1. **DEFINISI MASALAH**

Buatlah implementasi sebuah mesin ATM. Dalam mesinATM, password dapat dimasukkan oleh user. Selain itu, sistem memilki pilihan menu melihat saldo, menarik uang, dan mentransfer ke rekening lain!

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 | package Modul1\_Bab2;  import java.util.Scanner;  import java.text.SimpleDateFormat;  import java.util.Calendar;  public final class ATM {  Calendar cal = Calendar.getInstance();  SimpleDateFormat date = new SimpleDateFormat("dd/MM/yy HH:mm");  String date1 = date.format(cal.getTime());  Scanner input = new Scanner(System.in);  private static final String nameOfBank = "BANK KRUT";  private static long totalCustomer = 1234567890;  private final long customerId = ++totalCustomer;  private String name;  private int pin;  private double balance, choiceBalance;  private long choiceId;  ATM() {  System.out.print("--------------------------------\n"  + "\n\tMENU PENDAFTARAN AKUN BARU\n"  + "\t MASUKKAN DATA ANDA\n\n"  + " NAMA \t\t: ");  this.name = inputStringLine();  System.out.print(" PIN \t\t: ");  this.pin = input.nextInt();  System.out.print(" SETORAN TUNAI \t: ");  this.balance = input.nextDouble();  System.out.println("");  displayCustomer();  }  void displayCustomer() {  System.out.println("--------------------------------\n"  + "\t BANK KRUT\n"  + " BUKTI STRUK PENDAFTARAN ANDA\n"  + "\t " + date1  + "\n\n BANK \t\t: " + nameOfBank  + "\n NO REKENING \t: " + customerId  + "\n NAMA \t\t: " + name.toUpperCase()  + "\n PIN \t\t: " + pin  + "\n SALDO \t\t: " + balance  + "\n\n\t HARAP JAGA KERAHASIAAN"  + "\n\t\t PIN ANDA");  }  void atm() {  System.out.print("-------------------------------\n\n"  + "\t PILIH TRANSAKSI YANG\n"  + "\t ANDA INGINKAN\n"  + "\n 1 <- GANTI PIN PENARIKAN TUNAI -> 3"  + "\n 2 <- TRANSFER INFORMASI SALDO -> 4"  + "\n\n 0 <- MENU SEBELUMNYA"  + "\n\n\t\tPILIHAN : ");  int choice = input.nextInt();  System.out.println("");  switch (choice) {  case 1:  setPin();  break;  case 2:  transfer();  break;  case 3:  cashWithdrawal();  break;  case 4:  checkBalance();  break;  case 0:;  break;  default:  System.out.println("PILIHAN ANDA TIDAK TERSEDIA");  }  }  void setPin() {  System.out.print("-------------------------------\n\n"  + "\t\t\tMASUKKAN\n\t\t\tPIN\n\t\t\tBARU ANDA\n\n\t\t\t");  this.pin = input.nextInt();  System.out.println("\n\n\t PIN SUKSES DIGANTI\n\n");  }  void transfer() {  System.out.print("----------------------------------\n"  + "\n\t PILIH TUJUAN TRANSFER\n"  + "\t YANG ANDA INGINKAN\n"  + "\n\n\t\t KE REKENING SESAMA -> 1"  + "\n\t KE REKENING BANK LAIN -> 2"  + "\n\n\t\tPILIHAN : ");  int choice = input.nextInt();  if (choice == 1) {  transferTo();  } else if (choice == 2) {  transferToOther();  }  System.out.println();  }  void transferTo() {  System.out.print("---------------------------------\n"  + "\n\t SILAKAN MASUKKAN NOMOR\n"  + " REKENING TUJUAN PEMINDAHBUKUAN\n\n\t\t");  choiceId = input.nextLong();  System.out.print("\n\t\t PILIH JIKA BENAR -> 1"  + "\n\t\t PILIH JIKA SALAH -> 2"  + "\n\n\t\tPILIHAN : ");  int choice = input.nextInt();  if (choice == 1) {  nominal();  } else {  transferTo();  }  }  void transferToOther() {  System.out.println("--------------------------------\n"  + "\n\t TRANSFER ANTAR