1. **DEFINISI MASALAH**
2. Susun program dengan menggunakan overloading function dengan ketentuan :

Terdapat method bernama overloadingMeth berparameter String dan integer, dimana method tersebut mempunyai fungsi untuk merubah input teks menjadi bilangan dan input bilangan menjadi teks.

Misal :

Input : overloadingMeth(71)

Output : tujuh puluh satu

Input : overloadingMeth(tiga puluh lima)

Output : 35

Range untuk input parameter adalah 0-100

1. Buatlah program dengan memanfaatkan overoading constructor dan overloading method untuk membuat Sistem Informasi Penjualan Apotek. User diharuskan untuk menginputkan nama, alamat dan pilihan daftar obat dan harga yag dimiliki oleh sistem dan pengguna di haruskan untuk memilih obat yang diinginkan. Di akhir program terdapat output berupa total harga yang harus dibayar oleh pengguna.
2. **SOURCE CODE**

**Masalah 1**

|  |  |
| --- | --- |
| **Convertor.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 | package convertor;  public class Convertor {  private String terbilang;  private int bilangan;  public void setBilangan(int bilangan) {  this.bilangan = bilangan;  }  public void setTerbilang(String terbilang) {  this.terbilang = terbilang;  }  public int getBilangan() {  return bilangan;  }  public String getTerbilang() {  return terbilang;  }  public String overloadingMeth(int bilangan) {  String[] angka = {"Nol", "Satu", "Dua", "Tiga", "Empat",  "Lima", "Enam", "Tujuh", "Delapan", "Sebilan", "Sepuluh",  "Sebelas"};  String output = "";  if (bilangan < 12) {  output = angka[bilangan];  } else if (bilangan == 20  || bilangan == 30  || bilangan == 40  || bilangan == 50  || bilangan == 60  || bilangan == 70  || bilangan == 80  || bilangan == 90)  {  output = angka[bilangan / 10] + " Puluh";  } else if (bilangan < 20) {  output = angka[bilangan - 10] + " Belas";  } else if (bilangan < 100) {  output = angka[bilangan / 10] + " Puluh " +  angka[bilangan % 10];  } else if (bilangan == 100) {  output = "Seratus";  }  return output;  }  public int overloadingMeth(String bilangan) {  int output = 0;  String temp1 = "", temp2 = "";  String[] parts = bilangan.split(" ");  for (int x = 0; x < parts.length; x++) {  if (parts[x].equals("nol")) {  temp1 = "0";  } else if (parts[x].equals("satu")) {  temp1 = "1";  } else if (parts[x].equals("belas")) {  temp1 = "1";  } else if (parts[x].equals("dua")) {  temp1 = "2";  } else if (parts[x].equals("tiga")) {  temp1 = "3";  } else if (parts[x].equals("empat")) {  temp1 = "4";  } else if (parts[x].equals("lima")) {  temp1 = "5";  } else if (parts[x].equals("enam")) {  temp1 = "6";  } else if (parts[x].equals("tujuh")) {  temp1 = "7";  } else if (parts[x].equals("delapan")) {  temp1 = "8";  } else if (parts[x].equals("sembilan")) {  temp1 = "9";  } else if (parts[x].equals("sepuluh")) {  temp1 = "10";  } else if (parts[x].equals("sebelas")) {  temp1 = "11";  } else if (parts[x].equals("seratus")) {  temp1 = "100";  } else {  temp1 = "";  }  if (parts[x].equals("puluh")) {  if (parts.length == 2) {  temp2 = temp2 + "0";  } else {  temp2 = temp2 + temp1;  }  } else if (parts[x].equals("belas")) {  temp2 = temp1 + temp2;  } else {  temp2 = temp2 + temp1;  }  }  output = Integer.parseInt(temp2);  return output;  }  public void cetak(String cetak) {  System.out.println(cetak);  }  public void cetak(int cetak) {  System.out.println(cetak);  }  } |

|  |  |
| --- | --- |
| **Main\_in.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 | package convertor;  import java.util.Scanner;  public class Main\_in {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  Scanner scan2 = new Scanner(System.in);  Convertor conv = new Convertor();  String loop = "y";  do {  cetakMenu();  switch (scan.next()) {  case "0":  System.out.println("Program Telah Selesai");  loop = "n";  break;  case "1":  System.out.print("Masukkan angka : ");  conv.setBilangan(scan.nextInt());  conv.cetak(conv.overloadingMeth(  conv.getBilangan()));  System.out.println("");  break;  case "2":  System.out.print("Masukkan terbilang : ");  conv.setTerbilang(scan2.nextLine());  conv.cetak(conv.overloadingMeth(  conv.getTerbilang()));  System.out.println("");  break;  default:  System.out.println("Menu tidak ada");  System.out.println("");  break;  }  } while (loop == "y");  }  public static void cetakMenu() {  System.out.println( "PILIH MASUKAN");  System.out.println(">>============<<\*>>============<<");  System.out.println("1. Angka ==>> Terbilang");  System.out.println("2. Terbilang ==>> Angka");  System.out.println("0. Keluar");  System.out.println(">>============<<\*>>============<<");  System.out.print("pilih : ");  }  } |

