1. **DEFINISI MASALAH**
2. Anda saat ini bekerja sebagai IT engineer di bank XY, dan anda diminta untuk membuat sistem transaksi yang ada di mesin ATM yang baru dibeli. Aplikasi yang anda buat harus memenuhi requirement yang telah dirancang oleh analis di bank XY. Kira-kira begini requirementnya:

- Informasi rekening (saldo, nomor rekening, nama pemegang rekening) tidak bisa diubah ubah oleh nasabah secara langsung.

- Nomer rekening terdiri dari 15 digit, yang mana 3 digit awal menentukan jenis rekening yang dimiliki oleh nasabah:

- 001 : rekening gold, memiliki limit penarikan tunai 10 juta, tanpa biaya transfer

- 192 : rekening silver, memiliki limit penarikan tunai 7 juta, namun ketika transfer akan dikenakan biaya admin 6500

- 283 : rekening bronze, memiliki limit penarikan tunai 5 juta, namun ketika transfer akan dikenakan biaya admin 7500

- Nasabah harus memiliki saldo minimal 10000, jika saldo pasca transaksi kurang dari batas minimal tadi, maka transaksi dianggap gagal

- Buatlah sistem ATM tadi terbatas pada penarikan tunai dan transfer saja, dan sebelum memulai transaksi harus melakukan otentikasi dahulu dengan memasukkan PIN dan nomor rekening.

