1. **DEFINISI MASALAH**

1. Buatlah program untuk membuat kalkulator penjumlahan, pengurangan, perkalian dan pembagian dengan menggunakan parameter dan argumen.

1. **SOURCE CODE**

|  |  |
| --- | --- |
| OperasiAritmatika.java | |
| 1 | public class OperasiAritmatika { |
| 2 | private double angka1, angka2; |
| 3 |  |
| 4 | public OperasiAritmatika(){ |
| 5 |  |
| 6 | } |
| 7 |  |
| 8 | public void InpAngka(double angka1, double angka2){ |
| 9 | this.angka1 = angka1; |
| 10 | this.angka2 = angka2; |
| 11 | } |
| 12 |  |
| 13 | public int Penjumlahan(){ |
| 14 | return (int)angka1 + (int)angka2; |
| 15 | } |
| 16 |  |
| 17 | public int Pengurangan(){ |
| 18 | return (int)angka1 - (int)angka2; |
| 19 | } |
| 20 |  |
| 21 | public double Perkalian(){ |
| 22 | return angka1 \* angka2; |
| 23 | } |
| 24 |  |
| 25 | public double Pembagian(){ |
| 26 | return angka1 / angka2; |
| 27 | } |
| 28 | } |

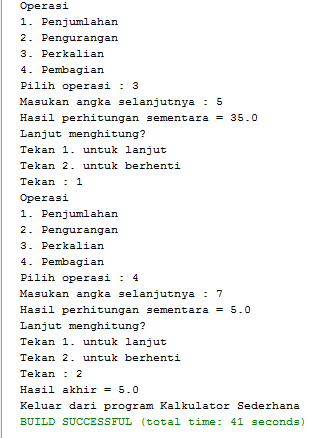
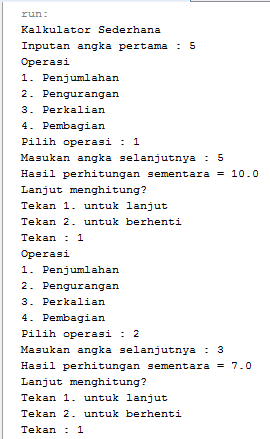
|  |  |
| --- | --- |
| MainKalkulator.java | |
| 1 | import java.util.Scanner; |
| 2 | public class MainKalkulator { |
| 3 | public static void main(String[] args) { |
| 4 | Scanner input = new Scanner(System.in); |
| 5 | OperasiAritmatika arit = new OperasiAritmatika(); |
| 6 | double angka1, angka2; |
| 7 | int pilih, pil; |
| 8 | System.out.println("Kalkulator Sederhana"); |
| 9 | System.out.print("Inputan angka pertama : "); |
| 10 | angka1 = input.nextDouble(); |
| 11 | do { |
| 12 | System.out.println("Operasi"); |
| 13 | System.out.println("1. Penjumlahan"); |
| 14 | System.out.println("2. Pengurangan"); |
| 15 | System.out.println("3. Perkalian"); |
| 16 | System.out.println("4. Pembagian"); |
| 17 | System.out.print("Pilih operasi : "); |
| 18 | pilih = input.nextInt(); |
| 19 | System.out.print("Masukan angka selanjutnya : "); |
| 20 | angka2 = input.nextDouble(); |
| 21 | switch (pilih) { |
| 22 | case 1: |
| 23 | arit.InpAngka(angka1, angka2); |
| 24 | angka1 = arit.Penjumlahan(); |
| 25 | break; |
| 26 | case 2: |
| 27 | arit.InpAngka(angka1, angka2); |
| 28 | angka1 = arit.Pengurangan(); |
| 29 | break; |
| 30 | case 3: |
| 31 | arit.InpAngka(angka1, angka2); |
| 32 | angka1 = arit.Perkalian(); |
| 33 | break; |
| 34 | case 4: |
| 35 | arit.InpAngka(angka1, angka2); |
| 36 | angka1 = arit.Pembagian(); |
| 37 | break; |
| 38 | } |
| 39 | System.out.println("Hasil perhitungan sementara = " + angka1); |
| 40 | System.out.println("Lanjut menghitung? \nTekan 1. untuk lanjut \nTekan 2. untuk berhenti "); |
| 41 | System.out.print("Tekan : "); |
| 42 | pil = input.nextInt(); |
| 43 | } while (pil != 2); |
| 44 | System.out.println("Hasil akhir = " + angka1); |
| 45 | System.out.println("Keluar dari program Kalkulator Sederhana"); |
| 46 | } |
| 47 | } |