**Masalah 2**

|  |  |
| --- | --- |
| **Obat.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 | public class Obat {  private String KodeObat;  private String NamaObat;  private int HargaObat;  public void setKodeObat(String KodeObat) {  this.KodeObat = KodeObat;  }  public void setNamaObat(String NamaObat) {  this.NamaObat = NamaObat;  }  public void setHargaObat(int HargaObat) {  this.HargaObat = HargaObat;  }  public String getKodeObat() {  return KodeObat;  }  public String getNamaObat() {  return NamaObat;  }  public int getHargaObat() {  return HargaObat;  }  public static Obat[] daftarObat(Obat[] obat) {  String[] kodeObat = {"001", "002", "003", "004", "005"};  String[] namaObat = {"Albotil ", "Puyer 16",  "Entro Stop", "Neosep Forte", "Sana Flu"};  int[] hargaObat = {10000, 11000, 5000, 15000, 6000};  for (int x = 0; x < obat.length; x++) {  obat[x] = new Obat();  obat[x].setKodeObat(kodeObat[x]);  obat[x].setNamaObat(namaObat[x]);  obat[x].setHargaObat(hargaObat[x]);  }  return obat;  }  public static void cetakDaftarObat(Obat[] obat) {  System.out.println("\nDaftar Obat");  System.out.println("Kode\tNama\t\tHarga");  for (int x = 0; x < obat.length; x++) {  System.out.print(obat[x].getKodeObat() + "\t");  System.out.print(obat[x].getNamaObat() + "\t");  System.out.printf("Rp." + obat[x].getHargaObat() + ",-" +"\n");  }  }  public static void pesananObat(String[] pesanan, Obat[] obat)  {  System.out.println("\nPesanan Obat");  int total = 0;  for (int x = 0; x < pesanan.length; x++) {  for (int y = 0; y < obat.length; y++) {  if (obat[y].getKodeObat().equals(pesanan[x])) {  System.out.print(obat[y].getNamaObat() + " - ");  System.out.print("Rp." + obat[y].getHargaObat() + ",-");  total = total + obat[y].getHargaObat();  System.out.println("");  }  }  }  System.out.println("---------------- +");  System.out.println("Total = Rp." + total + ",-");  }  public static void pesananObat(String[] pesanan, Obat[] obat,  User user) {  int total = 0;  System.out.println("APOTEK SEHAT");  System.out.println("------------");  System.out.println("a.n Bpk/Ibu : "+user.getNama());  System.out.println("alamat : "+user.getAlamat());  System.out.println("------------");  System.out.println("Pembelian");  System.out.println("Kode\tNama\t\tHarga");  for (int x = 0; x < pesanan.length; x++) {  for (int y = 0; y < obat.length; y++) {  if (obat[y].getKodeObat().equals(pesanan[x])) {  System.out.print(obat[y].getKodeObat() + "\t");  System.out.print(obat[y].getNamaObat() + "\t");  System.out.printf("Rp." + obat[y].getHargaObat() +  ",-" + "\n");  total = total + obat[y].getHargaObat();  }  }  }  System.out.println("---------------------------------- +");  System.out.println("Total Rp." + total +  ",-");  }  } |

|  |  |
| --- | --- |
| **User.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 | import java.util.Scanner;  public class User {  Scanner scanNama = new Scanner(System.in);  Scanner scanAlamat = new Scanner(System.in);  private String Nama;  private String Alamat;  public User(){  this.Nama="Belum diisi";  this.Alamat="Belum diisi";  }  public User(String nama, String alamat){  this.Nama=nama;  this.Alamat=alamat;  }  public void setNama() {  System.out.print("Masukkan nama\t: ");  this.Nama = scanNama.nextLine();  }  public void setAlamat() {  System.out.print("Masukkan alamat\t: ");  this.Alamat = scanAlamat.nextLine();  }  public String getNama() {  return Nama;  }  public String getAlamat() {  return Alamat;  }  } |