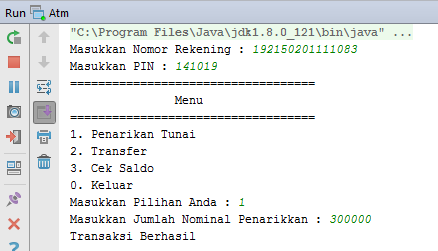
1. **SOURCE CODE**

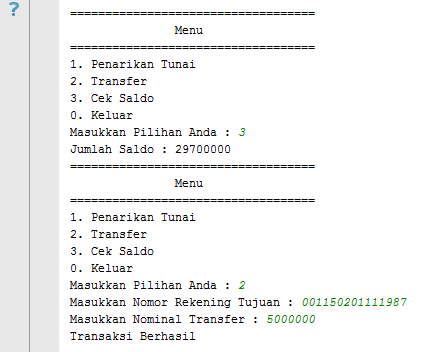
|  |  |
| --- | --- |
|  | Atm.java |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226 | import java.util.ArrayList;  import java.util.Scanner;  /\*\*  \* Created by Zain Fikri H on 01/04/2017.  \*/  public class Atm {  public static ArrayList<Rekening> boxNasabah = new ArrayList<>();  public void setNasabah(String nama, String noRek, String pin, long saldo){  Rekening rekening = new Rekening(nama,noRek,pin,saldo);  boxNasabah.add(rekening);  }  public static void main(String[] args) {  Scanner in = new Scanner(System.in);  Atm list = new Atm();  list.setNasabah("Zain Fikri Hanastyono", "192150201111083", "141019", 30000000);  list.setNasabah("Aditya Putra", "001150201111987", "101914", 20000000);  list.setNasabah("Ricky Yuliantoro", "283150201111875", "765487", 20000000);  System.out.print("Masukkan Nomor Rekening : ");  String noRekening = in.next();  System.out.print("Masukkan PIN : ");  String pin = in.next();  if(noRekening.length()==15){  list.display(noRekening,pin);  }else{  System.out.println("Nomor Rekening Dan PIN Yang Anda Masukkan Salah");  }  }  public void display(String noRekening, String pin){  Scanner in = new Scanner(System.in);  String cekRek = noRekening.substring(0,3);  int cek = 0;  int limit = 0;  if (cekRek.equalsIgnoreCase("001")){  cek = 1;  }if (cekRek.equalsIgnoreCase("192")){  cek = 2;  }if (cekRek.equalsIgnoreCase("283")) {  cek = 3;  }  boolean status = true;  for(Rekening nasabah : boxNasabah){  boolean sistem=true;  while (sistem){  if(nasabah.getNoRek().equalsIgnoreCase(noRekening) && nasabah.getPin().equalsIgnoreCase(pin)){  if(nasabah.getSaldo()==10000){ status = false;  System.out.println("===================================");  System.out.println(" Menu");  System.out.println("===================================");  System.out.println("1. Penarikan Tunai");  System.out.println("2. Transfer");  System.out.println("3. Cek Saldo");  System.out.println("0. Keluar");  System.out.print("Masukkan Pilihan Anda : ");  int pilihan = in.nextInt();  boolean ulang = true;  while (ulang){  switch (pilihan){  case 1:  System.out.print("Masukkan Jumlah Nominal Penarikkan : ");  double tarik = in.nextInt();  if(cek==1){  if(limit<=10000000){  if(tarik<=10000000){  limit += tarik;  nasabah.tarik(tarik);  System.out.println("Transaksi Berhasil");  ulang = false;  }else{  System.out.println("Maaf jumlah penarikan Anda melebihi batas");  ulang = false;  }  }else{  System.out.println("Maaf Anda sudah mencapai batas penarikan");  ulang = false;  }  }if(cek==2){  if(limit<=7000000){  if(tarik<=7000000){  limit += tarik;  nasabah.tarik(tarik);  System.out.println("Transaksi Berhasil");  ulang = false;  }else{  System.out.println("Maaf jumlah penarikan Anda melebihi batas");  ulang = false;  }  }else{  System.out.println("Maaf Anda sudah mencapai batas penarikan");  ulang = false;  }  }if(cek==3){  if(limit<=5000000){  if(tarik<=5000000){  limit += tarik;  nasabah.tarik(tarik);  System.out.println("Transaksi Berhasil");  ulang = false;  }else{  System.out.println("Maaf jumlah penarikan Anda melebihi batas");  ulang = false;  }  }else{  System.out.println("Maaf Anda sudah mencapai batas penarikan");  ulang = false;  }  }  break;  case 2:  System.out.print("Masukkan Nomor Rekening Tujuan : ");  String transfer = in.next();  System.out.print("Masukkan Nominal Transfer : ");  long jumlah = in.nextLong();  if(cek==1){  for (int i=0; i<boxNasabah.size();i++){  if(boxNasabah.get(i).getNoRek().equalsIgnoreCase(transfer)){  nasabah.transfer(boxNasabah.get(i),jumlah);  System.out.println("Transaksi Berhasil");  }  }  ulang = false;  }if(cek==2){  for (int i=0; i<boxNasabah.size();i++){  if(boxNasabah.get(i).getNoRek().equalsIgnoreCase(transfer)){  nasabah.transfer(boxNasabah.get(i),jumlah);  nasabah.tarik(6500);  System.out.println("Transaksi Berhasil");  }  }  ulang=false;  }if(cek==3){  for (int i=0; i<boxNasabah.size();i++){  if(boxNasabah.get(i).getNoRek().equalsIgnoreCase(transfer)){  nasabah.transfer(boxNasabah.get(i),jumlah);  nasabah.tarik(7500);  System.out.println("Transaksi Berhasil");  }  }  ulang=false;  }break;  case 3:  System.out.println("Jumlah Saldo : " + nasabah.getSaldo());  ulang = false;  break;  case 0:  System.out.println("Program Berhenti");  sistem = false;  ulang = false;  break;  default:  System.out.println("Maaf Pilihan Tidak Tersedia");  break;  }  }  }if(status){  System.out.println("Nomor Rekening Dan PIN Yang Anda Masukkan Salah");  } }else System.out.println(“Saldo Rp.10.000”);  }break;  }  }  } |
