1. **DEFINISI MASALAH**

Buatlah implementasi sebuah mesin ATM. Dalam mesin ATM, password dapat dimasukkan oleh user. Selain itu, sistem memiliki piihan menuli melihat saldo, menarik uang, dan mentransfer ke rekening lain!

1. **SOURCE CODE**

|  |  |
| --- | --- |
|  | **Nasabah.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 | package Modul\_2;  public class Nasabah {  private int noRekening, pin, saldo;  private String nama;  public Nasabah (){  saldo = 0;  pin = 0;  noRekening = 0;  nama = "";  }  public Nasabah (int Saldo, int pin, int noRekening, String nama){  this.saldo = Saldo;  this.pin = pin;  this.noRekening = noRekening;  this.nama = nama;  }    public void setSaldo (int saldo){  this.saldo = saldo;  }  public void setNama (String nama){  this.nama = nama;  }  public int getSaldo(){  return saldo;  }  public int getPin(){  return pin;  }  public int getNoRekening() {  return noRekening;  }  public String getNama() {  return nama;  }  } |

|  |  |
| --- | --- |
|  | **ATM.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 | package Modul\_2;  public class ATM {  private Nasabah user[] = new Nasabah[3];  private int c = -1;  private static int temp, penarikan;  public ATM(){  user[0] = new Nasabah (1000000, 1023, 1234, "Anto");  user[1] = new Nasabah (1500000, 1022, 2234, "Barli");  user [2] = new Nasabah (2000000, 2021, 3334, "Cahyo");  }  public void login (int a){  for (int i = 0 ; i < 3; i++){  if (user[i].getPin() == a){  System.out.println("-----------------------------------------");  System.out.println("\tLOGIN BERHASIL\t");  System.out.println("-----------------------------------------");  System.out.println("Nama Nasabah:\t" + user[i].getNama());  System.out.println("No. Rekening:\t"+user[i].getNoRekening());  System.out.println("");  c = i;  break;  }  }  }  public int getCari(){  return c;  }  public int cekSaldo(){  return user[getCari()].getSaldo();  }  public void transfer (int c, int n){  for (int i = 0; i <3;i++){  if (user[getCari()].getSaldo() - c < 0){  System.out.println("Maaf, Saldo Tidak Mencukupi ...");  break;  }  if (user[getCari()].getSaldo() - c >= 0){  if (user[i].getNoRekening() == n){  user[getCari()].setSaldo(user[getCari()].getSaldo() - c);  System.out.println("Anda Mengirim Uang Ke\t : " + n);  System.out.println("Atas Nama\t\t : " +user[i].getNama());  System.out.println("Dengan Nominal\t\t : "+ c);  System.out.println("PROSES BERHASIL");  System.out.println("");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  break;  }  }  }  }  public void penarikan (int a){  for (int i = 0; i < 3; i++){  if (user[getCari()].getSaldo() - a < 0){  System.out.println("Maaf, Saldo Tidak Mencukupi ...");  break;  }  if (user[getCari()].getSaldo() - a >= 0){  user[getCari()].setSaldo(user[getCari()].getSaldo() - a);  System.out.println("PROSES BERHASIL");  break;  }  }  }  } |