|  |  |
| --- | --- |
| **MainApotek.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 | import java.util.Scanner;  public class MainApotek {  public static void main(String[] args) {  Scanner scan = new Scanner(System.in);  String[] pesanan = new String[100];  Obat[] obat = new Obat[5];  User user = new User();  String l = "y",loop = "y";  int x = 0;  do {  cetakMenu();  switch (scan.nextInt()) {  case 0:  System.out.println("Program selesai");  l = "n";  break;  case 1:  System.out.println("\nData User");  user.setNama();  user.setAlamat();  System.out.println(user.getAlamat());  System.out.println(user.getNama());  System.out.println("");  break;  case 2:  obat = Obat.daftarObat(obat);  Obat.cetakDaftarObat(obat);  System.out.println("Masukkan kode obat (ketik 0 untuk selesai)");  do {  System.out.print(">");  pesanan[x] = scan.next();  if (pesanan[x].equals("0")) {  pesanan[x] = "";  break;  } else {  x++;  }  } while (loop == "y");  System.out.println("");  break;  case 3:  Obat.pesananObat(pesanan, obat);  System.out.println("");  break;  case 4:  Obat.pesananObat(pesanan, obat, user);  System.out.println("");  l="n";  break;  }  } while (l == "y");  }  public static void cetakMenu() {  System.out.println( "MENU" );  System.out.println(">>==========>>\*<<==========<<");  System.out.println("1. Masukkan data user");  System.out.println("2. Masukkan data obat");  System.out.println("3. Lihat pesanan");  System.out.println("4. Cetak struk");  System.out.println(">>==========>>\*<<==========<<");  System.out.println("0. Keluar");  System.out.print("=> : ");  }  } |

1. **PEMBAHASAN**

**Masalah 1**

|  |  |
| --- | --- |
| **Convertor.java** | |
| 1  2  3  4  5  7  8  10  11  13  14  16  17  20  21  22  23  31  32  33  34  35  37  38  40  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  79  80  81  82  83  85  86  87  88  91  92  94  95  97  98  99  100 | Yang kita lakukan pertama kali yaitu mengdeklarasi class dengan nama Convertor.  Inisialisasi variable terbilang sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable sebagai atribut bertipe data integer dengan access modifier private.  Pendeklarasian method setBilangan tanpa pengembalian data (void) dan berparameter variable bilangan bertipe data integer.  Pengisian variable bilangan dengan nilai dari parameter bilangan.  Pendeklarasian method setTerbilang tanpa pengembalian data (void) dan berparameter variable terbilang bertipe data String.  Pengisian variable terbilang dengan nilai dari parameter terbilang.  Pendeklarasian method getBilangan dengan pengembalian bertipe data integer.  Pengembalian nilai variable bilangan ke method getBilangan.  Pendeklarasian method getTerbilang dengan pengembalian bertipe data String.  Pengembalian nilai variable terbilang ke method getTerbilang.  Pendeklarasian method overloadingMeth dengan pengembalian data bertipe String dan berparameter bilangan dengan tipe data integer.  Dekalrasi array angka bertipe data String dengan nilai {“Nol”,”Satu”,”Dua”,”Tiga”,  ”Empat”,”Lima”,”Enam”,”Tujuh”,”Delapan”,”Sembilan”,”Sepuluh”,”Sebelas”}.  Inisialisasi variable output bertipe data String dengan nilai “”.  Pengecekan kondisi apakah nilai dari variable bilangan kurang dari dua belas;  Pengisian variable output dengan nilai dari array angka indeks ke bilangan.  Pengecekan kondisi apakah nilai dari bilangan samadengan 20 atau 30 atau 40 atau 50 atau 60 atau 70 atau 80 atau 90.  Pengisian variable output dengan nilai dari array angka indeks ke bilangan dibagi 10 dan ditambah “ Puluh”.  Pengecekan kondisi apakah nilai dari variable bilangan kurang dari dua puluh.  Pengisian variable output dengan nilai dari array angka indeks ke bilangan dikurangi sepuluh ditambah “ Belas”.  Pengecekan kondisi pakah nilai dari variable bilangan kurang dari seratus.  Pengisian variable output dengan nilai dari array angka indeks ke bilangan dibagi 10 diikuti “ Puluh “ dan nilai dari array angka indeks ke bilangan mod 10.  Pengecekan kondisi apakah nilai dari variable bilangan sama dengan seratus.  Pengisian variable output dengan nilai “Seratus”.  Pengembalian nilai ke method overloadingMeth dengan nilai dari variable output.  Pendeklarasian method overloadingMeth dengan pengembalian nilai bertipe data integer dan berparameter bilangan bertipe data String  Inisialisasi variable output bertipe data integer dan bernilai 0.  Inisialisasi variable temp1 dan temp2 bertipe data String dan bernilai “”.  Inisialisasi array parts bertipe data string bernilai hasil pemecahan nilai variable bilangan.  Inisialisasi perulangan for dengan variable x bertipe data integer dan bernilai 0. Perulangan akan terus berjalan hingga kurang dari panjang array parts dan nilai x akan bertambah satu setiap perulangan.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “nol”.  Pengisian nilai variable temp1 dengan “0”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “satu”.  Pengisian nilai variable temp1 dengan “1”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “belas”.  Pengisian nilai variable temp1 dengan “1”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “dua”.  Pengisian nilai variable temp1 dengan “2”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “tiga”.  Pengisian nilai variable temp1 dengan “3”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “empat”.  Pengisian nilai variable temp1 dengan “4”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “lima”.  Pengisian nilai variable temp1 dengan “5”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “enam”.  Pengisian nilai variable temp1 dengan “6”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “tujuh”.  Pengisian nilai variable temp1 dengan “7”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “delapan”.  Pengisian nilai variable temp1 dengan “8”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “sembilan”.  Pengisian nilai variable temp1 dengan “9”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “sepuluh”.  Pengisian nilai variable temp1 dengan “10”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “sebelas”.  Pengisian nilai variable temp1 dengan “11”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “seratus”.  Pengisian nilai variable temp1 dengan “100”.  Fungsi yang dijalankan apabila fungsi if diatas tidak terpenuhi.  Pengisian variable temp1 dengan “”.  Pengecekan apakah nilai dari array parts indeks ke x bernilai “puluh”.  Pengecekan apakah panjang dari array parts adalah 2.  Pengisian nilai variable temp2 dengan temp2 diikuti “0”.  Fungsi yang dijalankan apabila fungsi if diatas tidak terpenuhi.  Pengisian nilai variable temp2 dengan temp2 diikuti dengan temp1.  Pengecekan kondisi apakah nilai dari array parts indeks ke x bernilai “belas”.  Pengisian nilai variable temp2 dengan temp1 diikuti dengan temp2.  Fungsi yang dijalankan apabila nilai dari if diatas tidak terpenuhi.  Pengisian nilai variable temp2 dengan temp2 diikuti dengan temp1.  Konversi nilai temp2 bertipe String ke integer dan ditampung pada variable output.  Pengembalian nilai ke method overloadingMeth dengan nilai dari variable output.  Pendeklarasian method cetak tanpa pengembalian data (void) berparameter cetak bertipe data String.  Proses mencetak nilai dari variable cetak.  Pendeklarasian method cetak tanpa pengembalian data (void) berparameter cetak bertipe data int.  Proses mencetak nilai dari variable cetak. |

