1. **DEFINISI MASALAH**
2. Sebelummengerjakan soal dibawah initentukan dahulu objek, attribut, behaviour dan class.

Buatlah sebuah sistem sederhana yang menyerupai Sistem Informasi Akademik

Mahasiswa (SIAM), dengan ketentuan user menginputkan Nama, Nim, IP sertajurusan. Selain itu, mahasiswa juga dapat memasukkan kode Matakuliah, Nama Matakuliah dan jumlah sks matakuliah tersebut. Jumlah sks yang diambil harus sesuai dengan IP yang didapat pada semester lalu. Buat skenario dengan banyak mahasiswa minimal 3 orang.

|  |  |
| --- | --- |
| **Mahasiswa** (objek) | **MataKuliah** (objek) |
| Attribute : | Attribute : |
| * NIM * Nama * IP * Jurusan * JumlahSKS | * Kode\_Matkul * Nama\_Matkul * SKS |
| Behavior : | Behavior : |
| * setNIM * setNama * setIP * setJurusan * setJumlahSKS * getNIM * getNama * getIP * getJurusan * getJumlahSKS | * setKode\_Matkul * setNama\_Matkul * setSKS * getKode\_Matkul * getNama\_Matkul * getSKS |

Class : Mahasiswa, MataKuliah, Main.

1. **SOURCE CODE**

|  |  |
| --- | --- |
|  | **Mahasiswa.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 | import java.util.Scanner;  public class Mahasiswa {  Scanner scan = new Scanner(System.in);  private String Nama;  private String NIM;  private double IP;  private String Jurusan;  private int JumlahSKS;  public void setNama() {  System.out.print("Nama\t: ");  this.Nama = scan.nextLine();  }  public void setNIM() {  System.out.print("NIM\t: ");  this.NIM = scan.nextLine();  }  public void setIP() {  System.out.print("IP\t: ");  this.IP = scan.nextDouble();  }  public void setJurusan() {  System.out.print("Jurusan\t: ");  this.Jurusan = scan.nextLine();  }  public String getNama() {  return this.Nama;  }  public String getNIM() {  return this.NIM;  }  public double getIP() {  return this.IP;  }  public String getJurusan() {  return this.Jurusan;  }  public void setJumlahSKS(int sks){  this.JumlahSKS=this.JumlahSKS+sks;  }  public int getJumlahSKS() {  return this.JumlahSKS;  }  } |

|  |  |
| --- | --- |
|  | **MataKuliah.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 | import java.util.Scanner;  public class MataKuliah {  Scanner scan = new Scanner(System.in);  private String Kode\_Matkul;  private String Nama\_Matkul;  private int SKS;  public void setKode\_Matkul() {  System.out.print("Kode Mata Kuliah\t: ");  this.Kode\_Matkul = scan.nextLine();  }  public void setNama\_Matkul() {  System.out.print("Nama Mata Kuliah\t: ");  this.Nama\_Matkul = scan.nextLine();  }  public void setSKS() {  System.out.print("SKS yang diambil\t: ");  this.SKS = scan.nextInt();  }  public String getKode\_Matkul() {  return Kode\_Matkul;  }  public String getNama\_Matkul() {  return Nama\_Matkul;  }  public int getSKS() {  return SKS;  }  } |

|  |  |
| --- | --- |
|  | **Main.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  85  86  87 | import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  double[][] sks = {{3, 22, 24}, {2.5, 19, 21}, {2, 16, 18},  {1.5, 12, 15}, {0, 0, 11}};  System.out.println("PROGRAM SISTEM INFORMASI AKADEMIK");  System.out.print("Masukkan jumlah mahasiswa : ");  Mahasiswa[] mahasiswa = new Mahasiswa[scan.nextInt()];  MataKuliah[][] matakuliah = new MataKuliah[25][25];  String loop = "n";  int i;  if (mahasiswa.length >= 3) {  for (int x = 0; x < mahasiswa.length; x++) {  System.out.println("");  System.out.println("Mahasiswa " + (x + 1));  mahasiswa[x] = new Mahasiswa();  mahasiswa[x].setNama();  mahasiswa[x].setNIM();  mahasiswa[x].setJurusan();  mahasiswa[x].setIP();  i = -1;  do {  i++;  System.out.println("\nMatakuliah "+(i+1));  matakuliah[x][i] = new MataKuliah();  matakuliah[x][i].setKode\_Matkul();  matakuliah[x][i].setNama\_Matkul();  matakuliah[x][i].setSKS();  mahasiswa[x].setJumlahSKS(matakuliah[x][i].  getSKS());  System.out.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_");  loop = "y";  for (int y = 0; y < sks.length; y++) {  if (mahasiswa[x].getIP() >= sks[y][0]) {  if (mahasiswa[x].getJumlahSKS() < sks[y][1]) {  System.out.println("Kurang " + (sks[y][1] –  mahasiswa[x].getJumlahSKS()) + " SKS");  loop = "n";}  else if (mahasiswa[x].getJumlahSKS() >  sks[y][2]) {  System.out.println("Kelebihan " +  (mahasiswa[x].getJumlahSKS() - sks[y][2]) +  " SKS");  mahasiswa[x].setJumlahSKS(0 - mahasiswa[x]  .getJumlahSKS());  loop = "n";}  break;  }  }  if (loop == "y") {  System.out.print("Sudah?(y/n)");  loop = scan.next();  }  } while (loop.equals("n"));  }  for (int x = 0; x < mahasiswa.length; x++) {  System.out.println("\nMAHASISWA "+(x+1));  System.out.println("NIM\t: "+mahasiswa[x].getNIM());  System.out.println("Nama\t: "+mahasiswa[x].getNama()  + " ");  System.out.println("IP\t: "+mahasiswa[x].getIP()  + " ");  System.out.println("Jurusan\t: "+mahasiswa[x].  GetJurusan() + " ");  System.out.println("SKS\t: "+mahasiswa[x].  getJumlahSKS() + " ");  System.out.println("");  System.out.println("MATAKULIAH");  System.out.println("Kode\tNama\tSKS");  for (int y = 0; y < matakuliah[x].length; y++) {  if(matakuliah[x][y] == null) break;  System.out.print( matakuliah[x][y].  getKode\_Matkul());  System.out.print("\t"+matakuliah[x][y].  getNama\_Matkul());  System.out.print(" "+matakuliah[x][y].getSKS());  System.out.println("");  }  System.out.println("------------------");  System.out.println("TOTAL\t\t"+mahasiswa[x].  getJumlahSKS()+" SKS");  }  } else {  System.out.println("Mahasiswa minimal 3!");  }  }  } |