|  | Rekening.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 | /\*\*  \* Created by Zain Fikri H on 01/04/2017.  \*/  public class Rekening {  private long saldo;  private String nama;  private String noRek;  private String pin;  public long getSaldo() {  return saldo;  }  public void setor(double jumlah){  saldo += jumlah;  }  public void tarik(double jumlah){  saldo -= jumlah;  }  public void transfer(Rekening tujuan, double jumlah){  tarik(jumlah);  tujuan.setor(jumlah);  }  public Rekening(String nama, String noRek, String pin, long jumlah) {  this.nama = nama;  this.noRek = noRek;  this.pin = pin;  this.saldo = jumlah;  }  public String getNama() {  return nama;  }  public String getNoRek() {  return noRek;  }  public String getPin() {  return pin;  }  } |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
|  | Atm.java |
| 1  2  7  8  10  12  14  16  17  18  19  21  23  25  26  27  28  29  30  31  32  37  38  39  40  41  42  43  44  45  46  47  49  50  51  52  54  56  58  59  62  63  65  66  67  68  70  71  72  73  74  75  77  79  80  82  83  86  88  89  90  92  94  96  98  100  102  103  104  105  107  109  110  111  113  115  117  119  121  123  124  126  127  130  132  133  134  136  138  140  142  144  147  148  149  151  153  155  157  158  161  164  166  169  170  171  174  177  179  181  184  185  186  189  192  194  196  199  200  201  202  204  205  206  208  209  210  211  212  213  215  219  220  221  223 | Import ArrayList  Import Scanner  Deklarasi class dengan nama Atm  Instansiasi Rekening dengan nama boxNasabah dengan ArrayList dan modifier public  Deklarasi method setNasabah bertipe void dan berparameter  Instansiasi Rekening dengan nama object rekening berkonstraktor  Menambahkan data rekening ke ArrayList  Deklarasi main method  Deklarasi scanner  Instansiasi Atm dengan nama object list  Pemanggilan method setNasabah berparameter  Pemanggilan method setNasabah berparameter  Pemanggilan method setNasabah berparameter  Print “Masukkan Nomor Rekening : “  Input user dengan variabel noRekening  Print “Masukkan PIN : “  Input user dengan variabel pin  Seleksi kondisi if dengan kondisi jika panjang noRekening sama dengan 15  Pemanggilan method display berparameter  Seleksi kondisi else  Print “Nomor Rekening Dan PIN Yang Anda Masukkan Salah”  Deklarasi method display berparameter  Instansiasi scanner  Penggunaan substring untuk variabel noRekening  Deklarasi dan inisialisasi variabel cek  Deklarasi dan inisialisasi variabel limit  Seleksi kondisi if dengan kondisi variabel cekRek sama dengan 001  Variabel cek bernilai 1  Seleksi kondisi if dengan kondisi variabel cekRek sama dengan 192  Variabel cek bernilai 2  Seleksi kondisi if dengan kondisi variabel cekRek sama dengan 283  Variabel cek bernilai 3  Deklarasi dan inisialisasi variabel status bernilai true  Perulangan for dengan kondisi sesuai object boxNasabah dengan varibel nasabah  Deklarasi dan inisialisasi variabel status bernilai true  Perulangan while dengan kondisi sesuai variabel system  Seleksi kondisi dengan kondisi getNoRek sama dengan noRekening dan getPin sama dengan pin  Seleksi kondisi jika getSaldo sama dengan 10000 dan Variabel status bernilai false  Print “==================================”  Print “ Menu”  Print “==================================”  Print “1. Penarikan Tunai”  Print “2. Transfer”  Print “3. Cek Saldo”  Print “0. Keluar”  Print “Masukkan Pilihan Anda : “  Deklarasi variabel pilihan dengan inputan user  Deklarasi variabel ulang dengan nilai true  Perulangan while dengan kondisi variabel ulang  Seleksi kondisi dengan kondisi pilihan  Case 1  Print “Masukkan Jumlah Nominal Penarikan : “  Deklarasi variabel tarik dengan inputan user  Seleksi kondisi dengan kondisi cek sama dengan 1  Seleksi kondisi dengan kondisi variabel limit kurang dari sama dengan 10000000  Seleksi kondisi dengan kondisi variabel tarik kurang dari sama dengan 10000000  Variabel limit ditambah variabel tarik  Pemanggilan