================");    //merubah warna dari objek m1  System.out.println("Mobil pertama yang dimasukkan identitasnya dirubah warnanya menjadi ");  m1.setWarna("Hijau");    //menampilkan hasil perubahan  m1.displayMessage();  }  } |

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)

|  |  |
| --- | --- |
| mobil.java | |
| 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 | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kecepatan;  private double jam;    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 jam){  this.jam = jam;  }  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("dalam waktu "+jam+" jam");  System.out.println("mobil mampu menempuh kecepatan "+kecepatan);  }  } |

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!

|  |  |
| --- | --- |
| mobil.java | |
| 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 | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kecepatan;  private double jam, detik;    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 jam){  this.jam = jam;  rubahSekon(jam);  }  private void rubahSekon(double jam){  detik = jam \* 3600;  }  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.print("dalam waktu "+jam+" jam");  System.out.println(" atau "+detik+" detik");  System.out.println("mobil mampu menempuh kecepatan "+kecepatan);  }  } |

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!

|  |  |
| --- | --- |
| mobil.java | |
| 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 | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kms;  private double jam, detik, ms;    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 kms){  this.kms = kms;  rubahKecepatan (kms);  }  public void setWaktu(double jam){  this.jam = jam;  rubahSekon(jam);  }  public void rubahSekon(double jam){  detik = jam \* 3600;  }  public void rubahKecepatan(int kms){  ms = (kms\*1000)/3600;  }  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.print("dalam waktu "+jam+" jam");  System.out.println(" atau "+detik+" detik");  System.out.print("mobil mampu menempuh kecepatan "+kms+" km/h");  System.out.println(" atau "+ms+" m/s");  }  } |

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!

|  |  |
| --- | --- |
| mobil.java | |
| 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 | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kms;  private double jam, detik, ms, km, m;    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 kms){  this.kms = kms;  rubahKecepatan (kms);  }  public void rubahKecepatan(int kms){  ms = (kms\*1000)/3600;  }  public void setWaktu(double jam){  this.jam = jam;  rubahSekon(jam);  }  public void rubahSekon(double jam){  detik = jam \* 3600;  }    public void hitungJarak(double km){  this.km = kms\*jam;  }  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.print("dalam waktu "+jam+" jam");  System.out.println(" atau "+detik+" detik");  System.out.print("mobil mampu menempuh kecepatan "+kms+" km/h");  System.out.println(" atau "+ms+" m/s");  System.out.println("jarak yang ditempuh adalah "+km+" km");  }  } |

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

|  |  |
| --- | --- |
| mobil.java | |
| 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 | public class mobil {  private String noPlat;  private String warna;  private String manufaktur;  private int kms;  private double jam, detik, ms, km, m;    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 kms){  this.kms = kms;  rubahKecepatan (kms);  }  public void rubahKecepatan(int kms){  ms = (kms\*1000)/3600;  }  public void setWaktu(double jam){  this.jam = jam;  rubahSekon(jam);  }  public void rubahSekon(double jam){  detik = jam \* 3600;  }    public void hitungJarak(double km){  this.km = kms\*jam;  rubahJarak(km);  }  public void rubahJarak(double km){  m = this.km\*1000;  }  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.print("dalam waktu "+jam+" jam");  System.out.println(" atau "+detik+" detik");  System.out.print("mobil mampu menempuh kecepatan "+kms+" km/h");  System.out.println(" atau "+ms+" m/s");  System.out.println("jarak yang ditempuh adalah "+km+" km");  System.out.println("atau "+m+" m");  }  } |

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.

|  |  |
| --- | --- |
| tulis.java | |
| 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 | public class tulis {  private double lembar, halaman, hari;    public void kertas (double lembar) {  this.lembar = lembar;  }  public void halaman (double halaman) {  this.halaman = halaman;  }  public void hari () {  hari = halaman / lembar;  }    public void tampilkan () {  System.out.println("Jumlah seluruh halaman buku adalah " + halaman + " halaman");  System.out.println("Setiap hari mahasiswa dapat menulis " + lembar + " lembar");  System.out.println("Mahasiswa dapat menulis selama " + hari + " hari" );    }    } |

|  |  |
| --- | --- |
| tulisBeraksi.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | public class tulisBeraksi {  public static void main(String[] args) {    tulis tt = new tulis();  tt.halaman(50);  tt.kertas(0.5);  tt.hari();  tt.tampilkan();    }  } |

1. **KESIMPULAN**

(OOP) merupakan suatu pendekatan pemrograman yang menggunakan object dan class. Dalam OOP, setiap bagian dari program adalah object. Sebuah object mewakili suatu bagian program yang akan diselesaikan. Program dalam OOP membungkus (encapsulate) data dan fungsi (atau prosedur) dalam suatu obyek yang umumnya diimplementasikan sebagai suatu kelas (class).

1. CLASS

Kelas pada Java didefenisikan dengan menggunakan kata kunci class. Berikut ini merupakan deklarasi kelas pada Java : {modifier1} class Namakelas [modifier2] { Class body;}

1. METHOD

Method digunakan untuk memodularisasi program pemisah satu tugas menjadi satu unit tersendiri. Berikut ini meupakan deklarasi method pada Java : [modifier1] tipe NilaiKembalian namaMethod (parameter input) { Method body; }

1. CONSTRUCTOR

Constructor merupakan method yang namanya sama dengan nama kelas dimana method itu berada. Constructor digunakan untuk menginisialisasi nilai awal (memberikan nilai default) pada atribut – atribut yang dimiliki oleh suatu objek saat objek itu pertama kali dibuat.

1. MODIFIER

Modifier adalah sifat yang dimiliki oleh setiap atribut, method maupun kelas dalam Java . Modifier akses adalah modifier yang selalu digunakan, modifier akses terdiri dari private, default, protected, dan public.

Ada beberapa modifier penting lainnya, yaitu:

1) Static yaitu modifier yang digunakan agar suatu atribut maupun method dapat diakses oleh Objek atau kelas meski tanpa dilakukan instansi terhadap kelas dimana atribut maupun method itu berada, method main adalah salah satu contoh method modifier static yang sering digunakan.

2) Final yaitu modifier yang digunakan untuk mencegah kemungkinan modofikasi terhadap atribut maupun method, dengan modifier ini suatu atribut akan berlaku sebagai konstanta.

1. OBJEK

Sebuah kelas dapat digunakan untuk membuat banyak objek, setiap objek dapat diperlakukan secara berbeda oleh objek – objek lain yang menggunakannya. Dibutuhkan operatir new untuk membuat objek dari suatu kelas.

Objek dibuat dengan cara menuliskan :

1) NamaKelas NamaObjek = new NamaConstructor();

2) NamaKels NamaObjek;

NamaObjek = new NamaConstructor();