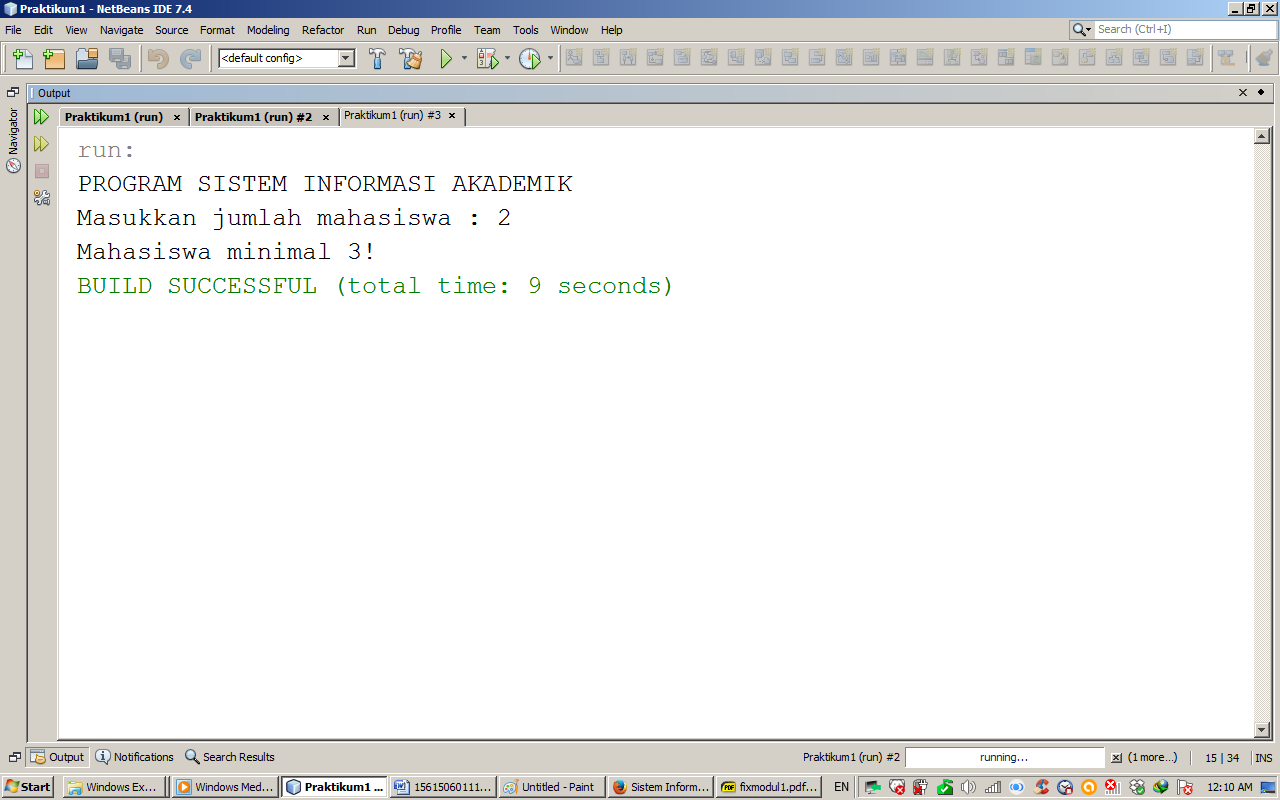
1. **PEMBAHASAN**

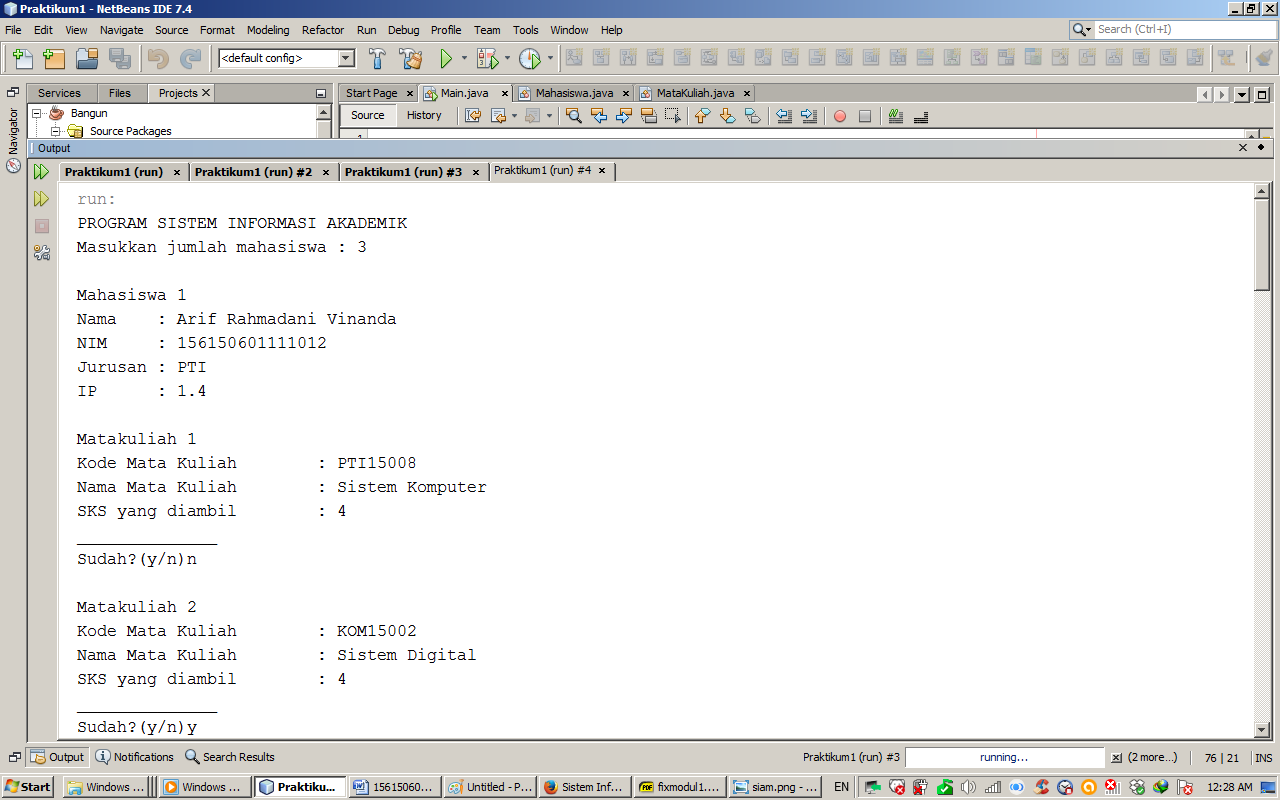
|  |  |
| --- | --- |
|  | **Mahasiswa.java** |
| 1  2  3  4  5  6  7  8  9  10  11  13  14  15  17  18  19  21  22  23  25  26  28  29  31  32  34  35  37  38  40  41 | Memanggil fungsi Scanner java.  Deklarasi class dengan nama Mahasiswa.  Inisialisasi fungsi Scanner baru bernama scan.  Inisialisasi variable Nama sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable NIM sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable IP sebagai atribut bertipe data double dengan access modifier private.  Inisialisasi variable Jurusan sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable JumlahSKS sebagai atribut bertipe data integer dengan access modifier private.  Pendeklarasian method setNama tanpa ada pengembalian data (void) dengan access modifier public.  Proses mencetak “Nama : ”.  Proses input kedalam variable Nama bertipe data String pada class Mahasiswa.  Pendeklarasian method setNIM tanpa ada pengembalian data (void) dengan access modifier public.  Proses mencetak “NIM : ”.  Proses input kedalam variable NIM bertipe data String pada class Mahasiswa.  Pendeklarasian method setIP tanpa ada pengembalian data (void) dengan access modifier public.  Proses mencetak “IP : ”.  Proses input kedalam variable IP bertipe data double pada class Mahasiswa.  Pendeklarasian method setJurusan tanpa ada pengembalian data (void) dengan access modifier public.  Proses mencetak “Jurusan : ”.  Proses input kedalam variable Jurusan bertipe data String pada class Mahasiswa.  Pendeklarasian method getNama dengan pengembalian data bertipe String dari variable Nama class Mahasiswa dengan access modifier public.  Proses pengembalian data method getNama dari variable Nama.  Pendeklarasian method getNIM dengan pengembalian data bertipe String dari variable NIM class Mahasiswa dengan access modifier public.  Proses pengembalian data method getNIM dari variable NIM.  Pendeklarasian method getIP dengan pengembalian data bertipe double dari variable IP class Mahasiswa dengan access modifier public.  Proses pengembalian data method getIP dari variable IP.  Pendeklarasian method getJurusan dengan pengembalian data bertipe String dari variable Jurusan class Mahasiswa dengan access modifier public.  Proses pengembalian data method getJurusan dari variable Jurusan.  