|  |  |
| --- | --- |
|  | **MainATM.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 | package Modul\_2;  import java.util.Scanner;  public class MainATM {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  ATM atm = new ATM();  do {  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("\tSELAMAT DATANG DI ATM MORIS\t\t");  System.out.print("Silahkan masukkan pin anda : ");  int pin = input.nextInt();  atm.login(pin);  if (atm.getCari() == -1){  System.out.println("PIN YANG ANDA MASUKKAN SALAH");  System.out.println("SILAHKAN MASUKKAN KEMBALI");  }  } while (atm.getCari() == -1);  for (int i = 0; i < 3; i++){  while (true){  for (int j = 1; j > 0; j++){  System.out.println("");  System.out.println("-------------------------------------");  System.out.println("\tMENU\t\t\t");  System.out.println("-------------------------------------");  System.out.println("1. Cek Saldo");  System.out.println("2. Penarikan Tunai");  System.out.println("3. Transfer");  System.out.println("4. Keluar");  System.out.println("");  System.out.print("Masukkan Pilihan Anda: ");  int pil = input.nextInt();  switch (pil){  case 1 :  System.out.println("\t\tCEK SALDO\t\t");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("Saldo Anda sebesar Rp "+ atm.cekSaldo());  break;  case 2 :  System.out.println("\t\tPENARIKAN TUNAI\t\t");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.print("Masukkan Jumlah Uang Yang Akan Diambil : ");  int a = input.nextInt();  atm.penarikan(a);  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  break;  case 3 :  System.out.println("\t\tTRANSFER\t\t");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.print("Masukkan Nomor Rekening Tujuan : ");  int y = input.nextInt();  System.out.print("Masukkan Jumlah Uang yang Ingin Dikirim : ");  int z = input.nextInt();  atm.transfer(z,y);  break;  case 4 :  System.out.println("\t\tTERIMA KASIH\t\t");  System.exit(0);  break;  default :  System.out.println("Pilihan yang Anda Masukkan Tidak Ada");  }  System.out.println("Apakah Anda Ingin Melakukan Transaksi Lagi ? <y/n>");  String y = input.next();  if ("y".equals(y)){  } else if ("n".equals(y)){  System.out.println("TERIMA KASIH");  System.exit(0);  }  }  }  }  }  } |

1. **PEMBAHASAN**

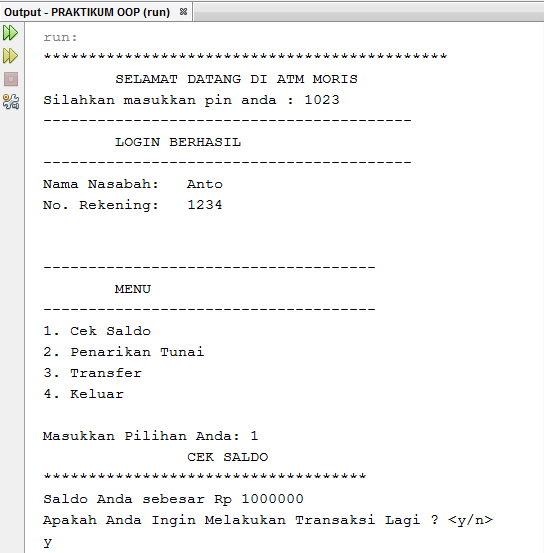
|  |  |
| --- | --- |
|  | **Nasabah.java** |
| 2  3  4  5  6  7  8  10  11  12  13  14  15  18  19  21  22  24  25  27  28  30  31  33  34 | Deklarasi kelas Nasabah  Inisialisasi variable noRekening, pin, dan saldo dengan tipe data integer  Inisialisai variable nama dengan tipe data String  Deklarasi Default Constructor dengan nama Nasabah  Inisialisasi variable saldo  Inisialisasi variable pin  Inisialisasi variable noRekening  Inisialisasi variable nama  Deklarasi constructor dengan parameter integer Saldo, integer noRekening, String nama  Inisialisasi variable saldo  Inisialisasi variable pin  Inisialisasi variable noRekening  Inisialisasi variable nama  Deklarasi method setSaldo untuk member instance variable saldo dengan parameter integer  Inisialisasi variable saldo  Deklarasi method setNama untuk memberi instance variable nama dengan parameter String  Inisialisasi variable nama  Deklarasi method getSaldo untuk mengambil value pada instance variable  Deklarasi return untuk mengembalikan nilai saldo  Deklarasi method getPin untuk mengambil value pada instance variable  Deklarasi return untuk mengembalikan nilai pin  Deklarasi method getNoRekening untuk mengambil value pada instance variable  Deklarasi return untuk mengembalikan nilai noRekening  Deklarasi method getNama untuk mengambil value pada instance variable  Deklarasi return untuk mengembalikan nilai saldo |