BANK\n");  System.out.print(" BANK TUJUAN\t: ");  String otherBank = inputStringLine();  System.out.print(" NO REKENING\t: ");  choiceId = input.nextLong();  System.out.print(" NAMA\t\t: ");  String otherName = inputStringLine();  System.out.print(" JUMLAH\t\t: ");  choiceBalance = input.nextDouble();  System.out.print("\n\t\t PILIH JIKA BENAR -> 1"  + "\n\t\t PILIH JIKA SALAH -> 2"  + "\n\n\t\tPILIHAN : ");  int choice = input.nextInt();  if (choice == 1 && choiceBalance <= balance) {  System.out.println("----------------------------\n"  + "\t INFORMASI PEMINDAHBUKUAN"  + "\n\t " + date1  + "\n\n DARI BANK\t: " + nameOfBank  + "\n NO REKENING\t: " + customerId  + "\n NAMA\t\t: " + name  + "\n KE BANK\t: " + otherBank  + "\n NO REKENING\t: " + choiceId  + "\n NAMA\t\t: " + otherName  + "\n JUMLAH\t\t: " + choiceBalance  + "\n\t TRANSAKSI DIPROSES");  balance -= choiceBalance;  } else if (choice == 1 && choiceBalance > balance) {  System.out.println(" MAAF SALDO ANDA TIDAK MENCUKUPI");  } else {  transferToOther();  }  }  void nominal() {  System.out.print("---------------------------------\n"  + "\n MASUKKAN JUMLAH UANG YANG AKAN\n"  + "\t DIPINDAHBUKUKAN\n\n\t\t");  choiceBalance = input.nextDouble();  System.out.print("\n\t\t PILIH JIKA BENAR -> 1"  + "\n\t\t PILIH JIKA SALAH -> 2"  + "\n\n\t\tPILIHAN : ");  int choice = input.nextInt();  if (choice == 1 && choiceBalance <= balance) {  } else if (choice == 1 && choiceBalance > balance) {  System.out.println(" MAAF SALDO ANDA TIDAK MENCUKUPI\n");  } else {  nominal();  }  }  void cashWithdrawal() {  System.out.print("---------------------------------\n"  + "\n SILAKAN MEMASUKKAN JUMLAH UANG"  + "\n\t YANG DIINGINKAN\n\n\t\t");  choiceBalance = input.nextDouble();  System.out.print("\n\t\t PILIH JIKA BENAR -> 1"  + "\n\t\t PILIH JIKA SALAH -> 2"  + "\n\n\t\tPILIHAN : ");  int choice = input.nextInt();  if (choice == 1 && choiceBalance <= balance) {  balance -= choiceBalance;  System.out.println("---------------------------\n"  + "\t BANK KRUT\n"  + "\tBUKTI STRUK PENARIKAN TUNAI\n"  + "\t " + date1  + "\n\n NO REKENING\t: " + customerId  + "\n NAMA\t\t: " + name  + "\n JUMLAH\t\t: RP " + choiceBalance  + "\n SISA SALDO\t: RP " + balance  + "\n\n\t TINGKATKAN TRANSAKSI ANDA\n");  } else if (choice == 1 && choiceBalance > balance) {  System.out.println(" MAAF SALDO ANDA TIDAK MENCUKUPI\n");  } else {  cashWithdrawal();  }  choiceId = 0;  }  void checkBalance() {  System.out.println("-------------------------------\n"  + "\t BANK KRUT\n"  + "\t INFORMASI SALDO ANDA\n"  + "\t " + date1  + "\n\n BANK \t\t: " + nameOfBank  + "\n NO REKENING \t: " + customerId  + "\n NAMA \t\t: " + name.toUpperCase()  + "\n SALDO \t\t: " + balance  + "\n\n\t HARAP JAGA KERAHASIAAN"  + "\n\t\t PIN ANDA");  choiceId = 0;  }  public int getPin() {  return pin;  }  public void setBalance(double balance) {  this.balance = balance;  }  public double getBalance() {  return balance;  }  public long getCustomerId() {  return customerId;  }  public String getName() {  return name;  }  public double getChoiceBalance() {  return choiceBalance;  }  public long getChoiceId() {  return choiceId;  }  public static String inputStringLine() {  Scanner input = new Scanner(System.in);  return input.nextLine();  }  } |