method tarik dengan parameter variabel tarik  Print “Transaksi Berhasil”  Variabel ulang bernilai false  Seleksi kondisi else  Print “Maaf jumlah penarikan Anda melebihi batas”  Variabel ulang bernilai false  Seleksi kondisi else  Print “Maaf Anda sudah mencapai batas penarikan”  Variabel ualang bernilai false  Seleksi kondisi if dengan kondisi variabel cek sama dengan 2  Seleksi kondisi if dengan kondisi variabel limit kurang dari sama dengan 7000000  Seleksi kondisi if dengan kondisi variabel tarik kurang dari sama dengan 7000000  Variabel limit ditambah variabel tarik  Pemanggilan method tarik dengan parameter variabel tarik  Print “Transaksi Berhasil”  Variabel ulang bernilai false  Seleksi kondisi else  Print “Maaf jumlah penarikan Anda melebihi batas”  Variabel ulang bernilai false  Seleksi kondisi else  Print “Maaf Anda sudah mencapai batas penarikan”  Variabel ualang bernilai false  Seleksi kondisi if dengan kondisi variabel cek sama dengan 3  Seleksi kondisi if dengan kondisi variabel limit kurang dari sama dengan 5000000  Seleksi kondisi if dengan kondisi variabel tarik kurang dari sama dengan 5000000  Variabel limit ditambah variabel tarik  Pemanggilan method tarik dengan parameter variabel tarik  Print “Transaksi Berhasil”  Variabel ulang bernilai false  Seleksi kondisi else  Print “Maaf jumlah penarikan Anda melebihi batas”  Variabel ulang bernilai false  Seleksi kondisi else  Print “Maaf Anda sudah mencapai batas penarikan”  Variabel ualang bernilai false  Statement break  Case 2  Print “Masukkan Nomor Rekening Tujuan : “  Deklarasi variabel transfer dan inputan user  Print “Masukkan Nominal Transfer : “  Deklarasi variabel jumlah dan inputan user  Seleski kondisi if jika variabel cek sama dengan 1  Perulangan for dengan kondisi sesuai panjang array boxNasabah  Seleksi kondisi jika getNoRek sama dengan transfer  Pemanggilan method transfer dengan parameter  Print “Transaksi Berhasil”  Variabel ulang bernilai false  Seleski kondisi if jika variabel cek sama dengan 2  Perulangan for dengan kondisi sesuai panjang array boxNasabah  Seleksi kondisi jika getNoRek sama dengan transfer  Pemanggilan method transfer dengan parameter  Pemanggilan method tarik dengan parameter  Print “Transaksi Berhasi”  Variabel ulang bernilai false  Seleski kondisi if jika variabel cek sama dengan 3  Perulangan for dengan kondisi sesuai panjang array boxNasabah  Seleksi kondisi jika getNoRek sama dengan transfer  Pemanggilan method transfer dengan parameter  Pemanggilan method tarik dengan parameter  Print “Transaksi Berhasi”  Variabel ulang bernilai false  Statement break  Case 3  Print “Jumlah Saldo : “ dan cetak nilai dari nasabah.getSaldo  Variabel ulang bernilai false  Statement break  Case 0  Print “Program Berhenti”  Variabel system bernilai false  Variabel ulang bernilai false  Statement break  Case default  Print “Maaf Pilihan Tidak Tersedia”  Statement break  Seleksi kondisi if dengan kondisi sesuai variabel status  Print “Nomor Rekening Dan PIN Yang Anda Masukkan Salah”  Seleksi kondisi else dan print “Saldo Rp. 10.000”  Statement break |
|  | Rekening.java |
| 4  5  6  7  8  10  11  13  14  16  17  19  20  21  23  25  26  27  28  30  31  33  34  36  37 | Deklrasi class dengan nama Rekening  Deklarasi attribute saldo dengan tipe long modifier private  Deklarasi attribute nama dengan tipe String modifier private  Deklarasi noRek saldo dengan tipe String modifier private  Deklarasi noRek pin dengan tipe String modifier private  Deklarasi method getSaldo bertipe long  Statement return sesuai variabel saldo  Deklarasi method void setor berparameter  Variabel saldo ditambah variabel jumlah  Deklarasi method void tarik berparameter  Variabel saldo dikurangi variabel jumlah  Deklarasi method void transfer berparameter  Pemanggilan metod tarik dengan kondisi sesuai variabel jumlah  Pemanngilan method setor dengan kondisi sesuai variabel jumlah  Deklarasi konstruktor  Attribute nama sama dengan variabel nama  Attribute noRek sama dengan variabel noRek  Attribute pin sama dengan variabel pin  Attribute saldo sama dengan variabel jumlah  Deklarasi method getNama  Statement return sesuai attribute nama  Deklarasi method getNoRek  Statement return sesuai attribute noRek  Deklarasi method getPin  Statement return sesuai attribute pin |