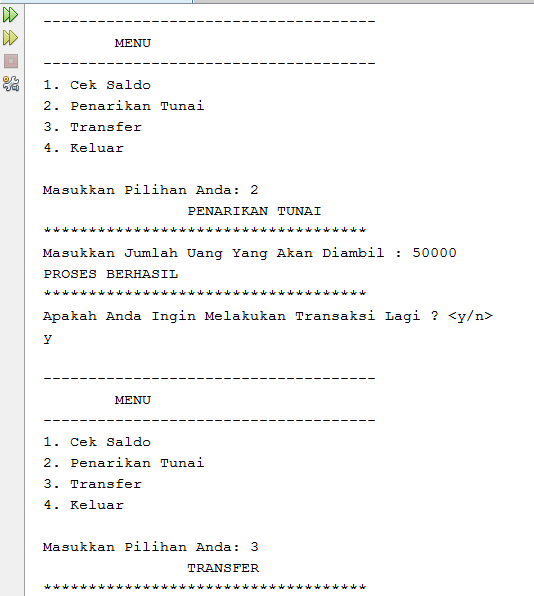
|  |  |
| --- | --- |
|  | **ATM.java** |
| 2  3  4  5  6  7  8  9  11  12  13  14  15  16  17  18  19  20  21  25  26  28  29  31  32  33  34  35  37  38  40  41  42  43  44  45  46  47  52  53  54  55  56  60  61  62 | Deklarasi kelas dengan nama ATM  Instansiasi objek dengan nama user dari kelas Nasabah dengan panjang array 3  Deklarasi variable c=-1  Deklarasi static variable temp dan penarikan dengan tipe data integer  Deklarasi default constructor dengan nama ATM  Deklarasi objek user pada array ke 0 dari kelas Nasabah dengan parameter 10000000, 1023, 1234, "Anto”  Deklarasi objek user pada array ke 1 dari kelas Nasabah dengan parameter 1500000, 1022, 2234, "Barli”  Deklarasi objek user pada array ke 2 dari kelas Nasabah dengan parameter 2000000, 2021, 3334, "Cahyo”  Deklarasi method void login dengan parameter integer a  Deklarasi for, akan berulang sebanyak tiga kali  Deklarasi if dengan kondisi pin yang diinput user sesuai dengan pin yang tersedia  Menampilkan output berup garis  Menampilkan output berupa LOGIN BERHASIL  Menampilkan output berupa garis  Menampilkan output berupa nama nasabah  Menampilkan output berupa no rekening  Menampilkan output berupa enter  Inisialisasi variable c = i  Deklarasi break untuk menghentikan perulangan  Deklarasi method getCari untuk mengambil value dari instance variable c  Deklarasi return untuk mengembalikan nilai c  Deklarasi method cekSaldo dengan tipe integer  Deklarasi return untuk mengembalikan nilai saldo  Deklarasi method transfer dengan tipe void dengan parameter no rekening dan uang  Deklarasi for dengan kondisi i<3 , akan berulang sebanyak tiga kali  Deklarasi if dengan kondisi ketika saldo setelah dikurangi uang yang ditransfer kurang dari 0  Menampilkan output berupa saldo tidak mencukupi  Deklarasi break untuk menghentikan perulangan  Deklarasi if dengan kondisi ketika saldo setelah dikurangi uang yang ditransfer lebih dari 0  Deklarasi if dengan kondisi ketika no rekening yang diinput sama dengan yang tersedia  Perhitungan saldo nasabah dikurangi uang yang ditransfer  Menampilkan output berupa tujuan transfer  Menampilkan output berupa tujuan transfer  Menampilkan output berupa nominal uang  Menampilkan output berupa BERHASIL  Menampilkan output berupa enter  Menampilkan output berupa garis  Deklarasi break untuk menghentikan perulangan  Deklarasi method penarikan dengan parameter a, a adalah uang yang ditarik  Deklarasi for dengan kondisi i<3, akan berulang sebanyak tiga kali  Deklarasi if dengan kondisi saldo dikurangi uang yg ditarik kurang dari 0  Menampilkan output berupa Maaf, Saldo Tidak Mencukupi . . .  Deklarasi break untuk menghentikan perulangan  Proses perhitungan saldo dikurangi uang yang diambil  Menampilkan output berupa PROSES BERHASIL  Delarasi break untuk menghentikan perulangan |

