

LAPORAN 2

Pemrograman Berorientasi Objek

**“MEMBUAT KELAS,
KONSTRUKTOR, AKSESOR
DAN MUTATOR”**

Dosen Pengampu : Paulina Heruningsih Prima Rosa



DIBUAT OLEH :

Nama : Yohanis Calvin D.P.U.Pati

NIM : 245314033

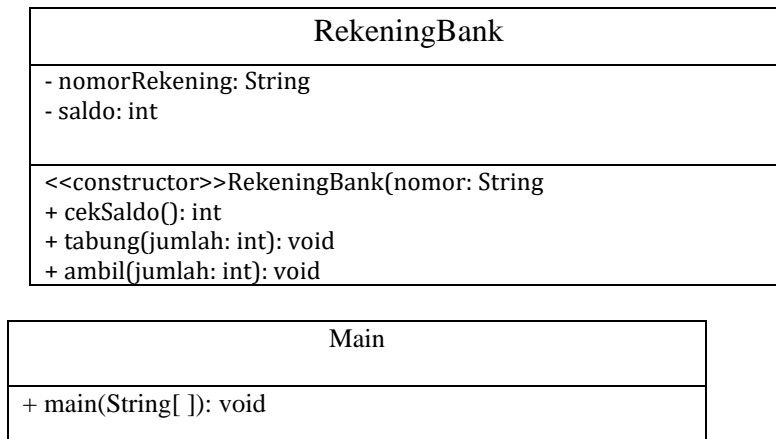
KELAS : BP

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2025**

latihan kasus 1

A. DIAGRAM CLASS

Penggambaran diagram kelas



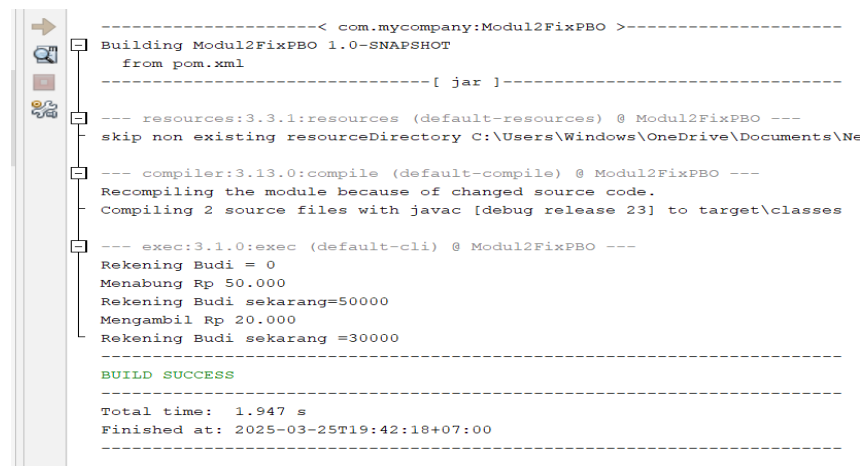
B. LISTING PROGRAM

```
8  *
9  * @author Calvin Pati
10 */
11 public class RekeningBank { //awal kelas rekeningBank
12     private String nomorRekening; //atribut nomorRekening untuk simpan nomor rekening
13     private int saldo; //atribut saldo untuk simpan saldo
14
15     public RekeningBank(String nomor) { //metode untuk rekeningBank
16         nomorRekening = nomor; //untuk seting nomorRekening
17         saldo = 0; //untuk seting saldo nilai default 0
18     }
19
20     public int cekSaldo() { //metode untuk cek saldo
21         return saldo;
22     }
23
24     public void tabung(int jumlah) { //metode untuk tabungan
25         saldo = saldo + jumlah;
26     }
27
28     public void ambil(int jumlah) { //metode untuk ambil saldo
29         saldo = saldo - jumlah;
30     }
31 }
32
```

```
8  *
9  * @author Calvin Pati
10 */
11 public class BankMain { //awal kelas BankMain
12
13     public static void main(String args[]) { //metode utama
14         RekeningBank rekeningBudi;
15         rekeningBudi = new RekeningBank("12345"); //membuat objek rekeningBudi
16         System.out.println("Rekening Budi = " + rekeningBudi.cekSaldo()); //untuk cetak informasi rekening
17         System.out.println("Menabung Rp 50.000"); //untuk cetak jumlah tabung
18         rekeningBudi.tabung(50000);
19         System.out.println("Rekening Budi sekarang = " + rekeningBudi.cekSaldo()); //untuk cetak dan cek saldo
20         System.out.println("Mengambil Rp 20.000");
21         rekeningBudi.ambil(20000);
22         System.out.println("Rekening Budi sekarang = " + rekeningBudi.cekSaldo()); //untuk menampilkan saldo Budi sekarang
23     }
24 } //akhir kelas BankMain
25
```