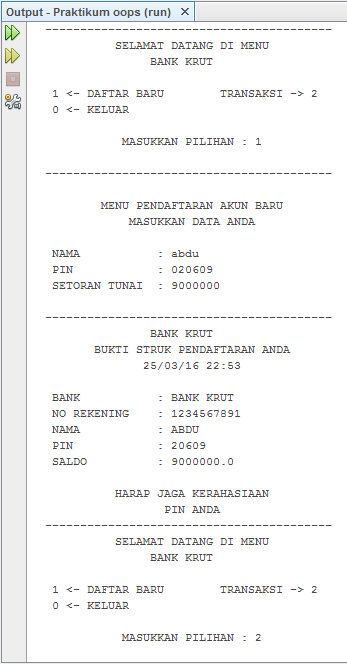
|  |  |
| --- | --- |
| **MainATM.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64 | package Modul1\_Bab2;  import java.util.Scanner;  public class MainATM {  public static void main(String[] args) {  Scanner input = new Scanner(System.in);  ATM atm[] = new ATM[100];  int choice, x = 0, pin;  do {  System.out.print("-------------------------------"  + "\n\t SELAMAT DATANG DI MENU\n"  + "\t BANK KRUT\n\n"  + " 1 <- DAFTAR BARU \t TRANSAKSI -> 2\n"  + " 0 <- KELUAR\n\n"  + "\t MASUKKAN PILIHAN : ");  choice = input.nextInt();  System.out.println();  switch (choice) {  case 1:  atm[x] = new ATM();  x++;  break;  case 2:  double send = 0,  from = 0;  for (int i = 0; i < x; i++) {  System.out.print("------------------\n\n"  + "\t\t\tSILAHKAN \n\t\t\tMEMASUKKAN\n"  + "\t\t\tPIN ANDA \n\n\t\t\t");  pin = input.nextInt();  if (pin == atm[i].getPin()) {  System.out.println("\n\t\t\tPIN SUKSES!\n");  atm[i].atm();  for (int j = 0; j < x; j++) {  if (atm[i].getChoiceId() == atm[j].getCustomerId()) {  System.out.println("---------------------------\n"  + "\n\t INFORMASI PEMINDAHBUKUAN\n"  + "\n DARI\t\t: " + atm[i].getCustomerId()  + "\n NAMA\t\t: " + atm[i].getName()  + "\n KE\t\t: " + atm[i].getChoiceId()  + "\n NAMA\t\t: " + atm[j].getName()  + "\n JUMLAH\t\t: " + atm[i].getChoiceBalance()  + "\n\n\t TRANSAKSI DIPROSES\n");  send = atm[i].getBalance() - atm[i].getChoiceBalance();  from = atm[j].getBalance() + atm[i].getChoiceBalance();  atm[j].setBalance(from);  atm[i].setBalance(send);  }  }  } else {  System.out.println("\n\n\t MAAF PIN ANDA SALAH!!\n\n");  }  i = x;  }  break;  case 0:  System.out.println("------------------------");  System.err.println("\t Terima Kasih!");  break;  default:  System.out.println(" PILIHAN ANDA TIDAK TERSEDIA");  }  } while (choice != 0);  }  } |