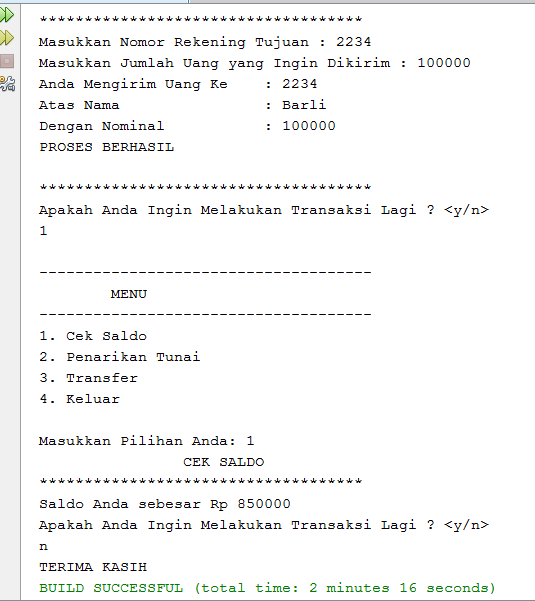
|  |  |
| --- | --- |
|  | **MainATM.java** |
| 2  3  4  5  6  7  9  11  12  13  14  15  16  17  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  39  40  41  42  43  44  45  46  47  49  50  51  53  55  56  57  58  59  60  61  62  63  64  65  66  67  69  70  71  72  73  74 | Deklarasi import dari package Scanner  Deklarasi class dengan nama MainATM  Deklarasi method main  Instansiasi alat java berupa Scanner dengan nama input  Instansiasi objek atm dari kelas ATM  Deklarasi do  Menampilkan output berupa garis  Menampilkan output berupa SELAMAT DATANG DI ATM MORIS  Menampilkan output berupa perintah menginput pin  Deklarasi variable pin yang disediakan oleh scanner  Memanggil method login dengan parameter pin  Deklarasi if dengan kondisi pencarian = -1  Menampilkan output berupa PIN YANG ANDA MASUKKAN SALAH  Menampilkan output berupa perintah menginput pin kembali  Deklarasi while ketika hasil pencarian = -1  Deklarasi for dengan kondisi i<3 akan berulang sebanyak tiga kali  Deklarasi while dengan kondisi true  Deklarasi for dengan kondisi j>0 akan berulang sampai diberhentikan  Menampilkan output berupa enter  Menampilkan output berupa garis  Menampilkan output berupa MENU  Menampilkan output berupa garis  Menampilkan output berupa pilihan 1. cek saldo  Menampilkan output berupa pilihan 2. penarikan tunai  Menampilkan output berupa pilihan 3. transfer  Menampilkan output berupa pilihan 4. keluar  Menampilkan output berupa enter  Menampilkan output berupa perintah menginput pilihan menu  Deklarasi variable pil yang disediakan oleh scanner  Deklarasi switch dengan ekspresi statement pil  Deklarasi case 1  Menampilkan output berupa CEK SALDO  Menampilkan output berupa garis  Menampilkan output berupa nilai saldo  Deklarasi break untuk menghentikan case  Deklarasi case 2  Menampilkan output berupa PENARIKAN TUNAI  Menampilkan output berupa garis  Menampilkan output berupa perintah menginput uang  Deklarasi variable a yang disediakan oleh Scanner  Memanggil method penarikan dengan parameter a  Menampilkan output berupa garis  Deklarasi break untuk menghentikan case  Deklarasi case 3  Menampilkan output berupa TRANSFER  Menampilkan output berupa garis  Menampilkan output berupa perintah input no rekening  Deklarasi variable y yang disediakan scanner  Menampilkan output berupa perintah menginput uang  Deklarasi variable z yang disediakan Scanner  Memanggil method transfer dengan parameter(z,y)  deklarasi break untuk menghentikan case  deklarasi case 4  Menampilkan output berupa terima kasih  Deklarasi untuk menghentikan program  Deklarasi break  Deklarasi default  Menampilkan output berupa pilihan tidak ada  Menampilkan output berupa pertanyaan untuk transaksi lagi  Deklarasi variable ya yang disediakan oleh Scanner  Deklarasi if dengan kondisi input sama dengan ya  Deklarasi else if dengan kondisi input tidak sama dengan ya  Menampilkan output berupa terima kasih  Deklarasi untuk menghentikan program |

1. **SCREENSHOT PROGRAM**

**ATM**



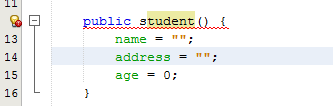




1. **PRAKTIKUM**
2. Constructor
3. Lakukan percobaan constructor diatas dan benahi jika menemukan kesalahan