C. OUTPUT Unit Test nya

- Setelah program nya saya perbaiki dan saya test memakai package test unit maka tidak ada lagi yang error dari program RekeningBank.



```
-----< com.mycompany:Modul2FixPBO >-----
Building Modul2FixPBO 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Modul2FixPBO ---
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\Ne

--- compiler:3.13.0:compile (default-compile) @ Modul2FixPBO ---
Recompiling the module because of changed source code.
Compiling 2 source files with javac [debug release 23] to target\classes

--- exec:3.1.0:exec (default-cli) @ Modul2FixPBO ---
Rekening Budi = 0
Menabung Rp 50.000
Rekening Budi sekarang=50000
Mengambil Rp 20.000
Rekening Budi sekarang =30000

BUILD SUCCESS

Total time: 1.947 s
Finished at: 2025-03-25T19:42:18+07:00
-----
```

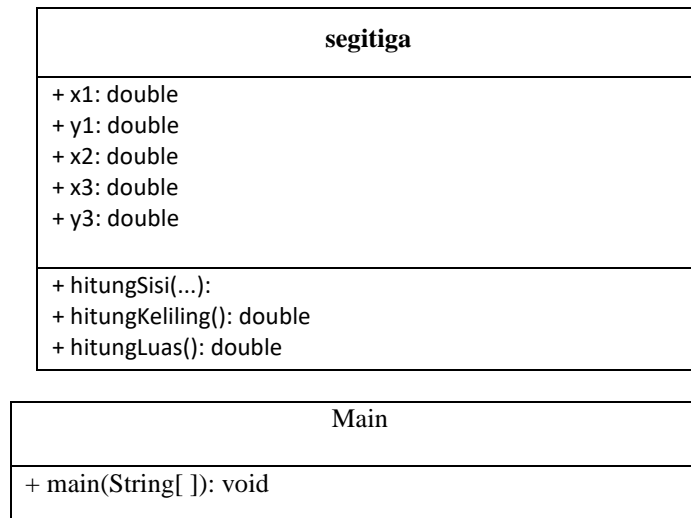
D. ANALISA

No	Syntax	penjelasan
1	private String nomorRekening;//atribut nomorRekening untuk simpan nomor rekening private int saldo;//atribut Saldo untuk simpan saldo	// ini adalah atribut yang bersifar private dan berfungsi untuk simppan informasi rekening bank dan saldo.
2	public RekeningBa nk(String nomor) { nomorRekening =nomor; saldo = 0; }	//konstruktor untuk memebuat objek rekeningBank baru, untuk mengatur nomor rekening dari parameter dan menginisialisasi saldo awal ke 0.
3	public int cekSaldo() { return saldo; }	// methode atau fungsi untuk mengecek saldo nasabah
4	public void tabung(int jumlah) { saldo = saldo + jumlah; }	// methode untuk menambah saldo rekening sebesar jumlah yang diinput