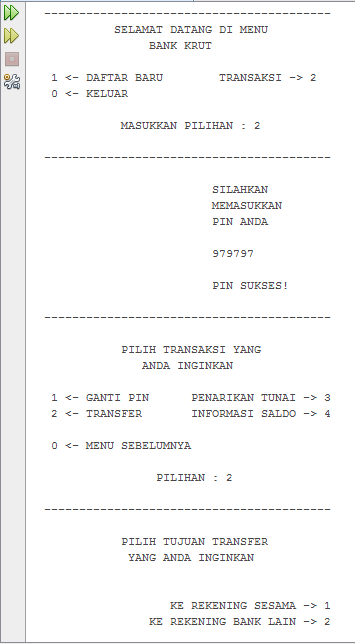
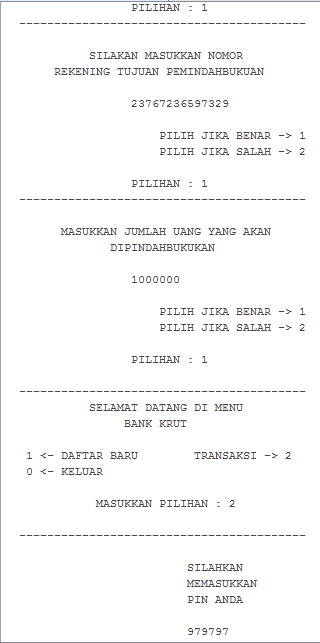
1. **PEMBAHASAN**