***Jawab :*** Terjadi kesalahan pada baris 12 di Student.java pada default constructor dan pada baris 21 di MainStudent.java saat merubah informasi objek pada class

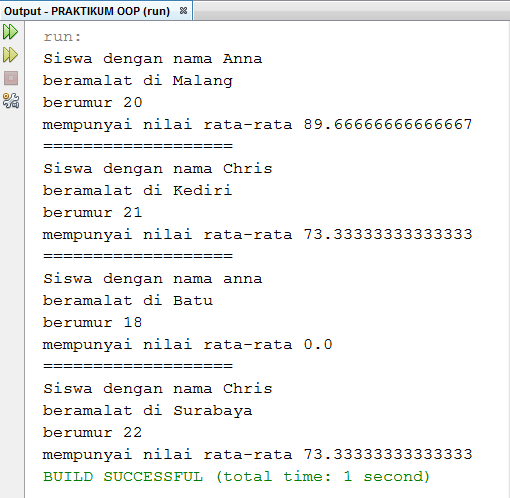
Student.java



MainStudent.java



Output



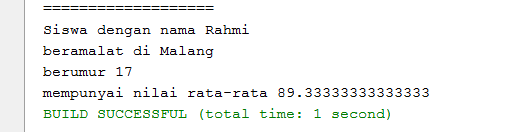
1. Tambahkan constructor pada class Student dengan parameter yang mempunyai

parameter masing masing nilai dari mata pelajaran yang ada! Kemudian buat contoh objeknya pada main Class!

***Jawab :***

|  |  |
| --- | --- |
|  | Student.java |
|  | public Student (double m, double e, double s){  mathGrade = m;  englishGrade = e;  scienceGrade = s;  } |

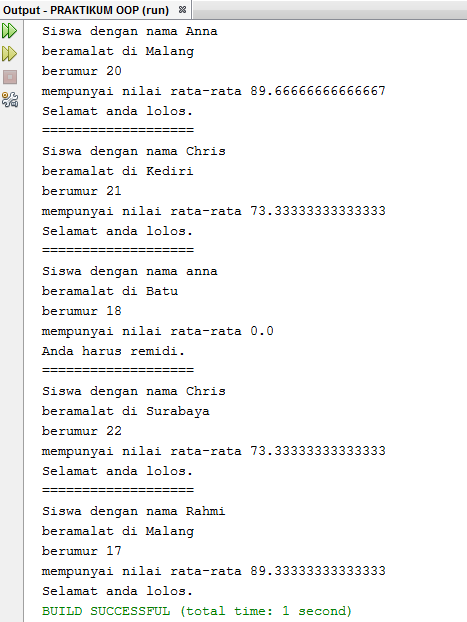
|  |  |
| --- | --- |
|  | MainStudent.java |
|  | Student rahmi = new Student(90, 85, 93);  rahmi.setName("Rahmi");  rahmi.setAddress("Malang");  rahmi.setAge(17);  rahmi.displayMessage(); |



1. Tambahkan method dengan nilai balikan berupa boolean pada class student bernama statusAkhir untuk menentukan apakah siswa tersebut remidi atau tidak. Ketentuannya adalah jika nilai lebih dari atau sama dengan 61 adalah lolos sedangkan nilai kurang dari atau sama dengan 60 adalah remidi.Nilai yang dicari adalah nilai ratarata untuk semua mapel. Kemudian nilai pada method statusAkhir tampilkan pada method displayMessage!

***Jawab :***

|  |  |
| --- | --- |
|  | Student.java |
|  | private double getAverage() {  double result = 0;  result = (mathGrade + scienceGrade + englishGrade) / 3;  return result;  }  public boolean statusAkhir() {  boolean hasil;  if (getAverage() >= 61) {  hasil = true;  } else {  hasil = false;  }  return hasil;  }  public void displayMessage() {  System.out.println("Siswa dengan nama " + name);  System.out.println("beramalat di " + address);  System.out.println("berumur " + age);  System.out.println("mempunyai nilai rata-rata " + getAverage());  if (statusAkhir() == true) {  System.out.println("Selamat anda lolos.");  } else {  System.out.println("Anda harus remidi.");  } |