5	public void ambil(int jumlah) { saldo = saldo - jumlah; }}	// method untuk menarik uang dari rekening, dan mengurangi saldo sebesar jumlah yang diinput
	Main Syntaks	Penjelasan
1	public class BankMain {	//awal kelas bank main
2	public static void main(String args[]) {	// metode utama yang berfungsi untuk menjalankan program
3	RekeningBank rekeningBudi; rekeningBudi = new RekeningBank("12345");	// membuat objek RekeningBudi dari kelas RekeningBank dengan nomer rekening '12345' dan saldo awal 0
4	System.out.println("Rekening Budi = " + rekeningBudi.cekSaldo());	//berfungsi untuk mencetak informasi saldoo rekening Budi ke konsol dengan format "Rekening Budi = [jumlah saldo]"
5	System.out.println("Menabung Rp 50.000"); rekeningBudi.tabung(50000);	//berfungsi untuk melakukan penyetoran uang sebesar Rp.50.000 ke rekening Budi dan menampilkan informasi transaksi ke konsol.
6	System.out.println("Rekening Budi sekarang=" + rekeningBudi.cekSaldo()); System.out.println("Mengambil Rp 20.000"); rekeningBudi.ambil(20000);	//untuk menampilkan saldo, menampilkan transaksi penarikan yang dilakukan, melakukan penarikan dana sebesar Rp.20.000 dari rekening

latihan kasus 2

A. DIAGRAM CLASS



B. LISTING PROGRAM

```
11 public class Segitiga {
12     // Deklarasi variabel koordinat titik 1 (x1, y1)
13     public double x1, y1;
14     // Deklarasi variabel koordinat titik 2 (x2, y2)
15     public double x2, y2;
16     // Deklarasi variabel koordinat titik 3 (x3, y3)
17     public double x3, y3;
18
19     // Method untuk menghitung panjang sisi antara dua titik
20     public double hitungSisi(double xa, double ya, double xb, double yb) {
21         // Menghitung jarak Euclidean antara titik (xa,ya) dan (xb,yb)
22         return Math.sqrt(Math.pow(xa - xb, 2) + Math.pow(ya - yb, 2));
23     }
24
25     // Method untuk menghitung keliling segitiga
26     public double hitungKeliling() {
27         // Menjumlahkan panjang ketiga sisi segitiga
28         return hitungSisi(x1, y1, x2, y2) +
29             hitungSisi(x2, y2, x3, y3) +
30             hitungSisi(x3, y3, x1, y1);
31     }
32
33     // Method untuk menghitung luas segitiga
34     public double hitungLuas() {
35         // Menghitung semi-perimeter (setengah keliling)
36         double s = hitungKeliling() / 2;
37         // Menghitung luas dengan rumus Heron
38         return Math.sqrt(s * (s - hitungSisi(x1, y1, x2, y2)) *
39             (s - hitungSisi(x2, y2, x3, y3)) *
40             (s - hitungSisi(x3, y3, x1, y1)));
41     }
42 }
```

```
10
11 public class SegitigaMain {
12     // Method utama sebagai entry point program
13     public static void main(String[] args) {
14
15         // Membuat objek segitiga pertama (seg1)
16         Segitiga seg1 = new Segitiga();
17
18         // Mengisi koordinat titik pertama segitiga (x1,y1)
19         seg1.x1 = 1; seg1.y1 = 2;
20         // Mengisi koordinat titik kedua segitiga (x2,y2)
21         seg1.x2 = 7; seg1.y2 = 2;
22         // Mengisi koordinat titik ketiga segitiga (x3,y3)
23         seg1.x3 = 7; seg1.y3 = 9;
24
25         // Menghitung dan menampilkan luas segitiga pertama
26         System.out.println("Luas segitiga dengan titik (1,2) (7,2) dan (7,9) adalah : " + seg1.hitungLuas());
27
28         // Membuat objek segitiga kedua (seg2)
29         Segitiga seg2 = new Segitiga();
30
31         // Mengisi koordinat titik pertama segitiga (x1,y1)
32         seg2.x1 = 0; seg2.y1 = 0;
33         // Mengisi koordinat titik kedua segitiga (x2,y2)
34         seg2.x2 = 5; seg2.y2 = 4;
35         // Mengisi koordinat titik ketiga segitiga (x3,y3)
36         seg2.x3 = 8; seg2.y3 = 2;
37
38         // Menghitung dan menampilkan luas segitiga kedua
39         System.out.println("Luas segitiga dengan titik (0,0) (5,4) dan (8,2) adalah : " + seg2.hitungLuas());
40     }
41 }
42
```