Pendeklarasian method setJumlahSKS tanpa ada pengembalian data (void) dengan access modifier public dan berparameter variable sks bertipe integer.  Proses penghitungan variable JumlahSKS ditambah nilai variable sks dan disimpan pada variable JumlahSKS.  Pendeklarasian method getJumlahSKS dengan pengembalian data bertipe integer dari variable JumlahSKS class Mahasiswa dengan access modifier public.  Proses pengembalian data method getJumlahSKS dari variable JumlahSKS. |

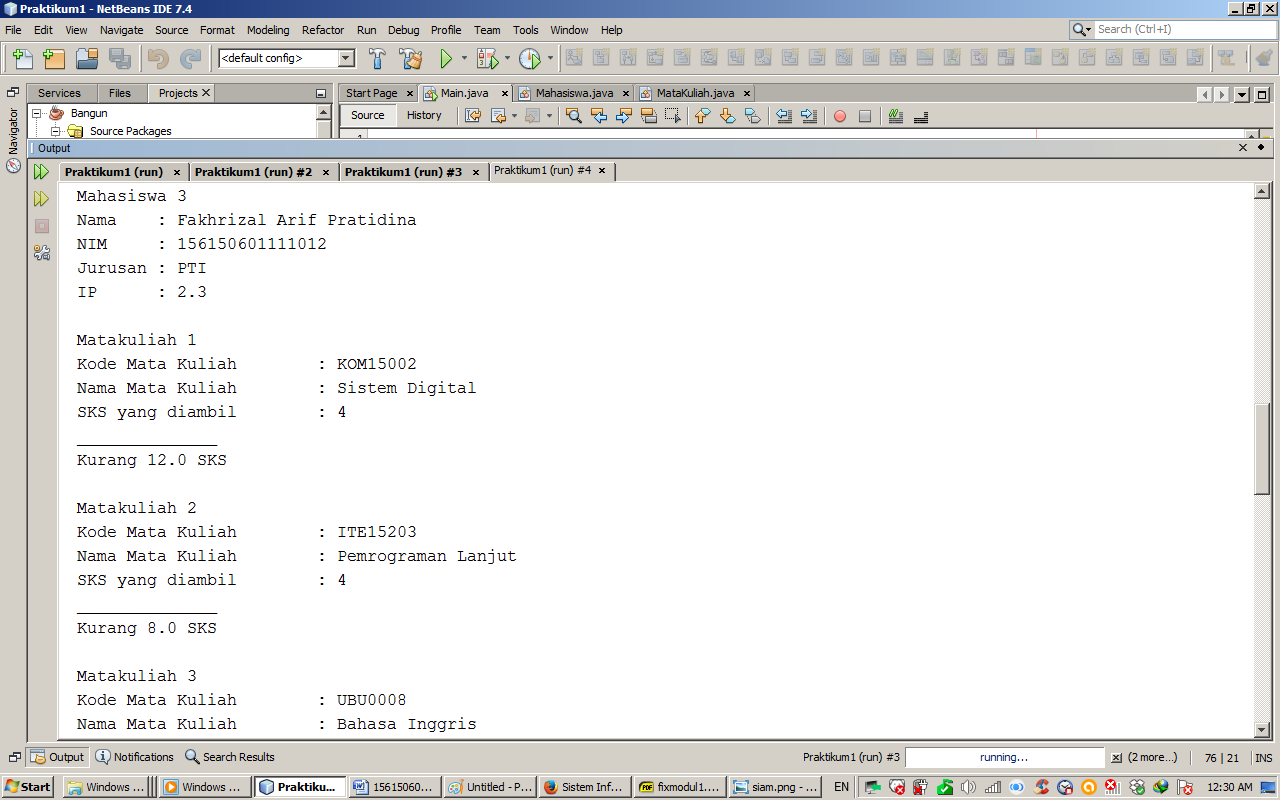
|  |  |
| --- | --- |
|  | **MataKuliah.java** |
| 1  2  3  4  5  6  7  8  9  11  12  14  15  16  17  19  20  22  23  25  26 | Memanggil fungsi Scanner java.  Deklarasi class dengan nama MataKuliah.  Inisialisasi fungsi Scanner baru bernama scan.  Inisialisasi variable Kode\_Matkul sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable Nama\_Matkul sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable SKS sebagai atribut bertipe data integer dengan access modifier private.  Pendeklarasian method setKode\_Matkul tanpa ada pengembalian data (void) dengan access modifier public.  Proses mencetak “Kode Mata Kuliah : ”.  Proses input kedalam variable Kode\_Matkul bertipe data String pada class MataKuliah.  Pendeklarasian method setNama\_Matkul tanpa ada pengembalian data (void) dengan access modifier public.  Proses mencetak “Nama Mata Kuliah : ”.  Proses input kedalam variable Nama\_Matkul bertipe data String pada class MataKuliah.  Pendeklarasian method setSKS tanpa ada pengembalian data (void) dengan access modifier public.  Proses mencetak “SKS yang diambil : ”.  Proses input kedalam variable SKS bertipe data integer pada class MataKuliah.  Pendeklarasian method getKode\_Matkul dengan pengembalian data bertipe String dari variable Kode\_Matkul class Mahasiswa dengan access modifier public.  Proses pengembalian data method getKode\_Matkul dari variable Kode\_Matkul.  Pendeklarasian method getNama\_Matkul dengan pengembalian data bertipe String dari variable Nama\_Matkul class Mahasiswa dengan access modifier public.  Proses pengembalian data method getNama\_Matkul dari variable Nama\_Matkul.  Pendeklarasian method getSKSdengan pengembalian data bertipe integer dari variable SKS class Mahasiswa dengan access modifier public.  Proses pengembalian data method getSKSdari variable SKS. |