|  |  |
| --- | --- |
| **Main\_in.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21  22  23  24  25  27  28  29  30  31  32  34  36  37  38  39  40  41  42  43  44  45  46  47 | Memanggil fungsi Scanner dalam java.  Deklarasi class dengan nama Main\_in.  Main method java.  Inisialisasi fungsi Scanner baru bernama scan.  Inisialisasi fungsi Scanner baru bernama scan2.  Deklarasi objek conv dari class Convertor.  Inisialisasi variable loop bertipe data String bernilai “y”.  Deklarasi perulangan do-while.  Pemanggilan method cetakMenu dalam class tersebut.  Proses pengecekan nilai dari inputan user menggunakan fungsi switch.  Fungsi yang dijalankan apabila nilai dari inputan bernilai 0.  Proses mencetak “Program selesai”.  Proses pengisian nilai variable loop dengan “n”.  Fungsi untuk menghentikan case 0.  Fungsi yang dijalankan apabila nilai dari inputan bernilai 1.  Proses mencetak “Masukkan angka : “.  Proses pemanggilan method setbilangan dari objek conv dengan parameter inputan user bertipe ata integer.  Proses pemanggilan method cetak dari objek conv dengan parameter method overloadingMeth yang berparameter nilai method getBilangan.  Proses mencetak “ ” (enter).  Fungsi untuk menghentikan case 1.  Fungsi yang dijalankan apabila nilai dari inputan bernilai 2.  Proses mencetak “Masukkan terbilang : “.  Proses pemanggilan method setTerbilang dari objek conv denan parameter input user bertipe data String.  Proses pemanggilan method cetak dari objek conv dengan parameter method overloadingMeth yang berparameter nilai method getTerbilang.  Proses mencetak “ ” (enter).  Fungsi untuk menghentikan case 2.  Fungsi yang dijalankan apabila inputan user tidak sesuai dengan kondisi diatas.  Proses mencetak “Menut tidak ada”.  Proses mencetak “ ” (enter).  Fungsi untuk menghentikan proses default.  Ekspresi dari perulangan do-while yang berulang apabila nilai variable loop bernilai  “y”.  Pendeklarasian method cetak menu tanpa pengembalian data (void).  Proses mencetak “PILIH MASUKAN”.  Proses mencetak “1. Angka -> Terbilang”.  Proses mencetak “2. Terbilang -> Angka”.  Proses mencetak “0. Keluar”.  Proses mencetak “pilih : “. |