C. OUTPUT

- Setelah program nya saya perbaiki dan saya test memakai package test unit maka tidak ada lagi yang error dari program karena sudah saya perbaiki.

```

-----< com.mycompany:Modul2FixPBO >-----
[ ] Building Modul2FixPBO 1.0-SNAPSHOT
    from pom.xml
    -----[ jar ]-----
[ ] --- resources:3.3.1:resources (default-resources) @ Modul2FixPBO ---
    skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\NetBeansProjects\Mc
[ ] --- compiler:3.13.0:compile (default-compile) @ Modul2FixPBO ---
    Recompiling the module because of changed source code.
    Compiling 4 source files with javac [debug release 23] to target\classes
[ ] --- exec:3.1.0:exec (default-cli) @ Modul2FixPBO ---
    Luas segitiga dengan titik (1,2) (7,2) dan (7,9) adalah :21.0
    Luas segitiga dengan titik (0,0) (5,4) dan (8,2) adalah :11.0
    -----
    BUILD SUCCESS
    -----
    Total time:  2.822 s
    Finished at: 2025-03-25T21:47:07+07:00
    -----
  
```

D. ANALISA

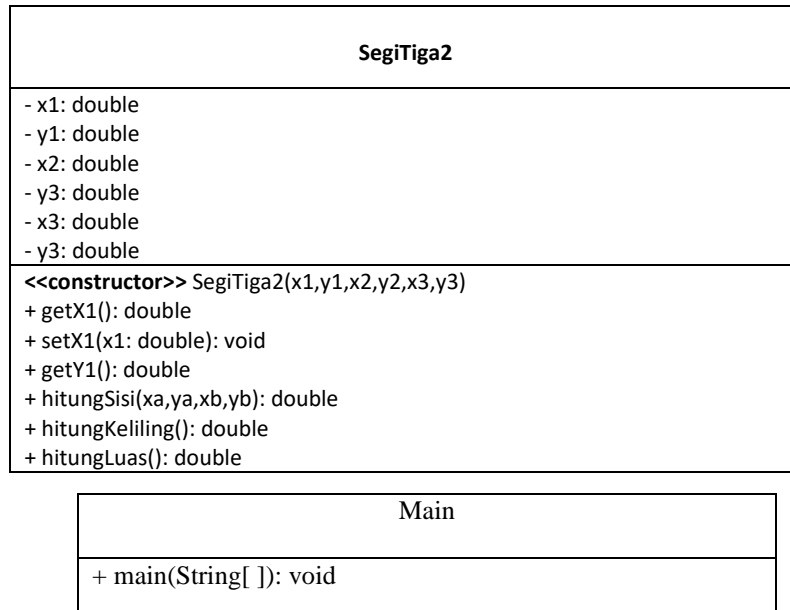
No	Syntax	penjelasan
1	public class Segitiga {	// kelas Segitiga yang digunakan untuk merepresentasikan segitiga berdasarkan koordinat tiga titiknya.
2	public double x1, y1;	//deklarasikan variabel x1 dan y1 sebagai koordinat titik pertama segitiga.
3	public double x2, y2;	//deklarasikan variabel x2 dan y2 sebagai koordinat titik kedua segitiga.
4	public double x3, y3;	// deklarasikan variabel x3 dan y3 sebagai koordinat titik ketiga segitiga.
5	public double hitungSisi(double xa, double ya, double xb, double yb) {	// Method untuk menghitung panjang sisi antara dua titik berdasarkan rumus Euclidean.
6	return Math.sqrt(Math.pow(xa - xb, 2) + Math.pow(ya - yb, 2));	// Menggunakan rumus $\sqrt{(xa - xb)^2 + (ya - yb)^2}$ untuk menghitung panjang sisi.