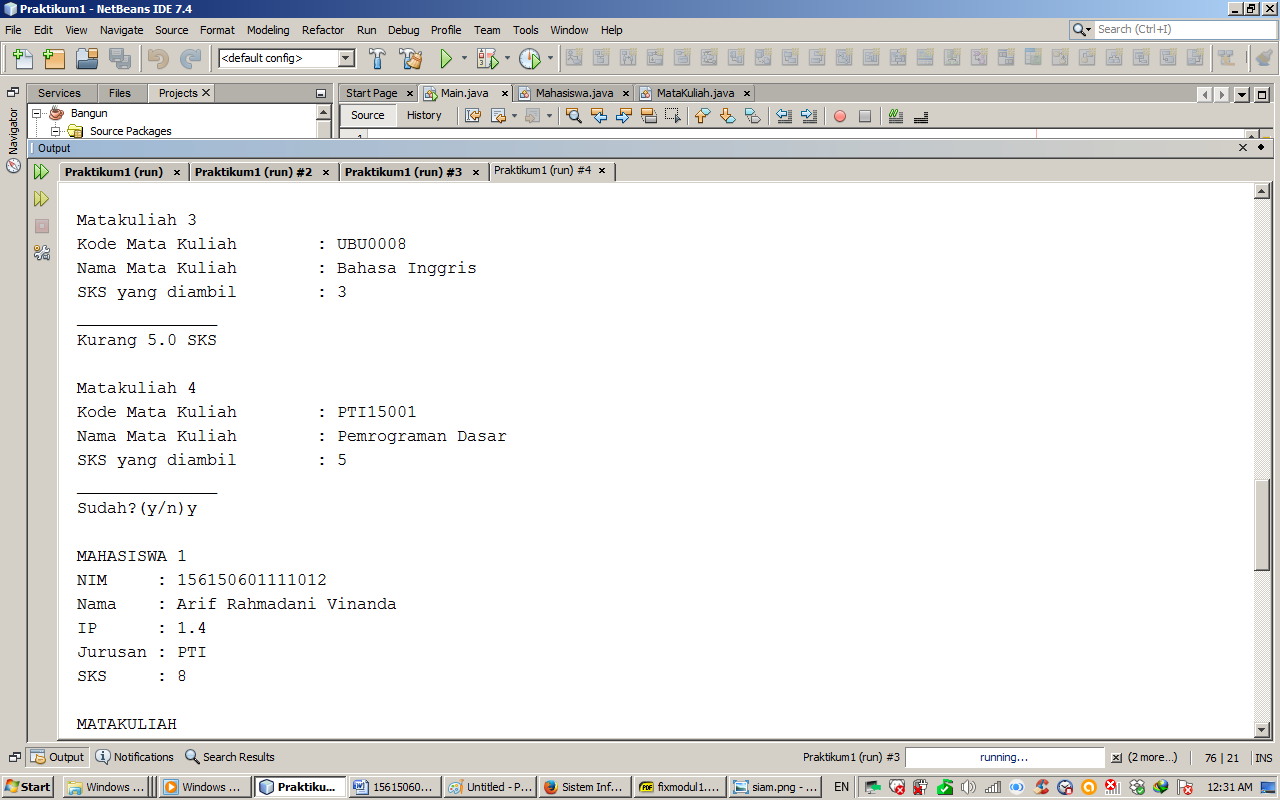
|  |  |
| --- | --- |
|  | **Main.java** |
| 1  2  3  4  5  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  32  33  34  35  36  37  39  40  42  45  47  48  51  52  53  55  57  58  59  60  62  64  66  68  69  70  71  72  73  75  77  78  80  81  85  86 | Memanggil fungsi Scanner java.  Deklarasi class dengan nama Main  Main method java.  Inisialisasi fungsi Scanner baru bernama scan.  Deklarasi array dua dimensi bernama sks dengan nilai {3,22,24},{2.5,19,21} {2,16,18},{1.5,12,15},{0,0,11}.  Proses mencetak “PROGRAM SISTEM INFORMASI AKADEMIK”.  Proses mencetak “Masukkan jumlah mahasiswa : ”.  Inisialisasi objek mahasiswa sebagai array satu dimensi dengan pajang indeks array sesuai inputan user dengan tipe data integer.  Inisialisasi objek matakuliah sebagai array dua dimensi dengan panjang maksimal masing-masing indeks 25.  Inisialisasi variable loop bertipe data String bernilai “n”.  Inisialisasi variable i bertipe data integer.  Pengecekan kondisi apakah panjang dari indeks array mahasiswa lebih dari sama dengan 3. Jika benar akan lanjut ke baris selanjutnya.  Inisialisasi perulangan for dengan variable x bertipe data integer bernilai 0. Akan terus berulang hingga nilai kurang dari panjang array mahasiswa serta nilai variable x akan terus bertambah 1 setiap perulangan.  Proses mencetak “ ” (enter).  Proses mencetak “Mahasiswa ” diikuti nilai variable x dari perulangan ditambah 1.  Inisialisasi array mahasiswa dengan indeks ke x.  Proses pemanggilan method setNama dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses pemanggilan method setNIM dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses pemanggilan method setJurusan dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses pemanggilan method setIP dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses pengisian variable i denga nilai -1.  Proses dimulainya looping do-while.  Penambahan variable i dengan angka 1.  Proses mencetak “Matakuliah ” diikuti nilai dari variable i ditambah dengan 1.  Inisialisasi array matakuliah indeks ke x,i.  Proses pemanggilan method setKode\_Matkul dari class MataKuliah pada array matakuliah indeks ke x,i.  Proses pemanggilan method setNama\_Matkul dari class MataKuliah pada array matakuliah indeks ke x,i.  Proses pemanggilan method setSKS dari class MataKuliah pada array matakuliah indeks ke x,i.  Proses pemanggilan method setJumlahSKS dari class Mahasiswa padai array mahasiswa indeks ke x berargumen nilai yang didapat dari method getSKS dari array matakuliah indeks ke x,i.  Proses mencetak “\_\_\_\_\_\_\_\_\_\_\_\_\_\_”.  Proses mengisi variable loop dengan nilai “y”.  Inisialisasi perulangan for dengan variable y bertipe data integer bernilai 0. Akan terus berulang hingga nilai kurang dari panjang array sks serta nilai variable y akan terus bertambah 1 setiap perulangan.  Pengecekan kondisi apakah nilai dari method getIP dari class Mahasiswa pada array mahasiswa indeks ke x lebih dari sama dengan nilai pada array sks indeks ke y,0.  Pengecekan kondisi apakah nilai dari method getJumlahSKS dari class Mahasiswa pada array mahasiswa indeks ke x kurang dari nilai pada array sks indeks ke y,1.  