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**

mainPraktikum.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  910  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | package cidro;  import java.util.Scanner;  public class mainPraktikum {  public static void main(String[] args) {  Scanner in = new Scanner (System.in);  praktikum mhs1 = new praktikum();  System.out.println("====SELAMAT DATANG DI SIAM====");  for (int i=0; i<3; i++) {  System.out.print("Nama :");  String nama = in.next();  System.out.print("NIM :");  String nim = in.next();  System.out.print("Jurusan :");  String jurusan = in.next();  System.out.print("IP :");  double ip = in.nextDouble();  System.out.print("Mata Kuliah :");  String km = in.next();  mhs1.setNama(nama);  mhs1.setNIM(nim);  mhs1.setJurusan(jurusan);  mhs1.setIP(ip);  mhs1.setKodematkul(km);  mhs1.displayData();  }  }  } |

Praktikum.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  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 | package cidro;  public class praktikum {    String nama;  String nim;  String jurusan;  String jumlahSks;  String kodematkul;  double ip;;    public void setNama(String name) {  nama = name;  }  public void setNIM (String N) {  nim = N;  }  public void setJurusan (String jurus) {  jurusan = jurus;  }  public void setKodematkul (String km) {  kodematkul = km;  if ("IFK5002".equals(km)) {  kodematkul = "Pemrograman Lanjut";  }  else if ("SIF15103".equals(km)) {  kodematkul = "Sistem Fungsional Bisnis";  }  else if ("PTI15005".equals(km)) {  kodematkul = "Interaksi Manusia dan Komputer";  }  else if ("PTI15008".equals(km)) {  kodematkul = "Sistem Operasi";  }  else if ("UNG4008".equals(km)) {  kodematkul = "Bahasa Indonesia";  }  else {  kodematkul = "Kode yang anda masukkan salah";  }  }  public void setIP (double I) {  ip = I;  if (ip >= 3.00) {  jumlahSks = "22-24";  }  else if (ip >= 2.50 && ip < 3.00) {  jumlahSks = "19-21";  }  else if (ip >= 2.00 && ip < 2.50) {  jumlahSks = "16-18";  }  else if (ip >= 1.50 && ip < 2.00) {  jumlahSks = "12-15";  }  else if (ip < 1.50) {  jumlahSks = "kurang dari 12";  }  }  public void displayData () {  System.out.println("========DATA MAHASISWA========");  System.out.println("Nama mahasiswa :" + nama);  System.out.println("NIM :" + nim);  System.out.println("Jurusan :" + jurusan);  System.out.println("IP :" + ip);  System.out.println("Mata Kuliah :" + kodematkul);  System.out.println("Pengambilan sks :" + jumlahSks);  System.out.println("==============================");  System.out.println();  }  } |