7	public double hitungKeliling() {	// Method untuk menghitung keliling segitiga dengan menjumlahkan panjang tiga sisinya.
8	return hitungSisi(x1, y1, x2, y2) + hitungSisi(x2, y2, x3, y3) + hitungSisi(x3, y3, x1, y1);	//method hitungSisi() untuk menjumlahkan panjang ketiga sisi segitiga.
9	public double hitungLuas() {	// Method untuk menghitung luas segitiga menggunakan rumus Heron.
10	double s = hitungKeliling() / 2;	// Menggunakan rumus Heron untuk menghitung luas segitiga: $\sqrt{s(s-a)(s-b)(s-c)}$.
	SegitigaMain Syntaks	Penjelasan
1	public class SegitigaMain {	// deklarasikan kelas SegitigaMain, yang berfungsi sebagai program utama untuk menjalankan perhitungan segitiga.
2	public static void main(String[] args) {	// Method utama main(), yang berfungsi sebagai titik masuk (entry point) program.
3	Segitiga seg1 = new Segitiga();	// Membuat objek seg1 dari kelas Segitiga.
4	seg1.x1 = 1; seg1.y1 = 2;	// Mengisi koordinat titik pertama (x1, y1) dari objek seg1.
5	seg1.x2 = 7; seg1.y2 = 2;	// Mengisi koordinat titik kedua (x2, y2) dari objek seg1.
6	seg1.x3 = 7; seg1.y3 = 9;	// Mengisi koordinat titik ketiga (x3, y3) dari objek seg1.
7	System.out.println("Luas segitiga dengan titik (1,2) (7,2) dan (7,9) adalah :"+ seg1.hitungLuas());	// hitung luas segitiga pertama menggunakan hitungLuas() dan mencetak hasilnya ke konsol.
8	Segitiga seg2 = new Segitiga();	// buat objek seg2 dari kelas Segitiga

9	<code>seg2.x1 = 0; seg2.y1 = 0;</code>	// Mengisi koordinat titik pertama (x1, y1) dari objek seg2.
10	<code>seg2.x2 = 5; seg2.y2 = 4;</code>	// Mengisi koordinat titik kedua (x2, y2) dari objek seg2.
11	<code>seg2.x3 = 8; seg2.y3 = 2;</code>	// Mengisi koordinat titik ketiga (x3, y3) dari objek seg2.
12	<code>System.out.println("Luas segitiga dengan titik (0,0)(5,4) dan (8,2) adalah ." + seg2.hitungLuas());</code>	// hitung luas segitiga kedua menggunakan hitungLuas() dan mencetak hasilnya ke konsol.