1. Bagaimana cara memasukkan jumlah siswa sesuai dengan keinginan user? Tuliskan kodenya dengan inputan user yang interaktirf (key : menggunakan array)

***Jawab :*** Menerima masukan banyak siswa sesuai keinginan user dapat menggunakan Scanner kemudian dibuat variabel siswa menggunakan array dengan jumlah sesuai dengan banyak siswa.

|  |  |
| --- | --- |
|  | MainStudent.java |
|  | import java.util.Scanner;  Scanner input = new Scanner (System.in);  Student siswa []=new Student[3];  for( int i=0; i<siswa.length;i++){  siswa[i]=new Student();  System.out.println("Data Mahasiswa ke- "+(i+1));  System.out.print("Masukkan Nama Siswa: ");  String nama=input.next();  siswa[i].setName(nama);  System.out.print("Masukkan Umur: ");  int umur=input.nextInt();  siswa[i].setAge(umur);  System.out.print("Masukkan Alamat: ");  String alamat=input.next();  siswa[i].setAddress(alamat);  System.out.print("Masukkan Nilai Matematika:");  int mtk=input.nextInt();  siswa[i].setMath(mtk);  System.out.print("Masukkan Nilai Science: ");  int s=input.nextInt();  siswa[i].setMath(s);  System.out.print("Masukkan Nilai Bahasa Inggris: ");  int big=input.nextInt();  siswa[i].setMath(big);  System.out.println("");  System.out.println("----------------------------------------------");  siswa[i].displayMessage();  System.out.println("----------------------------------------------");  } |

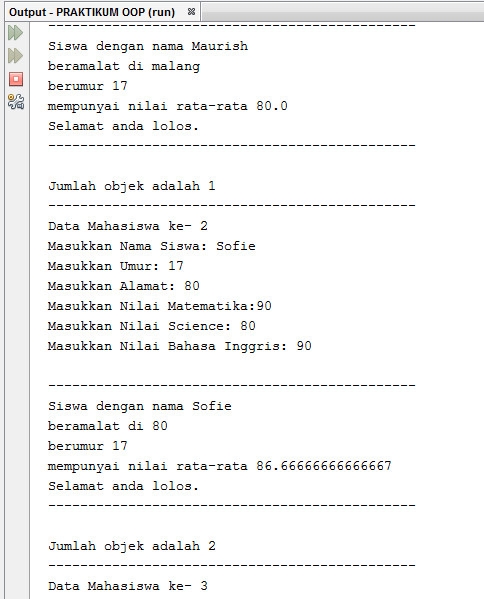


1. Bagaimana cara menghitung banyaknya objek yang kita buat dari sebuah menginstance objek dari main class? Tuliskan kodenya kemudian tampilkan informasinya dengan memanggil method jumlahObjek() bertipe void!

***Jawab :***

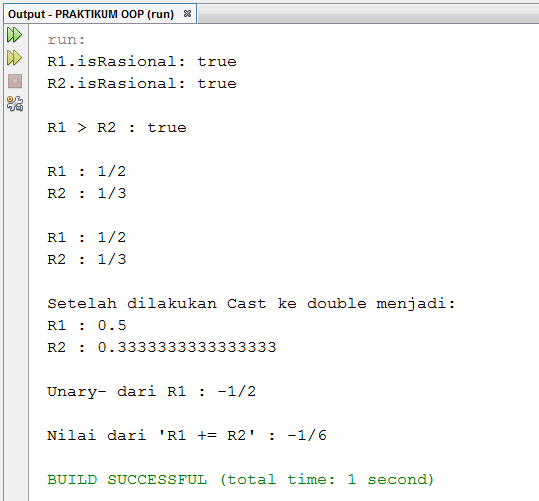
|  |  |
| --- | --- |
|  | Student.java |
|  | private static int studentCount;  public Student() {  studentCount++;  name = "";  address = "";  age = 0;  }  //nomor 2 constructor  public Student(double m, double e, double s) {  studentCount++;  mathGrade = m;  englishGrade = e;  scienceGrade = s;  } |

|  |  |
| --- | --- |
|  | MainStudent.java |
|  | System.out.println("----------------------------------------------");  Student.jumlahObjek();  System.out.println("----------------------------------------------"); |