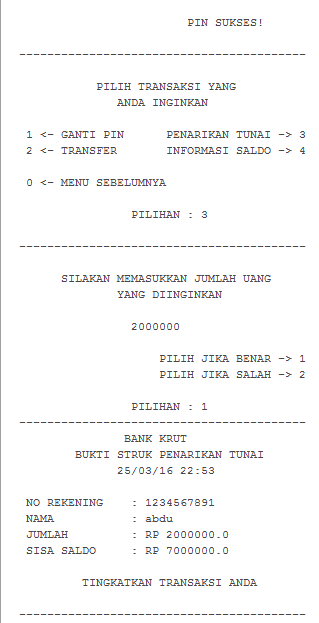
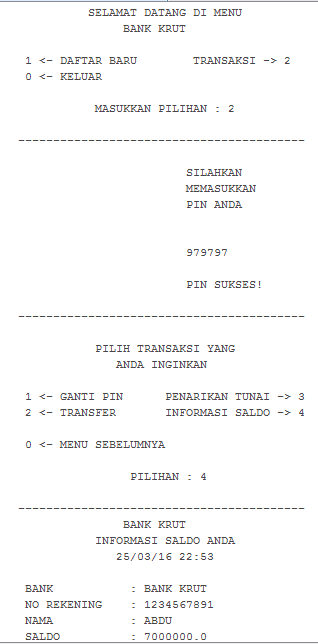
|  |  |
| --- | --- |
| **ATM.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  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 | Meletakkan program pada package Modul\_Bab2  Melakukan import Scanner pada library java  Melakukan import java.text.SimpleDateFormat untuk mengubah tampilan tanggal  Melakukan import java.util.Calendar untuk mendapatkan tanggal sekarang  Deklarasi class Customer  Instansiasi cal untuk mendapatkan tanggal  Instansiasi date untuk mendapatkan format tanggal dd/MM/yy HH:mm  Deklarasi date1 sebagai variabel untuk menadapatkan tanggal yang sudah di set  Instansiasi input sebagai pemanggilan Scanner  deklarasi nameOfBank yang menyimpan nama Bank (BANK KRUT)  deklarasi totalCustomer mengeset agar no rekening user di mulai setelah 1234567890  deklarasi customerId dimana setiap object akan mendapatkan no rek. yang berbeda  deklarasi name untuk menyimpan nama nasabah yang diinputkan  deklarasi pin untuk menyimpan pin nasasbah  deklarasi balance dan choiceBalance digunakan untuk menyimpan jumlah saldo  deklarasi choiceId menyimpan no rekening inputan sementara  constructor Customer()  mencetak ------------------------------------------  MENU PENDAFTARAN AKUN BARU  MASUKKAN DATA ANDA  NAMA :  Inputan user (nama)  PIN :  Inputan user (pin)  SETORAN TUNAI :  Inputan user (saldo yang disetorkan awal)  Mencetak enter  Memanggil method displayCustomer() untuk menampilkan hasil inputan  }  method displayCustomer()  mencetak ------------------------------------------  BANK KRUT  BUKTI STRUK PENDAFTARAN ANDA  Mencetak tanggal dan jam  BANK : (nama bank)  NO REKENING : (no rekening)  NAMA : (nama yang mendaftar)  PIN : (pin ATM)  SALDO : (saldo)  HARAP JAGA KERAHASIAAN  PIN ANDA  }  Method atm()  Mencetak ------------------------------------------  PILIH TRANSAKSI YANG  ANDA INGINKAN  1 <- GANTI PIN PENARIKAN TUNAI -> 3  2 <- TRANSFER INFORMASI SALDO -> 4  0 <- MENU SEBELUMNYA  PILIHAN :  Menerima pilihan user  Mencetak baris jeda  Kondisi pilihan user  Jika memilih 1  Maka kan memanggil method setPin()  Digunakan agar tidak lanjut ke kondisi berikutnya  Jika memilih 2  Maka kan memanggil method transfer()  Digunakan agar tidak lanjut ke kondisi berikutnya  Jika memilih 2  Maka kan memanggil method cashWithdrawal()  Digunakan agar tidak lanjut ke kondisi berikutnya  Jika memilih 4  Maka kan memanggil method checkBalance()  Digunakan agar tidak lanjut ke kondisi berikutnya  Jika memilih 0  Maka akan keluar dari program  Jika masukan pilihan salah  mencetak PILIHAN ANDA TIDAK TERSEDIA  }  }  method setPin()  mencetak ------------------------------------------  MASUKKAN PIN BARU ANDA  Masukkan pin baru  mencetak PIN SUKSES DIGANTI  }  method transfer()  mencetak ------------------------------------------  PILIH TUJUAN TRANSFER  YANG ANDA INGINKAN  KE REKENING SESAMA -> 1  KE REKENING BANK LAIN -> 2  PILIHAN :  variabel choice menerima inputan user  jika inputan user = 1 maka  memanggil method transferTo();  jika = 2 maka  memanggil method transferToOther();  }  Mencetak untuk membuat line baru  }  method transferTo()  mencetak ------------------------------------------  SILAHKAN MASUKKAN NOMOR  REKENING TUJUAN PEMINDAHBUKUAN  Variabel choiceId menerima inputan rekening dari user  Mencetak PILIH JIKA BENAR -> 1  PILIH JIKA SALAH -> 2"  PILIHAN :  variabel choice menerima inputan user  jika pilihan = 1 maka  memanggil method nominal()  jika tidak maka  memangil methot transferTo() itu sendiri, agar mengulangi lagi  }  }  Method transferToOther()  Mencetak ------------------------------------------  Mencetak TRANSFER