1. **PEMBAHASAN**

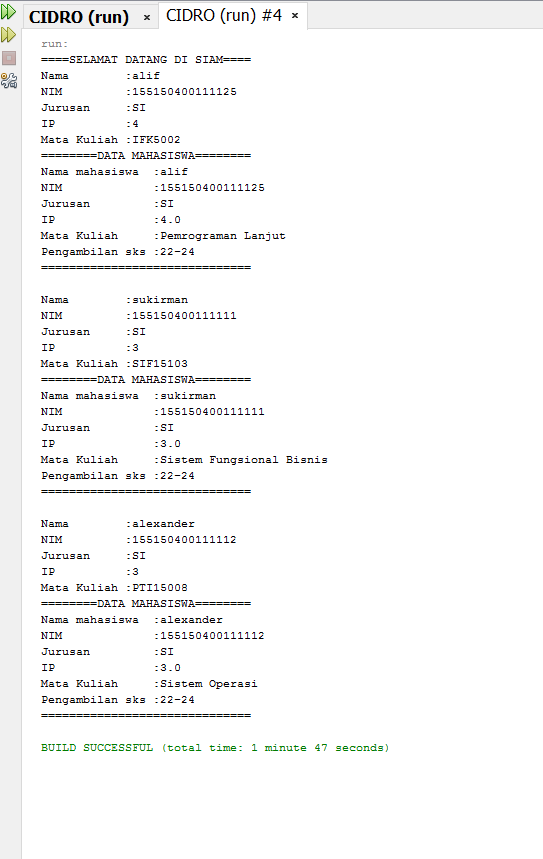
mainPraktikum.java

|  |  |
| --- | --- |
| 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 | Deklarasi package cidro  Mengimpor utilitas java yaitu scanner  Pendeklarasian kelas mainpraktikum  pendeklarasain main method  Pendeklarasian new Scanner in  Nama class praktikum dan nama class mhs1 = new nama class praktikum  Mencetak “====SELAMAT DATANG DI SIAM====”  Perulangan dengan tipe data integer  Mencetak “Masukkan Nama :”  Menunggu inputan user dengan variabel Nama dan tipe data String  Mencetak “NIM :”  Menunggu inputan user dengan variabel NIM dan tipe data String  Mencetak ”Jurusan :”  Menunggu inputan user dengan variabel Jurusan dan tipe data String  Mencetak “Masukkan IP :”  Menunggu inputan user dengan variabel IP dan tipe data double dengan parameter IP  Mencetak “Mata Kuliah”  Menunggu masukkan user dengan variabel km dan tipe data String  Objek mhs1 memanggil method setNama dengan parameter nama  Objek mhs1 memanggil method setNIM dengan parameter nim  Objek mhs1 memanggil method setJurusan dengan parameter jurusan  Objek mhs1 memanggil method setIP dengan parameter ip  Objek mhs1 memanggil method setKodematkul (km)  Objek mhs1 memanggil method displaydata  Menutup perulangan pada baris ke-7  Menutup main method  Menutup main class |

praktikum.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 | Deklarasi package cidro  Pendeklarasian class praktikum  Blank line  Inisialisasi variabel nama dengan tipe data string  Inisialisasi variabel nim dengan tipe data string  Inisialisasi variabel jurusan dengan tipe data string  Inisialisasi variabel jumlahSks dengan tipe data string  Inisialisasi variabel kodematkul dengan tipe data string  Inisialisasi variabel ip dengan tipe data double  Public void dengan nama setNama dan parameter String name  Assignment value variabel name ke dalam variabel nama  Menutup public void baris ke-11  Public void dengan nama setNIM dan parameter String N  Assignment value variabel N ke dalam variabel nim  Menutup public void baris ke-14  Public void dengan nama setJurusan dan parameter String jurus  Assignment value variabel jurus ke dalam variable jurusan  Menutup public void pada baris ke 17  Public void dengan nama setKodematkul dan parameter String km  Assignment value variabel km ke dalam variabel kodematkul  Seleksi kondisi jika kodeMatkul sama dengan masukkan yaitu IFK5002  Mencetak pemrograman lanjut disimpan dalam variabel kodematkul  Menutup seleksi kondisi pada baris ke 22  Seleksi kondisi jika kodeMatkul sama dengan masukkan yaitu SIF15103  Mencetak sistem fungsional bisnis disimpan dalam variabel kodematkul  Menutup seleksi kondisi pada baris ke 25  Seleksi kondisi jika kodeMatkul sama dengan masukkan yaitu PTI15005  Mencetak interaksi manusia dan komputer disimpan dalam variabel kodematkul  Menutup seleksi kondisi pada baris ke 28  Seleksi kondisi jika kodeMatkul sama dengan masukkan yaitu PTI15008  Mencetak system operasi disimpan dalam variabel kodematkul  Menutup seleksi kondisi pada baris ke 31  Seleksi kondisi jika kodeMatkul sama dengan masukkan yaitu UNG4008  Mencetak bahasa indonesia disimpan dalam variabel kodematkul  Menutup seleksi kondisi pada baris ke 34  Seleksi kondisi  Mencetak kode yang anda masukkan salah dismpan dalam variabel kodematkul  Menutup seleksi kondisi pada baris ke 37  Menutup public void pada baris ke 20  Public void dengan nama setIP dan parameter double I  Assignment value variabel I ke dalam variabel ip  Seleksi kondisi jika ip lebih besar sama dengan 3,00.  Mencetak sks yang dapat diambil : 22-24 SKS  Menutup seleksi kondisi pada baris ke 43  Seleksi kondisi jika ip lebih besar sama dengan 2,50 dan lebih kecil sama dengan 2,99.  Mencetak sks yang dapat diambil : 19-21 SKS  Menutup seleksi kondisi pada baris 46  Seleksi kondisi jika ip lebih besar sama dengan 2,00 dan kurang dari sama dengan 2,50.  Mencetak sks yang dapat diambil : 16-18 SKS  Menutup seleksi kondisi pada baris 49  Seleksi kondisi if jika ip lebih besar sama dengan 1,50 dan IP kurang dari sama dengan 2,00.  Mencetak sks yang dapat diambil : 12-15 SKS  Menutup seleksi kondisi pada baris 52  Seleksi kondisi jika ip kurang dari 1,50  Mencetak sks yang dapat diambil : <12 SKS  Menutup seleksi kondisi pada baris 31  Menutup public void pada baris 41  Public void displayData  Mencetak "=========DATA MAHASISWA========"  Mencetak "Nama mahasiswa :" +nama  Mencetak "NIM :" +nim  Mencetak "Jurusan :"+Jurusan  Mencetak "IP :"+IP  Mencetak "Mata Kuliah :" +kodematkul  Mencetak "Pengambilan sks :" +sks  Mencetak " =============================="  Mencetak ()  Menutup public void display data  Menutup main class |