**Masalah 2**

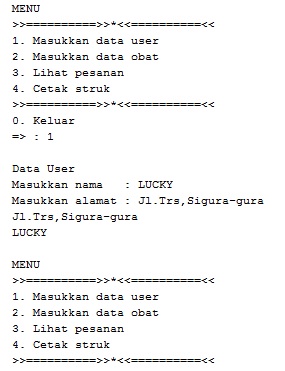
|  |  |
| --- | --- |
| **Obat.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  46  48  49  50  51  52  53  54  56  57  58  61  62  64  66  67  68  69  70  71  72  73  74  75  76  77  78  79  81  82  85  86 | Deklarasi class dengan nama Obat.  Inisialisasi variable KodeObat sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable NamaObat sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable HargaObat sebagai atribut bertipe data integer dengan access modifier private.  Pendeklarasian method setKodeObat tanpa pengembalian data (void) dengan parameter KodeObat bertipe data String.  Pengisian variable KodeObat dengan nilai variable KodeObat.  Pendeklarasian method setNamaObat tanpa pengembalian data (void) dengan parameter NamaObat bertipe data String.  Pengisian variable NamaObat dengan nilai variable NamaObat.  Pendeklarasian method setHargaObat tanpa pengembalian data (void) dengan parameter NamaObat bertipe data integer.  Pengisian variable HargaObat dengan nilai variable HargaObat.  Pendeklarasian method getKodeObat dengan pembembalian data bertipe String.  Proses pengembalian nilai ke method getKodeObat dari variable KodeObat.  Pendeklarasian method getNamaObat dengan pembembalian data bertipe String.  Proses pengembalian nilai ke method getNamaObat dari variable NamaObat.  Pendeklarasian method getHargaObat dengan pembembalian data bertipe integer.  Proses pengembalian nilai ke method getHargaObat dari variable HargaObat.  Pendeklarasian method daftarObat dengan pengembalian nilai bertipe data Array of objek Obat dengan parameter array obat.  Pendeklarasian array kodeObat bertipe data String dengan nilai {“001”,”002”,”003”,”004”,”005”}.  Pendeklarasian array namaObat bertipe data String dengan nilai {“Parasetamol”,”Antalgin”,”Tetrasiclin”,”Neosep Forte”,”Paramex Plus”}.  Pendeklarasian array hargaObat bertipe data integer dengan nilai {“10000”,”9000”,”8000”,”7000”,”6000”}.  Inisialisasi perulangan for dengan variable x bertipe data integer dengan niali 0. Akan terus berulang hingga nilai x kurang dari panjang array obat dan variable x akan bertambah 1 setiap perulangan.  Inisialisasi array obat indeks ke x.  Proses memanggil method setKodeObat dengan parameter nilai array kodeObat indeks ke x.  Proses memanggil method setNamaObat dengan parameter nilai array namaObat indeks ke x.  Proses memanggil method setHargaObat dengan parameter nilai array hargaObat indeks ke x.  Proses pengembalian nilai ke method daftarObat dengan array obat.  Pendeklarasian method cetakDaftarObat tanpa pengembalian data (void) dengan parameter array obat.  Proses mencetak “Daftar Obat”.  Proses mencetak “Kode Nama Harga”.  Inisialisasi perulangan for dengan variable x bertipe data integer dengan niali 0. Akan terus berulang hingga nilai x kurang dari panjang array obat dan variable x akan bertambah 1 setiap perulangan.  Proses mencetak nilai dari method getKodeObat objek array obt indeks ke x.  Proses mencetak nilai dari method getNamaObat objek array obt indeks ke x.  Proses mencetak nilai dari method getHargaObat objek array obt indeks ke x.  Pendeklarasian method pesananObat tanpa pengembalian nilai (void) dengan parameter array pesanan dan array obat.  Proses mencetak “Pesanan Obat”.  Inisialiasi variable total bertipe data integer dengan nilai 0.  Inisialisasi perulangan for dengan variable x bertipe data integer dengan niali 0. Akan terus berulang hingga nilai x kurang dari panjang array pesanan dan variable x akan bertambah 1 setiap perulangan.  Inisialisasi perulangan for dengan variable y bertipe data integer dengan niali 0. Akan terus berulang hingga nilai y kurang dari panjang array obat dan variable y akan bertambah 1 setiap perulangan.  Pengecekan apakah nilai dari method getKodeObat sama dengan array pesanan indeks ke x.  Proses mencetak nilai method getNamaObat.  Proses mencetak niali method getHargaObat.  Proses penjumlahan nilai dari variable total ditambah nilai dari method getHargaObat.  Proses mencetak “ “ (enter).  Proses mencetak “---------------”.  Proses mencetak “Total = Rp.” Diikuti nilai dari varialbe total.  Pendeklarasian method pesananObat tanpa pengembalian nilai (void) dengan parameter array pesanan, obat, dan objek user.  Inisialisasi variable total bertipe data integer dengan nilai 0.  Proses mencetak “APOTEK SEHAT”.  Proses mencetak “----------------------”.  Proses mencetak “a.n Bpk/Ibu : “ diikuti nilai dari method getNama.  Proses mencetak “alamat : “ diikuti nilai dari method getAlamat.  Proses mencetak “----------------------”.  Proses mencetak Pembelian.  Proses mencetak “Kode Nama Harga”.  Inisialisasi perulangan for dengan variable x bertipe data integer dengan niali 0. Akan terus berulang hingga nilai x kurang dari panjang array pesanan dan variable x akan bertambah 1 setiap perulangan.  Inisialisasi perulangan for dengan variable y bertipe data integer dengan niali 0. Akan terus berulang hingga nilai y kurang dari panjang array obat dan variable y akan bertambah 1 setiap perulangan.  Pengecekan apakah nilai dari method getKodeObat sama dengan array pesanan indeks ke x.  Proses mencetak nilai method getKodeObat.  Proses mencetak nilai method getNamaObat.  Proses mencetak niali method getHargaObat.  Proses penjumlahan nilai dari variable total ditambah nilai dari method getHargaObat.  Proses mencetak “------------------------------------------“.  Proses mencetak “Total Rp.” Diikuti nilai dari variable total. |