ANTAR BANK  Mencetak BANK TUJUAN :  User diminta untuk menginputkan nama bank tujuan  Mencetak NO REKENING :  User diminta untuk menginputkan no rekening tujuan  Mencetak NAMA :  User diminta untuk menginputkan nama penerima  Mencetak JUMLAH :  User diminta untuk menginputkan jumlah yang ingin di transfer  Mencetak PILIH JIKA BENAR -> 1  PILIH JIKA SALAH -> 2  PILIHAN :  User diminta memasukkan pilihan  Jika pilihan =1 dan jumlah dibawah atau sama dengan saldo maka  Mencetak ------------------------------------------  INFORMASI PEMINDAHBUKUAN  Mencetak tanggal saat ini  Mencetak DARI BANK : (nama bank pengirim)  NO REKENING : (no rek. pengirim)  NAMA: (nama pengirim)  KE BANK: (nama bank penerima)  NO REKENING: (no. rek. penerima)  NAMA: (nama penerima)  JUMLAH: (jumlah yang di transfer)  TRANSAKSI DIPROSES  Saldo dikurangi jumlah yang ditransfer  Jika pilihan = 1 dan jumlah melebihi saldo maka  Mencetak MAAF SALDO ANDA TIDAK MENCUKUPI  Namun jika semua kondisi sebelumnya tidak terpenuhi maka  Memanggil method transferToOther()  }  }  method nominal()  Mencetak ------------------------------------------  MASUKKAN JUMLAH UANG YANG AKAN  DIPINDAHBUKUKAN  User diminta mengimputkan nominal yang akan di transfer  Mencetak PILIH JIKA BENAR -> 1  PILIH JIKA SALAH -> 2  PILIHAN :  User diminta untuk memasukkan pilihan  Jika memilih 1 dan nominal dibawah saldo maka akan keluar dari method  jika memilih 1 dan nominal melebihi saldo maka  mencetak MAAF SALDO ANDA TIDAK MENCUKUPI  namun jika semua kondisi tidak terpenuhi maka  memanggil method nominal()  }  }  Method cashWithdrawal()  Mencetak ------------------------------------------  SILAHKAN MEMASUKKAN JUMLAH UANG  YANG DIINGINKAN  User diminta menginputkan nominal uang yang akan di tarik tunai  Mencetak PILIH JIKA BENAR -> 1  PILIH JIKA SALAH -> 2  PILIHAN :  User diminta memasukkan pilihan  Jika memilih 1 dan nominal dibawah saldo  Maka saldo dikurangi nominal  Mencetak ------------------------------------------  BANK KRUT  BUKTI STRUK PENARIKAN TUNAI  Mencetak tanggal saat ini  NO REKENING: (no rekening)  NAMA: (nama)  JUMLAH: (jumlah nominal yang di ambil)  SISA SALDO: (mencetak saldo yang sudah dikurangi nominal yang diambil)  TINGKATKAN TRANSAKSI ANDA  Jika pilihan = 1 namun nominal melebihi saldi maka  mencetak MAAF SALDO ANDA TIDAK MENCUKUPI  jika kondisi sebelumnya tidak ada yang sesuai maka  akan memanggil method cashWithdrawal()  }  Masukan rekening yg salah di hapus (0)  }  method checkBalance()  mencetak -----------------------------------------  BANK KRUT  INFORMASI SALDO ANDA  Menampilkan tanggal saat ini  mencetak BANK : (nama bank)  NO REKENING: (no rekening)  NAMA : (nama)  SALDO : (saldo)  HARAP JAGA KERAHASIAAN  PIN ANDA  Inputan rekening user di hapus  }  method getPin() digunakan agar field pin bisa di akses  mengembalikan nilai pin  }  method setBalance(double balance) digunakan untuk mengedit field balance  isi parameter balance dicopykan ke field balance  }  method getBalance()digunakan agar mendapatkan nilai field balance  mengembalikan nilai balance  }  method getCustomerId()digunakan agar mendapatkan nilai field costumerId  mengembalikan nilai costumerId  }  Method getName() digunakan agar mendapatkan nilai field name  Mengembalikan nilai name  }  method getChoiceBalance() digunakan agar mendapatkan nilai field choiceBalance  mengembalikan nilai choiceBalance  }  Method getChoiceId()digunakan agar mendapatkan nilai field choiceId  Mengembalikan nilai choiceId  }  method inputStringLine() digunakan agar tidak terjadi skip inputan user  deklarasi input sebagai Scanner untuk menerima masukkan user  mengembalikan nilai yang diimputkan user  }  } |