Proses mencetak “Kurang ” diikuti nilai dari array sks indeks ke y,1 dikurangi nilai dari method getJumlahSKS dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses mengisi variable loop dengan nilai “n”.  Pengecekan kondisi apakah nilai dari method getJumlahSKS dari class Mahasiswa pada array mahasiswa indeks ke x lebih dari nilai pada array sks indeks ke y,2.  Proses mencetak “Kelebihan ” diikuti nilai dari method getJumlahSKS dari class Mahasiswa pada array mahasiswa indeks ke x dikurangi nilai dari array sks indeks ke y,2.  Proses pemanggilan method setJumlahSKS dari class Mahasiswa pada array mahasiswa indeks ke x dengan argumen nilai 0 dikurangi nilai dari method getJumlahSKS dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses mengisi variable loop dengan nilai “n”.  Fungsi untuk menghentikan atau keluar dari perulangan for diatas.  Pengecekan kondisi apakah nilai dari variable loop adalah “y”.  Proses mencetak “Sudah?(y/n)”.  Proses input kedalam variable loop bertipe data String.  Statement yang menunjukkan loop do-while akan berjalan ketika nilai dari variable loop bernilai “n”.  Inisialisasi perulangan for dengan variable x bertipe data integer bernilai 0. Akan terus berulang hingga nilai kurang dari panjang array mahasiswa, serta nilai variable x akan terus bertambah 1 setiap perulangan.  Proses mencetak “MAHASISWA ” diikuti nilai variable x dari perulangan ditambah 1.  Proses mencetak “NIM : ” diikuti nilai dari method getNIM dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses mencetak “Nama : ” diikuti nilai dari method getNama dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses mencetak “IP : ” diikuti nilai dari method getIP dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses mencetak “Jurusan : ” diikuti nilai dari method getJurusan dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses mencetak “SKS : ” diikuti nilai dari method getJumlahSKS dari class Mahasiswa pada array mahasiswa indeks ke x.  Proses mencetak “ ” (enter).  Proses mencetak “MATAKULIAH”  Proses mencetak “Kode Nama SKS”.  Inisialisasi perulangan for dengan variable y bertipe data integer bernilai 0. Akan terus berulang hingga nilai kurang dari panjang array matakuliah serta nilai variable y akan terus bertambah 1 setiap perulangan.  Pengecekan kondisi apakah nilai dari array matakuliah indeks ke x,y bernilai kosong. Jika benar maka akan melakukan fungsi untuk menghentikan atau keluar dari perulangan for.  Proses mencetak method getKode\_Matkul dari class MataKuliah pada array matakuliah indeks ke x,y.  Proses mencetak method getNama\_Matkul dari class MataKuliah pada array matakuliah indeks ke x,y.  Proses mencetak method getSKS dari class MataKuliah pada array matakuliah indeks ke x,y.  Proses mencetak “ ” (enter).  Proses mencetak “-------------------------“.  Proses mencetak “TOTAL ” diikuti nilai dari method getJumlahSKS dari class Mahasiswa pada array mahasiswa indeks ke x.  Fungsi yang dijalankan apabila nilai dari if tidak terpenuhi.  Proses mencetak “Mahasiswa minimal 3!”. |