1. Instance Method
2. Lakukan percobaan Instance Method diatas dan benahi jika menemukan kesalahan!

***Jawab :*** Tidak ada kesalahan, program dapat dijalankan



1. Tambahkan method untuk operator, <. <=, => !

***Jawab :***

|  |  |
| --- | --- |
|  | Rasional.java |
|  | public boolean lessThan(Rasional A) {  return (pembilang \* A.penyebut < penyebut \* A.pembilang);  }  public boolean lessThanEquals(Rasional A) {  return (pembilang \* A.penyebut <= penyebut \* A.pembilang);  }  public boolean moreThanEquals(Rasional A) {  return (pembilang \* A.penyebut >= penyebut  \* A.pembilang);  } |

1. Ubah method sederhana pada baris 25-30 yang awalnya adalah menggunakan while menjadi for!

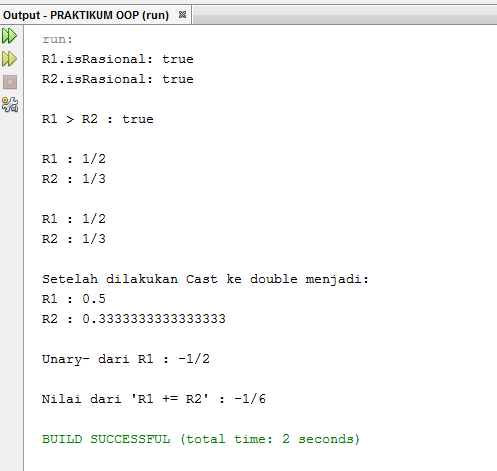
***Jawab :***

|  |  |
| --- | --- |
|  | Rasional.java |
|  | for (; B != 0;) {  temp = A % B;  A = B;  B = temp;  } |

1. Tambahkan method untuk operasi -,\*,/ !

***Jawab :***

|  |  |
| --- | --- |
|  | Rasional.java |
| 1 | public void unaryMinus(Rasional A) {  pembilang = pembilang \* A.penyebut - penyebut  \* A.pembilang;  penyebut \*= A.penyebut;  Sederhana();  }  public void multiply(Rasional A) {  pembilang \*= A.pembilang;  penyebut \*= A.penyebut;  Sederhana();  }  public void devide(Rasional A) {  pembilang /= A.pembilang;  penyebut /= A.penyebut;  Sederhana();  } |



1. **KESIMPULAN**
2. Jelaskan apa yang disebut dengan konstruktor

***Jawab :*** Konstruktor adalah method yang memiliki nama yang sama dengan class, dimana seluruh inisialisasi objek ditempatkan. Konstruktor tidak memiliki return value meskipun void.

Contoh syntax konstruktor :

public class Kendaraan{

public Kendaraan (){

//statement

}

}

1. Jelaskan perbedaan konstruktor dan method pada umumnya

***Jawab :*** Perbedaan method biasa dengan constructor adalah bahwa constructor harus memiliki nama yang sama dengan nama class-nya dan tidak memiliki nilai kembalian.

1. Tuliskan cara mendeklarasikan konstruktor

***Jawab :*** Constructor dijalankan pada saat sebuah objek diinisialisasi dengan cara menggunakan kata new pada saat menginstan objek dari class.

1. Jelaskan apa yang disebut dengan instance method

***Jawab :*** Instance method dapat didefinisikan sebagai lebih dari satu method dan dapat dipanggil dari method yang berbeda. Instance Method bisa dikatakan juga sebagai fungsi yg beroperasi pada objek untuk memanipulasi state objek lewat pesan yg diterima dari obek lain. Contoh :

public class Belajar{

public String baca;

public String tulis;

public void cetak(){

System.out.println(“bacaan diatas adalah”+baca);

Public void cetak1(){

System.out.println(“tulisan diatas adalah”+tulis); }

}

1. Bagaimana cara memanggil sebuah instance method

***Jawab :*** Untuk memanggil method ini harus membuat objek dengan instance class.

Bentuk Umum:

void namaMethodInstance(){

Untuk memanggil sebuah instance method, kita dapat menuliskan :

namaObjek.namaMethod();