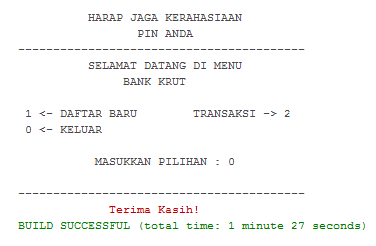
|  |  |
| --- | --- |
| MainATM.java | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64 | Meletakkan program pada package Modul1\_Bab2  Melakukan import Scanner pada library java  Deklarasi class MainATM  Deklarasi main method untuk menjalankan program  Deklarasi input sebagai scanner yang menerima masukan dari pengguna  Instansiasi array c sebagai referensi dari contructor Customer  Deklarasi variabel choise, x dan pin untuk menyimpan masukan user  Pengulangan do, digunakan agar program terus berulang sampai kondisi terpenuhi  Mencetak ------------------------------------------  SELAMAT DATANG DI MENU  BANK KRUT  1 <- DAFTAR BARU TRANSAKSI -> 2  0 <- KELUAR  MASUKKAN PILIHAN :  Variable choice menyimpan inputan user  Tidak Mencetak Mencetak apa-apa hanya line kosong  Kondisi “switch” membuat menu pilihan  Jika memilih 1  Maka akan membuat objek baru yang memanggil constructor  X akan bertambah 1 setiap user memilih 1  Digunakan agar tidak melanjutkan ke pilihan berikutnya  Jika memilih 2  Membuat variabel baru yang bertipe double dengan nama send  Dan juga membuat variabel from  Melakukan pengulangan selama i kurang dari x (banyaknya nasabah)  Mencetak ------------------------------------------  SILAHKAN MEMASUKKAN  PIN ANDA  Variabel pin menyimpan inputan user  Kondisi jika pin sama maka akan mencetak  PIN SUKSES!  Memanggil method atm() yang ada di object terpilih  Pengulangan untuk mencari ID yang sama dengan inputan user  Jika ID sama dengan ID di salah satu array  Mencetak ------------------------------------------  INFORMASI PEMINDAHBUKUAN  DARI : (no rekening pengirim)  NAMA : (nama pengirim)  KE : (no rekening penerima)  NAMA : (nama penerima)  JUMLAH : (nominal uang yang di kirim)  TRANSAKSI DIPROSES  variabel send menyimpan hasil dari pengurangan saldo pengirim  variabel from menyimpan hasil dari penambahan saldo penerima  saldo pengirim di rubah menjadi hasil yang sudah di kurangi  saldo penerima di rubah menjadi hasil yang sudah di tambah  }  }  Jika pin tidak sama maka akan mencetak  MAAF PIN ANDA SALAH!!  }  Dan pengulangan dihentikan dengan cara menyamakan i dan x  }  Menghentikan kondisi  Pilihan 0 akan kembali ke pengulangan shingga kembali ke menu awal  Mencetak ------------------------------------------  Terima Kasih!  Menghentikan kondisi  Jika inputan user diluar pilihan maka akan mencetak  PILIHAN ANDA TIDAK TERSEDIA  }  } while (choice != 0);  }  } |