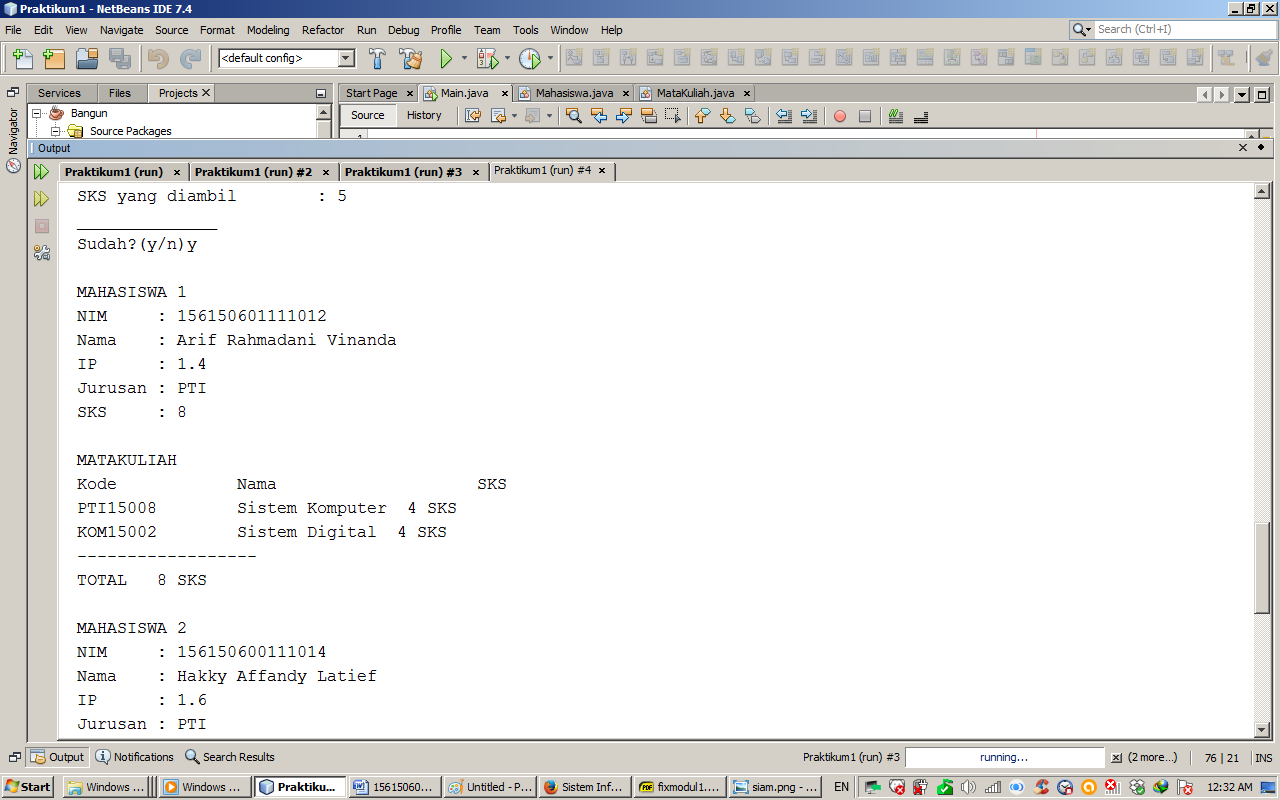
1. **SCREENSHOT PROGRAM**

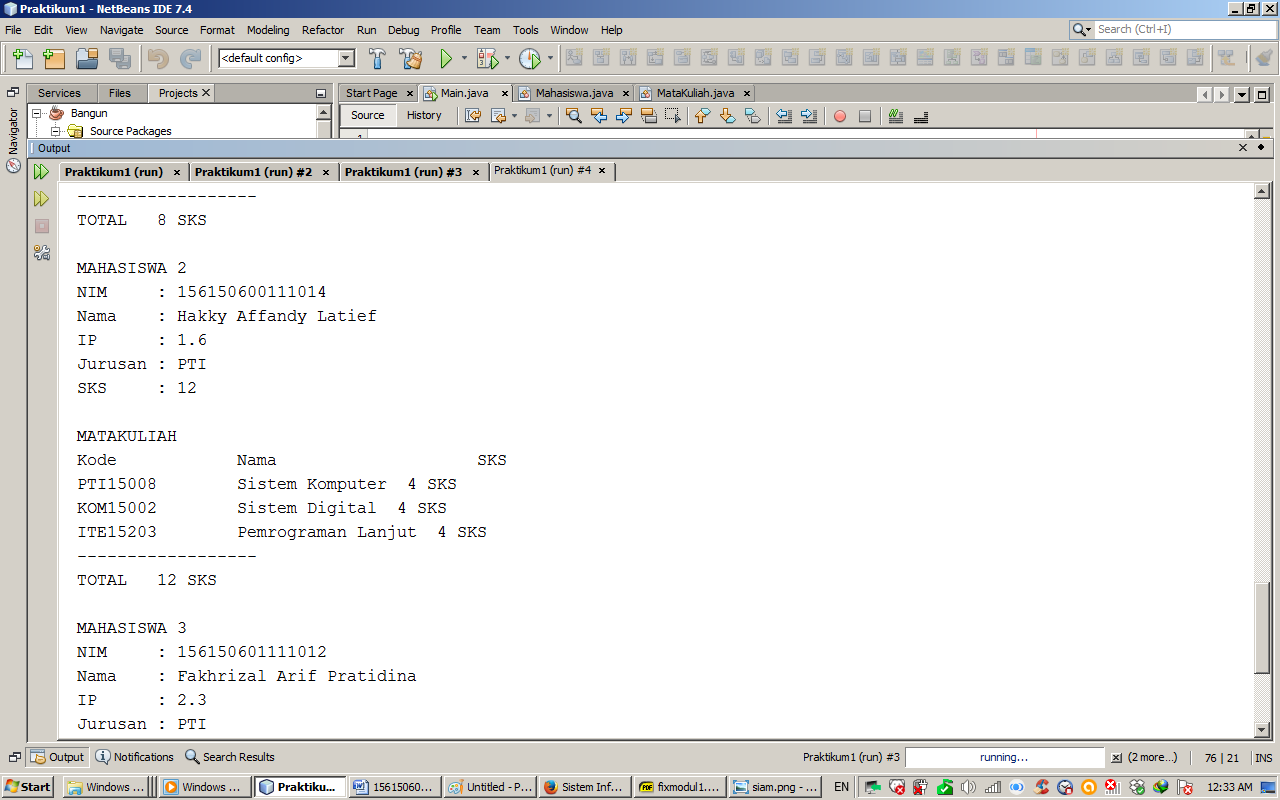
****

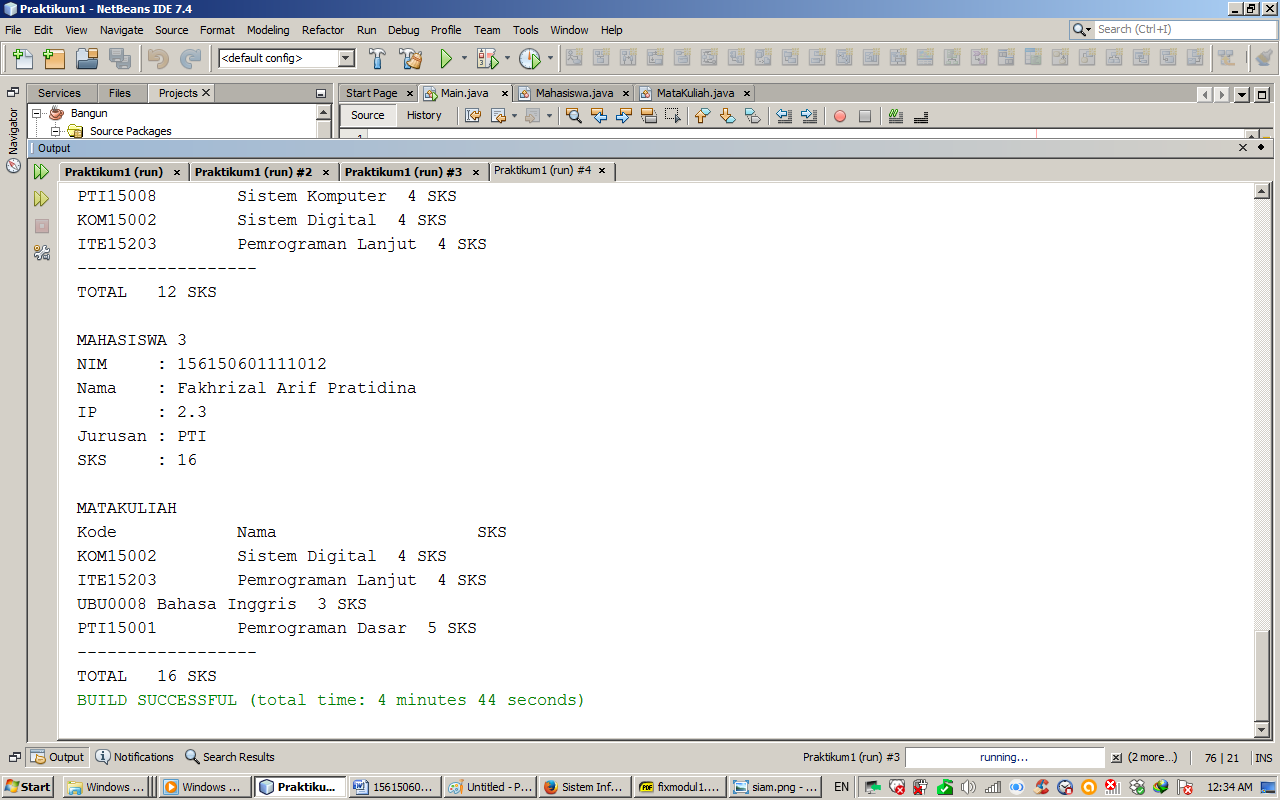
****

****

****

****

****

****

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

|  |  |
| --- | --- |
| Jawaban : | Instance variable adalah variable yang di deklarasikan di dalam kelas tetapi diluar method, konstruktor maupun blok statement dan dapat digunakannya access modifier seperti *public, private, protected* untuk digunakan pada method lain. Sedangkan variable lokal adalah variable yang di deklarasikan di dalam method, konstruktor ataupun blok statement yang hanya digunakan dalam method tersebut. Dalam penggunaan lokal variable dilarang memberikan access modifier pada lokal variable. |

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

|  |  |
| --- | --- |
| Jawaban : | Tidak ada kesalahan dalam source code program. |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Jawaban : | |  |  | | --- | --- | |  | **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 | import java.util.Scanner;  public class MainMobil {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  Mobil m1 = new Mobil();  System.out.println("Masukkan Mobil 1");  System.out.print("Kecepatan mobil : ");  m1.setKecepatan(scan.nextInt());  System.out.print("Merk mobil : ");  m1.setManufaktur(scan.next());  System.out.print("Plat nomor : ");  m1.setNoPlat(scan.next());  