1. **SCREENSHOT PROGRAM**

****

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

* Variabel instance milik sebuah instance dari kelas.
* Variabel lokal 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 ada kesalahan

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 | 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);  }  } |

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 | public class MainMobil {  public static void main(String[] args) {  //instan objek bernama m1  Mobil m1 = new Mobil();  m1.setKecepatan(50);  m1.setManufaktur("Toyota");  m1.setNoPlat("AB 1231 UA");  m1.setWarna("Merah");  m1.displayMessage();  System.out.println("================");  //instan objek baru bernama m2  Mobil m2 = new Mobil();  m2.setKecepatan(100);  m2.setManufaktur("Mitsubishi");  m2.setNoPlat("N 1134 AG");  m2.setWarna("Pink");  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();  }  } |

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 | import java.util.Scanner;  public class Mainmobil {  public static void main(String[] args) {  Scanner in = new Scanner(System.in);  for (int i = 0; i < 5; i++) {  mobil m1 = new mobil();  System.out.print("Merek mobil : ");  String manufaktur = in.next();  m1.setManufaktur(manufaktur);  System.out.print("Nomor Plat : ");  String noPlat = in.next();  m1.setNoPlat(noPlat);  System.out.print("Memiliki warna : ");  String warna = in.next();  m1.setWarna(warna);  System.out.print("Masukkan kecepatan dalam (km/jam) :");  int kecepatan = in.nextInt();  m1.setKecepatan (kecepatan);  m1.displayMessage();  System.out.println("==============================");  System.out.println("mobil pada objek m1 di rubah menjadi warna hijau");  m1.setWarna("Hijau");  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 | public class Mobil {  String noPlat;  String warna;  String manufaktur;  int kecepatan ;  double waktu;  public void setNoPlat(String plat){  noPlat = plat;  }  public void setWarna(String warna){  this.warna = warna;  }  public void setManufaktur(String faktur){  manufaktur = faktur;  }  public void setKecepatan(int kec){  kecepatan = kec;  }  public void setWaktu (double waktu){  this. waktu = waktu;  }  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("dengan watu "+waktu +" second"); |