1. **SCREENSHOT PROGRAM**









1. **PRAKTIKUM**
2. **Constructor**

*Pertanyaan*

1. Lakukan percobaan constructor diatas dan benahi jika menemukan kesalahan!

***Jawab:*** Pada baris ke-9 dari class Student dan baris ke-24 dari class MainStudent di atas terdapat sebuah kesalahan dalam penulisan nama dari sebuah constructor. Kesalahan tersebut meliputi penulisan huruf awal dari nama constructor yang menggunakan huruf kecil.

|  |  |
| --- | --- |
| **Student.java** | |
| 9 | Public **S**tudent () { |

|  |  |
| --- | --- |
| **MainStudent.java** | |
| 24 | Anna = new **S**tudent (“anna”, “Batu”, 18); |

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 mathGrade, double englishGrade, double scienceGrade) {  this.mathGrade = mathGrade;  this.englishGrade = englishGrade;  this.scienceGrade = scienceGrade;  jumlahMethod++;  } |

|  |
| --- |
| **MainStudent.java** |
| *Student anna = new Student();*  anna = new Student(100,89,80); |

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 adakah remidi. Nilai yang di cari adalah nilai rata rata untuk semua mapel. Kemudian nilai pada method statusAkhir tampilkan pada method displayMessage!

***Jawab:***

|  |
| --- |
| **Student.java** |
| boolean statusAkhir(){  return getAverage() >= 61;  }  public void displayMessage() {  System.out.println("lolos ? " + statusAkhir());  } |

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

***Jawab:***

|  |
| --- |
| **MainStudent.java** |
| import java.util.Scanner;  *public class MainStudent {*  *public static void main(String[] args) {*  Scanner input = new Scanner(System.in);  System.out.print("Masukkan jumlah mahasiswa : ");  int jumlah = input.nextInt();  Student[] dataMahasiswa = new Student[jumlah];  for (int i = 0; i < jumlah; i++) {  dataMahasiswa[i] = new Student();  dataMahasiswa[i].jumlahObjek();  System.out.print("Masukkan nama mahasiswa : ");  dataMahasiswa[i].setName(input.next());  System.out.print("Masukkan nama alamat : ");  dataMahasiswa[i].setAddress(input.next());  System.out.print("Masukkan umur anda : ");  dataMahasiswa[i].setAge(input.nextInt());  ... |

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:*** Yaitu dengan cara menambah variable static, yang digunakan untuk menyimpan jumlah object yang dibuat, di setiap membuat object baru maka variable static tersebut ditambah 1, misalkan namanya “*jumlahMethod*”. di dalam constructor kita tambahkan “*jumlahMethod++*” sehingga bertambah terus setiap constructor itu dipanggil, dan jumlahnya ditampilkan di method *jumlahobjek().*

|  |
| --- |
| **Student.java** |
| public void jumlahObjek(){  System.out.println("Objek yang dipanggil berjumlah " + jumlahMethod);  } |

|  |
| --- |
| Tambahkan di setiap constructor:  jumlahMethod++; |

1. **Instance Method**

***Pertanyaan***

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

***Jawab:*** Program berjalan dengan baik, tidak ditemukan kesalahan.

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

|  |
| --- |
| **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 (; i != 0;) {  temp = A % B;  A = B;  B = temp;  } |
|  |

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

***Jawab:***

|  |
| --- |
| **Rasional.java** |
| public double min(){  return pembilang-penyebut;  }  public double times(){  return pembilang\*penyebut;  }  public double divide(){  return pembilang/penyebut;  } |
|  |

1. **KESIMPULAN**

**Constructor** adalah hampir mirip bahkan sama dengan method dimana seluruh inisialisasi object ditempatkan. Saat kita menginstan sebuah object pada main class atau class lain, kita sebenarnya telah memanggil sebuah konstruktor pada sebuah class yang kita instan objeknya.

**Perbedaan constructor dan method** adalah constructor harus memiliki nama yang sama dengan nama class-nya dan tidak memiliki nilai kembalian. sedangkan class memiliki nama yang berbeda dan bisa mengembalikan nilai.

**Cara penulisan dari sebuah constructor**

|  |
| --- |
| [Modifier][NamaClass]([TypeVarParameter][NamaParameter],,, *dan seterusnya*){     Perintah; } |
|  |

**Method instance** adalah metode yang dikaitkan dengan instan class tertentu. Sewaktu metode instant dipanggil, ia mengakses ke data yang terdapat pada instan yang dikaitkan.

Penjelasan secara singkat cara memanggil method dari sebuah penginstanan class, cara memanggilnya adalah dengan cara membuat object baru dengan perintah: NamaClass NamaObject = new NamaConstructor();