1. **SCREENSHOT PROGRAM**

****

****

1. **PRAKTIKUM**
2. **Encapsulation 1**
3. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

*Jawaban:*

Terdapat kesalahan. Dan ini yang benar,

public class Student {

private String name;

private int mark;

public void setName(String n){

name=n;

}

public String getName(){

return name;

}

public void setMark(int m){

mark=m;

}

public int getMark(){

return mark;

}

}

public class Test {

public static void main(String [] args) {

Student s1=new Student();

s1.setName("Enkapsulasi");

s1.setMark(90);

System.out.println("s1Name is "+s1.getName());

System.out.println("s1Mark is "+s1.getMark());

System.out.println("name dan mark "+s1.getName()+" "+s1.getMark());

}

}

1. Jika pada baris 6 s1.setName diubah menjadi s1.getName apa yang terjadi? jelaskan!

*Jawaban:*

Jika pada baris 6 diubah menjadi s1.getName maka akan mengambil nilai dati Name yaitu “Enkapsulasi” sesuai yang di s1.setName.

1. Setelah diperbaiki, ubahlah hak akses pada baris 4 (pada class Student) menjadi private apa yang terjadi jika class Test dijalankan? Jelaskan!

*Jawaban:*

Jika pada baris 4 pada class Student modifier diganti private maka akan terjadi error pada class Test yaitu pada bagian s1.setName. Karena jika modifier diubah ke private maka artinya method tersebut hanya dapat diakses pada class itu sendiri.

1. Jika kedua kelas diatas terdapat dalam package yang sama apakah konsep enkapsulasi tetap berfungsi? jelaskan!

*Jawaban:*

Jika kedua class pada suatu package yang sama maka konsep enkapsulasi masih tetap berfungsi. Karena modifier private itu hanya dapat diakses pada class itu sendiri dan tidak dapat diakses pada class lain meskipun itu pada satu package.

1. **Instance Method**
2. Method apakah yang menjadi accessor (getter) ?

*Jawaban:*

public double getLoad(){

return this.load;

}

public double getMaxLoad(){

return this.maxLoad;

}

1. Tambahkan source code berikut dibawah baris ke 6 pada class TestVehicle1.

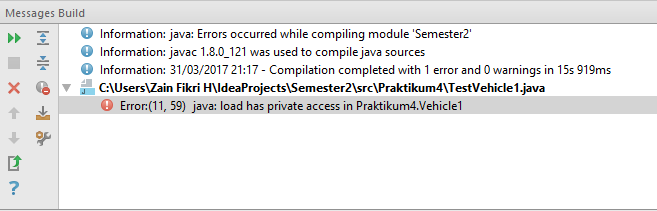
System.out.println("Add load(100kg) : " + (vehicle.load=500));

Jalankan program, apakah output dari program tersebut?

Kembalikan program seperti semula.