System.out.print("Warna : ");  m1.setWarna(scan.next());  System.out.println("");  m1.displayMessage();  System.out.println("================");  Mobil m2 = new Mobil();  System.out.println("Masukkan Mobil 2");  System.out.print("Kecepatan mobil : ");  m2.setKecepatan(scan.nextInt());  System.out.print("Merk mobil : ");  m2.setManufaktur(scan.next());  System.out.print("Plat nomor : ");  m2.setNoPlat(scan.next());  System.out.print("Warna : ");  m2.setWarna(scan.next());  System.out.println("");  m2.displayMessage();  System.out.println("================");  System.out.println("mobil pada objek m1 di  rubah menjadi warna ? ");  m1.setWarna(scan.next());  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)!

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Jawaban : | |  |  | | --- | --- | | Tambahkan source code dibawah pada file Mobil.java | | |  | **Mobil.java** | | 1  2  3  4 | private double waktu;  public void setWaktu(double waktu){  this.waktu=waktu;  } | |
|  | |  |  | | --- | --- | | Lalu tambahkan juga source code dibawah pada MainMobil.java untuk memberi keterangan pada user sekaligus input. | | |  | **MainMobil.java** | | 1  2 | System.out.print("Masukkan waktu (jam) : ");  m1.setWaktu(scan.nextDouble()); | |

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 dipanggil pada method setWaktu dengan nilai parameter adalah nilai dari variabel parameter method setWaktu!