|  |  |
| --- | --- |
| **User.java** | |
| 1  2  3  4  5  6  7  8  9  11  12  13  15  16  17  19  20  21  23  24  26  27 | Memanggil fungsi Scanner dalam java.  Deklarasi class dengan nama User.  Inisialisasi fungsi Scanner baru bernama scanNama.  Inisialisasi fungsi Scanner baru bernama scanAlaamat.  Inisialisasi variable Nama sebagai atribut bertipe data String dengan access modifier private.  Inisialisasi variable Alamat sebagai atribut bertipe data String dengan access modifier private.  Pendeklarasian konstruktor tanpa parameter.  Pengisian variable nama dengan “Belum diisi”.  Pengisian variable alamat dengan “Belum diisi”.  Pendeklarasian konstruktor dengan parameter nama, alamat bertipe data String.  Pengisian variable nama dengan parameter nama.  Pengisian variable alamat dengan parameter nama.  Pendeklarasian method setNama tanpa pengembalian nilai (void).  Proses mencetak “Masukkan nama : “.  Pengisian variable Nama dengan inputan dari user bertipe data string.  Pendeklarasian method setAlamat tanpa pengembalian nilai (void).  Proses mencetak “Masukkan alamat : “.  Pengisian variable Alamat dengan inputan dari user bertipe data string.  Pendeklarasian method getNama dengan pengembalian bertipe data String.  Proses pengembalian data ke method getNama dari variable Nama.  Pendeklarasian method getAlamat dengan pengembalian bertipe data String.  Proses pengembalian data ke method getAlamat dari variable Alamat. |