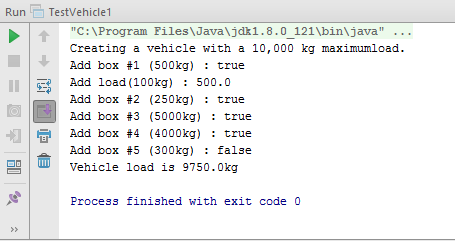
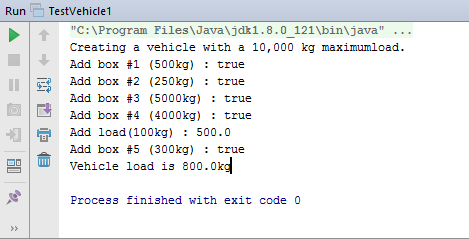
*Jawaban:*

Output dari program adalah error. Karena pada class Vehicle belum dibuat method load. Jadi object vehicle tidak dapat memanggil method load. Kemudian ada varriabel load yang bersifat private pada class Vehicle yang artinya tidak dapat diakses pada class lainnya.

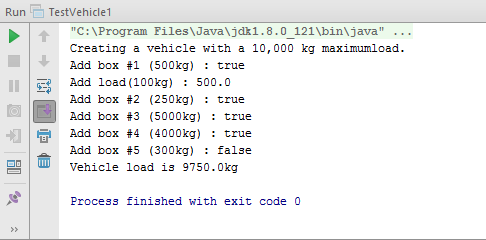
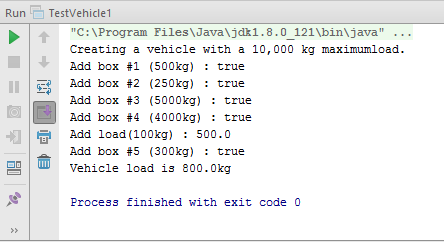


1. Ubahlah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi public. Jalankan program, apakah output dari program tersebut?
   1. Tambahkan source kode berikut dibawah baris ke 6 pada class TestVehicle1. System.out.println("Add load(100kg) : " + (vehicle.load=500)); Jalankan program, apakah output dari program tersebut? Kembalikan program seperti semula.
   2. Tambahkan source kode berikut dibawah baris ke 12 pada class TestVehicle1. System.out.println("Add load(100kg) : " + (vehicle.load=500)); Jalankan program, apakah output dari program tersebut? Kembalikan program seperti semula

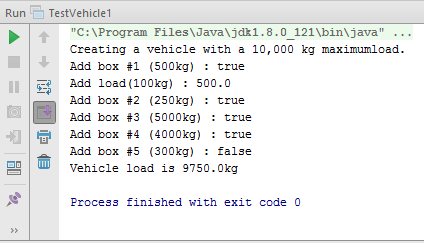
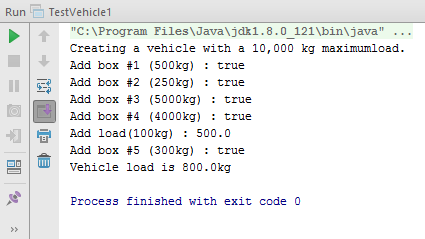
*Jawaban:*

1. 
2. 
3. Ulangi instruksi pada nomer 4 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi protected.

*Jawaban:*

1. 
2. 
3. Ulangi instruksi pada nomer 4 dengan mengubah tipe data pada atribut load dan maxload pada class Vehicle1 menjadi default.

*Jawaban:*

1. 
2. 
3. **KESIMPULAN**

* Encapsulation merupakan salah satu cara untuk menyembunyikan informasi. Encapsulation dapat dilakukan dengan mengatur acces modifier menjadi private. Dengan cara membuat data/attribute di dalam class dibuat menjadi private, kemudian beri method accessor (getter) dan mutator (setter).
* Acces modifier merupakan sebuah indicator akses yang berfungsi untuk memberikan batas akses. Dimana acces ini nanti yang berfungsi untuk melindungi suatu data, terutama untuk konsep enkapsulasi.
* Macam-macam acces modifier :

1. Private, dapat diakses hanya pada class tersebut.
2. Default, dapat diakses pada class tersebut dan satu package.
3. Protected, dapat diakses pada class tersebut, package, dan subclass.
4. Public, dapat diakses pada class tersebut, package, subclass, dan root/network.