1. **PEMBAHASAN**

|  |  |
| --- | --- |
| OperasiAritmatika.java | |
| 1  2  4  8  9  10  13  14  17  18  21  22  25  26  28 | Deklarasi class public bernama OperasiAritmatika.  Deklarasi variable private bertipe double angka1, angka2.  Deklarasi method contructor OperasiAritmatika tanpa parameter.  Deklarasi method void InpAngka dengan parameter double angka1, double angka2.  Inisialisasi this.angka1 = angka1.  Inisialisasi this.angka2 = angka2.  Deklarasi method int Penjumlahan.  Mengembalikan nilai (int)angka1 + (int)angka2.  Deklarasi method int Pengurangan.  Mengembalikan nilai (int)angka1 – (int)angka2.  Deklarasi method double Perkalian.  Mengembalikan nilai angka1 \* angka2.  Deklarasi method double Pembagian.  Mengembalikan nilai angka1 / angka2.  Batas akhir dari class OperasiAritmatika. |

|  |  |
| --- | --- |
| MainCalculator.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  39  40  41  42  43  44  45  47 | Perintah untuk import dari package scanner.  Deklarasi class public bernama MainKalkulator.  Deklarasi main method.  Deklarasi scanner baru dengan data. Dan data tersebut dapat diubah-ubah (variable input).  Deklarasi variable arit sebagai objek dari class OperasiAritmatika.  Deklarasi variable bertipe double angka1, angka2.  Deklarasi variable bertipe int pilih, pil.  Mengoutputkan kalimat “Kalkulator Sederhana”.  Mengoutputkan kalimat “Inputan angka pertama : “.  Perintah menginputkan nilai angka1.  Melakukan perulangan do-while.  Mengoutputkan kalimat “Operasi”.  Mengoutputkan kalimat “1. Penjumlahan”.  Mengoutputkan kalimat “2. Pengurangan”.  Mengoutputkan kalimat “3. Perkalian”.  Mengoutputkan kalimat “4. Pembagian”.  Mengoutputkan kalimat “Pilih Operasi : ”.  Perintah menginputkan nilai pilih.  Mengoutputkan “Masukkan angka selanjutnya : ”.  Perintah menginputkan nilai angka2.  Proses seleksi switch dengan mengetahui kondisi nilai “pilih”.  Jika dalam kondisi “pilih”=1 akan melakukan:  Melakukan proses method InpAngka dengan parameter angka1, angka2.  Perintah angka1 = arit.Pejumlahan().  Perintah untuk berhenti.  Jika dalam kondisi “pilih”=2 akan melakukan:  Melakukan proses method InpAngka dengan parameter angka1, angka2.  Perintah angka1 = arit.Pengurangan().  Perintah untuk berhenti.  Jika dalam kondisi “pilih”=3 akan melakukan:  Melakukan proses method InpAngka dengan parameter angka1, angka2.  Perintah angka1 = arit.Perkalian().  Perintah untuk berhenti.  Jika dalam kondisi “pilih”=4 akan melakukan:  Melakukan proses method InpAngka dengan parameter angka1, angka2.  Perintah angka1 = arit.Pembagian().  Perintah untuk berhenti.  Mengoutputkan kalimat “Hasil perhitungan sementara = angka1”.  Mengoutputkan kalimat “Lanjut Menghitung? \nTekan 1. untuk lanjut \n2. Untuk berhenti”.  Mengoutputkan kalimat “Tekan : ”.  Perintah menginputkan nilai pil.  Mengulang jika nilai pil != 2.  Mengoutputkan kalimat “Hasil akhir = angka1”.  Mengoutputkan kalimat “Keluar dari program Kalkulator Sederhana”.  Batas akhir dari class MainKalkulator. |

1. **SCREENSHOT PROGRAM**



1. **KESIMPULAN**
2. Yang dimaksud dengan OOP (Object Oriented Program) adalah membuat program dengan objek-objek yang dapat digunakan. Yang dimaksud dengan objek adalah perwujudan dari method dari data-data yang akan digunakan. Perbedaan utama dari OOP dan structure Programming adalah proses dari jalannya program tersebut. Jika structure Programing berjalan berurutan tanpa melompat-lompat, OOP dapat melakukan lompatan-lompatan pada prosesnya sehingga tidak selalu berurutan seperti structure Programing.
3. Perbedaan method void dengan non-void terdapat pada nilai yang akan dikembalikan. Method void tidak mengembalikan nilai apapun. Namun, hanya berisi proses-proses yang dapat berguna untuk pemanggil method atau dapat mencetak tulisan sedangkan method non-void yang terdiri dari banyak macam yang dapat mengembalikan nilai untuk mengisi variabel. Dalam penggunaanya method void banyak digunakan untuk mengisi nilai variabel dan method non void untuk mendapatkan hasilnya.   
   Contoh dari method void :

public void setInput (int input){

this.input = input;  
}  
  
Contoh Method non-void :  
  
public int getInput(){  
 return input;  
}

1. Perbedaan utama dari contoh di atas adalah adanya return value, return value adalah nilai yang dikembalikan sebuah method untuk pemanggilnya. Dari contoh diatas return value yang digunakan adalah tipe data integer. Seluruh tipe data yang tidak abstrak dapat menjadi return value.  
     
   contoh return value array :  
     
   public [] int getArrayInput(){  
    return arrayInput;  
   }