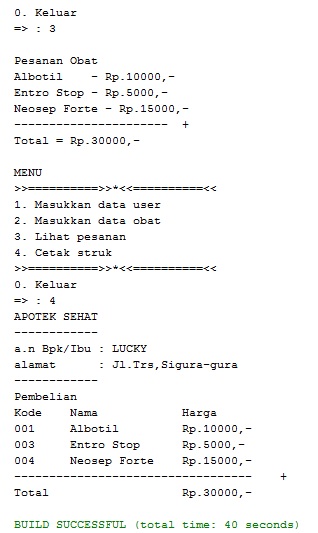
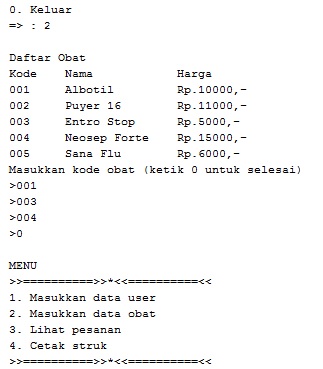
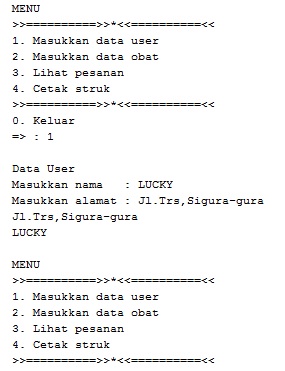
|  |  |
| --- | --- |
| **Main.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  30  31  32  33  34  35  36  37  39  40  41  42  43  44  45  46  47  48  49  50  52  54  55  56  57  58  59  60  61  62 | Memanggil fungsi Scanner dalam java.  Deklarasi class dengan nama Main.  Main method java.  Inisialisasi fungsi Scanner baru bernama scan.  Inisialisasi array pesanan bertipe data String dengan panjang 100.  Inisialisasi array of objek obat dengan panjang 5.  Inisialisasi objek user dari class User.  Inisialisasi variable l,lopp bertipe data String dengan nilai “y”.  Inisialisasi variable x bertipe data integer bernilai 0.  Deklarasi perulangan do-while.  Proses memanggil method cetakMenu.  Proses pengecekan nilai inputan user beritpe data integer dengan fungsi switch.  Fungsi yang dijalankan apabila inputan user bernilai 0.  Proses mencetak “Program selesai”.  Proses pengisian variable l dengan nilai “n”.  Fungsi untuk menghentikan proses case 0.  Fungsi yang dijalankan apabila inputan user bernilai 1.  Proses mencetak “Data User”.  Proses memanggil method setNama.  Proses memanggil method setLamat.  Proses mencetak nilai dari method getAlamat.  Proses mencetak nilai dari method getNama.  Proses mencetak “ ” (enter).  Fungsi untuk menghentikan proses case 1.  Fungsi yang dijalankan apabila inputan user bernilai 2.  Proses pengisian array obat dengan nilai dari method daftarObat.  Proses memanggil method cetakDaftarObat dengan parameter array obat.  Proses mencetak “Masukkan kode obat (ketik 0 untuk selesai)”.  Deklarasi perulangan do-while.  Proses mencetak “>”.  Proses pengisian array pesanan indeks ke x dengan inputan user bertipe data String.  Pengecekan pakah nilai dari array pesanan indeks ke x bernilai “0”.  Proses pengisian array pesanan indeks ke x dengan nilai “ ”.  Fungsi untuk keluar dari perulangan do-while.  Fungsi yang dijalankan apabila nilai dari if diatas tidak terpenuhi.  Proses menambah variable x dengan nilai satu (inkeremen).  Ekspresi perulangan do-while yang berulang jika nilai varialbe loop bernilai “y”.  Proses mencetak “ ” (enter).  Fungsi untuk menghentikan proses case 2.  Fungsi yang dijalankan apabila nilai dari inputan user bernilai 3.  Proses memanggil method pesnaanObat dengan parameter array pesanan dan obat.  Proses mencetak “ ” (enter).  Fungsi untuk menghentikan proses case 3.  Fungsi yang dijalankan apabila nilai dari inputan user bernilai 4.  Proses memanggil method pesananObat dengan parameter array pesanan,obat dan objek user.  Proses mencetak “ ” (enter).  Pengisian nilai variable l dengan “n”.  Fungsi untuk menghentikan proses case 4.  Ekspresi perulangan do-while yang berulang jika nilai varialbe l bernilai “y”.  Pendeklarasian method cetakMenu tanpa pengembalian nilai (void).  Proses mencetak “MENU”.  Proses mencetak “1. Masukkan data user”.  Proses mencetak “2. Masukkan data obat”.  Proses mencetak “3. Lihat pesanan”.  Proses mencetak “4. Cetak struk”.  Proses mencetak “0. Keluar”.  Proses mencetak “>”. |

1. **SCREENSHOT PROGRAM**

**Masalah 1**



**Masalah 2**



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

|  |  |
| --- | --- |
| Jawaban : | Tidak terjadi kesalah pada saat menjalankan class Overloading.java tersebut. |

1. Jika pada baris 7, pada parameter double value dan double value2 di hapus dan di ganti menjadi int a dan int b apa yang terjadi? Jelaskan!

|  |  |
| --- | --- |
| Jawaban : | Akan terjadi error karena method tersebut mempunyai nama yang sama dengan method sebelumnya (ini dinamakan overloading method) namun sebelum diganti parameter bertipe data berbeda. Jika parameter bertipe data sama dan berjumlah parameter sama maka java tidak akan bisa memilih satu dari overloading method tersebut untuk digunakan atau dengan kata lain dua method tersebut menjadi sama. |

1. Rubah method pada baris ketujuh menjadi method bertipe void, dan lakukan juga perubahan main method.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Jawaban : | Jika pada method void return harus dihilangkan maka source kode akan menjadi seperti ini :   |  |  | | --- | --- | | **Overloading.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 | import java.util.Scanner;  public class OverloadingMethod {  public static void HitungLuas(int a,  int b) {  int nilai = a \* b;  System.out.println("maka hasil luas :  " + nilai);  }  public static void HitungLuas(double  value, double value2) {  double nilai = value \* value2;  System.out.println("maka hasil luas :  " + nilai);  }  public static void main(String[] args)  {  Scanner in = new Scanner(System.in);  ystem.out.print("masukkan nilai  integer 1 : ");  int integer1 = in.nextInt();  System.out.print("masukkan nilai  integer 2 : ");  int integer2 = in.nextInt();  HitungLuas(integer1, integer2);  System.out.print("masukkan nilai  double 1 : ");  double double1 = in.nextDouble();  System.out.print("masukkan nilai  double 2 : ");  double double2 = in.nextDouble();  HitungLuas(double1, double2);  }  } | |