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Jawaban : | |  |  | | --- | --- | |  | **Mobil.java** | | 1  2  3  4  5  6  7 | public void setWaktu(double waktu){  this.waktu=rubahSekon(waktu);  }  private double rubahSekon(double waktu){  waktu=waktu\*3600;  return waktu;  } | |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Jawaban : | |  |  | | --- | --- | |  | **Mobil.java** | | 1  2  3  4  5  6  7 | public void setKecepatan(int i) {  kecepatan = rubahKecepatan(i);  }  private int rubahKecepatan(int i){  i=i\*1000;  return i;  } | |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Jawaban : | |  |  | | --- | --- | |  | **Mobil.java** | | 1  2  3  4 | public double hitungJarak(){  double jarak=this.kecepatan\*this.waktu;  return jarak;  } | |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Jawaban : | |  |  | | --- | --- | |  | **Mobil.java** | | 1  2  3  4  5  6  7  8  9  10  11  12 | public void displayMessage() {  System.out.println("Mobil anda bermerek " +  manufaktur);  System.out.println("mempunyai nomor plat " +  noPlat);  System.out.println("serta memililki warna "  + warna);  System.out.println("mampu menempuh  kecepatan " + kecepatan +" m/s");  System.out.println("dan dapat menempuh jarak  "+ (hitungJarak()/1000) +" 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.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Jawaban : | **RUMUS**  Waktu = (lembar x 2) / (kata / 200)  = (50 x 2) / (100 / 200)  = 100 / 0.5 = 200 hari  **IDENTIFIKASI**  Buku (Objek)  Atribut :   * Lembar   Behavior :   * Habis   Mahasiswa (Objek)  Atribut :   * KataPerHari   Behavior :   * Menulis   **IMPLEMENTASI**   |  |  | | --- | --- | |  | **Buku.java** | | 1  2  3  4  5  6  7  8  9  10  11  12  13 | public class Buku {  private int lembar;  public void setLembar(int lembar) {  this.lembar = lembar;  }  public int getLembar() {  return lembar;  }  public double getHabis(double kata){  double waktu = (this.lembar\*2)/(kata/200);  return waktu;  }  } |  |  |  | | --- | --- | |  | **Mahasiswa.java** | | 1  2  3  4  5  6  7  8  9  10  11  12  13 | public class Mahasiswa {  public static void main(String[] args) {  Buku buku = new Buku();  //Jumlah lembar buku  buku.setLembar(50);  //Kata yang dapat ditulis  double kata=100;  //Cetak waktu buku yang habis ditulis  System.out.println("Buku dengan "+  buku.getLembar()+" lembar akan habis  ditulis "+buku.getHabis(kata)+" hari");  }  } | |

1. **KESIMPULAN**

OOP adalah sebuah konsep/cara pemrograman dengan menggunakan objek sebagai elemen dasar dari suatu program. Jadi, setiap bagian dari suatu permasalahan adalah objek. Objek itu sendiri bisa merupakan gabungan dari beberapa objek yang lebih kecil lagi. Setiap objek memiliki dua karakteristik yaitu *Attribute* (*State*) dan *Behavior*. *Attribute* (*State*) merupakan identitas atau informasi objek itu sendiri atau disebut juga sebagai *variable*, sedangkan *Behavior* adalah tingkah laku atau apa yang dapat dilakukan oleh objek itu atau disebut juga sebagai *method*. Sebagai contoh *Attribute* (*State*) dari mobil adalah merk, tipe, warna, tahun produksi, dll. Sedangkan *Behavior* dari mobil adalah cara berjalan, belok, rem, dll. Selain itu, objek dapat memiliki relasi terhadap objek lainnya. Sebagai contoh, sebuah objek dapat berupa keturunan dari objek lain yang saling berhubungan/berelasi.

Dalam pembuatan objek di java, objek berasal dari satu class sebagai objek, field sebagai atribut dan method sebagai behavior objek tersebut. Contoh :

|  |  |
| --- | --- |
|  | **Persegi.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 | import java.util.Scanner;  //Objek bangun Persegi  public class Persegi {  Scanner scan = new Scanner(System.in);  //Atribut bangun Persegi  private double sisi;  //Behavior bangun Persegi  public double getSisi() {  return this.sisi;  }  public void setSisi() {  double sisi;  System.out.print("Masukkan sisi : ");  sisi = scan.nextDouble();  this.sisi = sisi;  }  public double getLuas() {  double luas = this.sisi \* this.sisi;  return luas;  }  } |

Seperti contoh diatas, yakni objek Persegi memiliki atribut sisi dengan behavior menetapkan dan mengambil nilai sisi serta dapat mencari luas persegi. Untuk pendeklarasian atau pembuatan objek dalam fungsi main dapat dilakukan seperti contoh dibawah :

|  |  |
| --- | --- |
|  | **Main.java** |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | public class Persegi {  public static void main(String[]args){  //Pendeklarasian objek persegi  Persegi persegi = new Persegi();  //Pemanggilan behavior/method dari objek persegi  persegi.setSisi();  //Mencetak luas persegi  System.out.println(persegi.getLuas());  }  } |

Dalam pembuatan objek menggunakan code seperti diatas untuk inisilisasi objek “Persegi” menjadi bernama “persegi” yang digunakan pada main class. Begitu pula dalam pemanggilannya, harus mencantumkan inisial dari objek lalu disertai method dari objek yang bersangkutan.

*Class* merupakan struktur dari code program (khususnya OOP) dalam java. Class dapat juga diartikan model yang berisi kumpulan attribute dan method dalam suatu unit yang nantinya digunakan sebagai template atau cetakan dari sebuah objek. Objek adalah kesatuan entitas(benda) yang merupakan representasi nyata dari sebuah kelas. Di dalam objek terdapat dua elemen penting yakni atribut dan behavior. Atribut merupakan nilai data  yang  terdapat pada  suatu object  yang berasal dari class. Atribut merepresentasikan karakteristik dari suatu object. Atau Atribut adalah sesuatu yang dimiliki oleh objek.  Setiap objek yang dibuat dari kelas yang sama akan memiliki atribut yang sama. Dalam pemrograman atribut adalah sebuah variable yang menampung data dari suatu objek. Contoh mobil memiliki atribut merk, warna, tahun produksi, dll. Sedangkan behavior adalah sesuatu yang dapat dilakukan oleh objek. Behavior dalam implementasi program ditulis dalam bentuk method yang dapat merepresentasikan fungsi yang dilakukan oleh suatu objek. Contoh mobil memiliki fungsi berjalan, belok, berhenti, dll.