Mainmobil.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | System.out.print("Masukkan waktu (jam :");  double waktu = in.nextDouble();  waktu = waktu\*3600;  m1.setWaktu(waktu);  m1.rubahSecond(waktu);  m1.rubahkecepatan();  m1.Jarak();  m1.displayMessage();  System.out.println("==============================");  System.out.println("mobil pada objek m1 di rubah menjadi warna hijau");  m1.setWarna("Hijau");  m1.displayMessage();  System.out.println("------------------------------");  }  }  } |

1. Tambahkan method bernama rubahSekon mempunyai parameter bertipe double dan hanya dapat dipanggil pada class mobil. Method ini memiiliki 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 | public void setWaktu(double Time) {  waktu = Time;  rubahSekon(waktu);  }  private void rubahSekon(double Time) {  waktu = Time \* 3600;  } |

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 | public void setKecepatan (int i) {  kecepatan = I;  rubahKecepatan (kecepatan);  }  Private void rubahKecepatan (int g) {  g = g \* 1000;  kecepatan = g / 3600; |

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 | private void rubahSekon(double Time) {  waktu = Time \* 3600;  hitungJarak (waktu, kecepatan);  }  public void hitungJarak(double waktu, int kecepatan) {  jarak = kecepatan \* waktu;  rubahKilometer(jarak);  } |

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 | public void hitungJarak(double waktu, int kecepatan) {  jarak = kecepatan \* waktu;  rubahKilometer(jarak);  }  private void rubahKilometer (double p) {  jarak = p / 1000; |

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.

Buku.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | public class Buku {  private int lembar;  private int perhari;  private int katadalambuku;  private int waktu;  public void Lembar(int x) {  lembar = x;  }  public void KaPeHa(int x) {  perhari = x;  }  public int HalPeHa() {  return (perhari / 100) \* 2;  }  public void Wak() {  waktu = lembar \* 4;  }  public void KaDaBu() {  katadalambuku = perhari \* waktu;  }  public void DisplayMassage() {  System.out.println("Jumlah Lembar : " + lembar);  System.out.println("Jumlah Kata per Hari : " + perhari);  System.out.println("Kata yang cukup untuk mengisi setengah halaman : " + perhari);  System.out.println("Jumlah waktu untuk menghabiskan satu buku : " + waktu + " hari");  System.out.println("Jumlah kata dalam satu buku : " + katadalambuku+" kata");  }  } |

mainBuku.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | public class mainBuku {  public static void main(String[] args) {  Buku input= new Buku();  input.Lembar(50);  input.KaPeHa(100);  input.Wak();  input.KaDaBu();  input.DisplayMassage();  }  } |

1. **KESIMPULAN**
2. Object Oriented Programming (OOP) adalah sebuah cara pembuatan program dengan menggunakan konsep “objek” yang memiliki data atau atribut yang menjelaskan tentang objek dan prosedur atau fungsi yang dikenal dengan method.
3. Membuat instan Objek dari sebuah class dilakukan dengan menggunakan kata kunci new. Contohnya pada suatu kasus kita memiliki Class bernama mahasiswa dan kita ingin menginstan objek dari class Mahasiswa pada class mainMahasiswa dan kita beri nama mahasiswa A.

Mahasiswa.java

public class Mahasiswa{

}

mainMahasiswa.java

public class mainMahasiswa{

public static void main(String[] args){

Mahasiswa mahasiswa A = new Mahasiswa();

}

}

3.pengertian

1. Class adalah struktur dasar dari OOP(Object Oriented Programming). Terdiri dari dua tipe yaitu : field dan method. Class digunakan untuk mendeklarasikan sebuah variabel yang berupa objek atau dinamakan referensi objek
2. Object adalah sebuah perangkat lunak yg berisi beberapa variabel dan method yang saling berhubungan.
3. Attribut berfungsi sebagai data, didefinisikan oleh class, individu, berbeda satu dengan lainnya dan menggambarkan tampilan, status, kualitas dari object.
4. Behavior berlaku sebagai method dan menggambarkan bagaiman sebuah instance class beroperasi.