1. **Overloading Constructor**
2. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

|  |  |
| --- | --- |
| Jawaban : | Tidak ada kesalah yang membuat error pada kedua class tersebut. |

1. Pada class lingkaran tambahkan constructor dengan parameter int tinggi, apa yang terjadi dan jelaskan!

|  |  |
| --- | --- |
| Jawaban : | Akan terjadi error karena constructor baru dengan parameter int tinggi akan dianggap sama dengan constructor dengan parameter int alas karena tipe data dan jumlah parameternya sama; |

1. Pada class lingkaran tambahkan constrctor dengan tipe data String alas dan String tinggi, kemudian tambahkan method untuk melakukan parsing atau perubahan tipe data dari String menjadi integer. Setelah itu pada method main lakukan instansiasi objek dengan nama objek Lstring dengan memanggil constructor bertipe data String. Jelaskan!

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Jawaban : | Pada class Lingkaran.java ditambahkan method seperti ini :   |  |  | | --- | --- | | **Lingkaran.java** | | | 1  2  3  4  5 | public Lingkaran(String alas, String tinggi){  this.alas=Integer.parseInt(alas);  this.tinggi=Integer.parseInt(tinggi);  } |   Sedangkan pada class LingkaranMain.java ditambahkan :   |  |  | | --- | --- | | **LingkaranMain.java** | | | 1  2  3 | Lingkaran Lstring = new Lingkaran("2","10");  Lstring.displayMessage(); |   Cara tersebut berhasil karena constructor yang baru dengan masing-masing parameter bertipe data String belum didefinisikan sebelumnya sehingga tidak dianggap sama. Pada pendefinisian constructor di main class juga sama namun dengan parameter bertipe data String. |

1. **KESIMPULAN**

Pada dasar nya Overloading adalah pemakaian beberapa methods ataupun properties dengan nama yang sama, tetapi memiliki daftar parameter/argument yang berbeda..

Begitu pula pada overloading method dan constructor, yakni pemakaian method dan constructor dengan nama yang sama dengan parameter yang berbeda sehingga menghasilkan dua fungsi yang berbeda sesuai dengan parameter yang dimasukkan.

Overloading method dan constructor boleh memiliki tipe data yang sama namun harus berbeda jumlah parameternya. Begitu pula sebaliknya, overloading method dan constructor boleh memiliki jumlah parameter yang sama namun dengan susunan tipe data yang berbeda.

Dalam pembuatan overloading method dengan contoh berikut ini. Contoh :

|  |  |
| --- | --- |
|  | **LuasPersegi.java** |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | public class LuasPersegi {  private int sisi;  public int hitungLuas(int sisi){  int luas=0;  luas=sisi\*sisi;  return luas;  }  public int hitungLuas(String sisi){  int intSisi=Integer.parseInt(sisi);  luas=intSisi\*intSisi;  return luas;  }  } |

Contoh diatas merupakan salah satu contoh dari overloading method. Method hitung Luas yang pertama bertipe data int dan yang kedua bertipe data String. Apabila kedua method tersebut dipanggil dalam main method tidak akan terjadi errror karena kedua method tersebut berbeda dari tipe data kedua method tersebut. Pada method yang kedua yang berparameter String untuk menghitung luas maka parameter tersebut akan dirubah terlebih dahulu menjadi integer sebelum dilakukan perhitungan.

Begitu pula dengan overloading constructor, telah dibahas diatas bahwa parameter berpengaruh dalam pendeklarasian overloading constructor. Berikut contoh dari pendeklarasian overloading constructor :

|  |  |
| --- | --- |
|  | **LuasPersegi.java** |
| 1  2  3  4  5  6  7  8  9  10  11  12 | public class LuasPersegi {  private int sisi;  public LuasPersegi(){  this.sisi = 0;  }  public LuasPersegi(int sisi){  this.sisi = sisi;  }  public LuasPersegi(String sisi){  this.sisi = Integer.parseInt(sisi);  }  } |

Seperti contoh diatas, dalam menginisilaisasi overloading constructor hampir sama dengan overloading method. Namun, pada overloading constructor biasanya hanya digunakan untuk menetapkan nilai awal suatu variable pada saat inisialisasi objek pada main class dengan parameter yang tepat.