Praktikum kausus 3

A. DIAGRAM CLASS



B. LISTING PROGRAM

```
11 public class SegiTiga2 {
12     private double x1, y1; // Titik pertama (x1, y1)
13     private double x2, y2; // Titik kedua (x2, y2)
14     private double x3, y3; // Titik ketiga (x3, y3)
15
16     // Constructor dengan parameter untuk inialisasi langsung
17     public SegiTiga2(double x1, double y1, double x2, double y2, double x3, double y3) {
18         this.x1 = x1; // Mengisi nilai x1 dari parameter ke atribut x1
19         this.y1 = y1; // Mengisi nilai y1 dari parameter ke atribut y1
20         this.x2 = x2; // Mengisi nilai x2 dari parameter ke atribut x2
21         this.y2 = y2; // Mengisi nilai y2 dari parameter ke atribut y2
22         this.x3 = x3; // Mengisi nilai x3 dari parameter ke atribut x3
23         this.y3 = y3; // Mengisi nilai y3 dari parameter ke atribut y3
24     }
25
26     // Getter untuk mendapatkan nilai x1
27     public double getX1() {
28         return x1; // Mengembalikan nilai x1
29     }
30
31     // Setter untuk mengubah nilai x1
32     public void setX1(double x1) {
33         this.x1 = x1; // Mengupdate nilai atribut x1 dengan nilai parameter
34     }
35
36     // Getter untuk mendapatkan nilai y1
37     public double getY1() {
38         return y1; // Mengembalikan nilai y1
39     }
40
41     // Setter untuk mengubah nilai y1
42     public void setY1(double y1) {
43         this.y1 = y1; // Mengupdate nilai atribut y1 dengan nilai parameter
44     }
45
46     // [Getter dan Setter untuk x2,y2,x3,y3 dibuat dengan pola yang sama...]
47
48     // Method untuk menghitung panjang sisi antara dua titik
49     public double hitungSisi(double xa, double ya, double xb, double yb) {
50         // Menghitung jarak Euclidean antara dua titik
51         return Math.sqrt(Math.pow(xa - xb, 2) + Math.pow(ya - yb, 2));
52     }
53
54     // Method untuk menghitung keliling segitiga
55     public double hitungKeliling() {
56         // Menjumlahkan ketiga sisi segitiga
57         return hitungSisi(x1, y1, x2, y2) +
58                hitungSisi(x2, y2, x3, y3) +
59                hitungSisi(x3, y3, x1, y1);
60     }
61
62     // Method untuk menghitung luas segitiga
63     public double hitungLuas() {
64         double s = hitungKeliling() / 2; // Menghitung semi-perimeter
65         // Menghitung luas dengan rumus Heron
66         return Math.sqrt(s * (s - hitungSisi(x1, y1, x2, y2)) *
67                           (s - hitungSisi(x2, y2, x3, y3)) *
68                           (s - hitungSisi(x3, y3, x1, y1)));
69     }
70 }
```

```
8
9
10 * @author Calvin Pati
11 */
12 public class SegiTiga2Main {
13     // Method utama sebagai entry point program
14     public static void main(String[] args) {
15         // Membuat objek segitiga pertama dengan constructor
16         SegiTiga2 seg1 = new SegiTiga2(1, 2, 7, 2, 7, 9);
17         // Menampilkan luas segitiga pertama
18         System.out.println("Luas segitiga 1: " + seg1.hitungLuas());
19
20         // Membuat objek segitiga kedua dengan constructor
21         SegiTiga2 seg2 = new SegiTiga2(0, 0, 5, 4, 8, 2);
22         // Menampilkan luas segitiga kedua
23         System.out.println("Luas segitiga 2: " + seg2.hitungLuas());
24
25         // Mengubah titik x1 segitiga pertama menggunakan setter
26         seg1.setX1(3);
27         // Menampilkan luas segitiga pertama setelah diubah
28         System.out.println("Luas segitiga 1 setelah diubah: " + seg1.hitungLuas());
29     }
30 }
```

C. OUTPUT

- Setelah program nya saya perbaiki dan saya test maka tidak ada lagi yang error di program.

```
-----< com.mycompany:Modul2FixPBO >-----
Building Modul2FixPBO 1.0-SNAPSHOT
  from pom.xml
-----[ jar ]-----

--- resources:3.3.1:resources (default-resources) @ Modul2FixPBO ---
- skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\NetBeansProjects\M
--- compiler:3.13.0:compile (default-compile) @ Modul2FixPBO ---
- Nothing to compile - all classes are up to date.
--- exec:3.1.0:exec (default-cli) @ Modul2FixPBO ---
Luas segitiga 1: 21.0
Luas segitiga 2: 11.0
Luas segitiga 1 setelah diubah: 14.0
BUILD SUCCESS
Total time: 1.644 s
Finished at: 2025-03-25T22:36:36+07:00
```

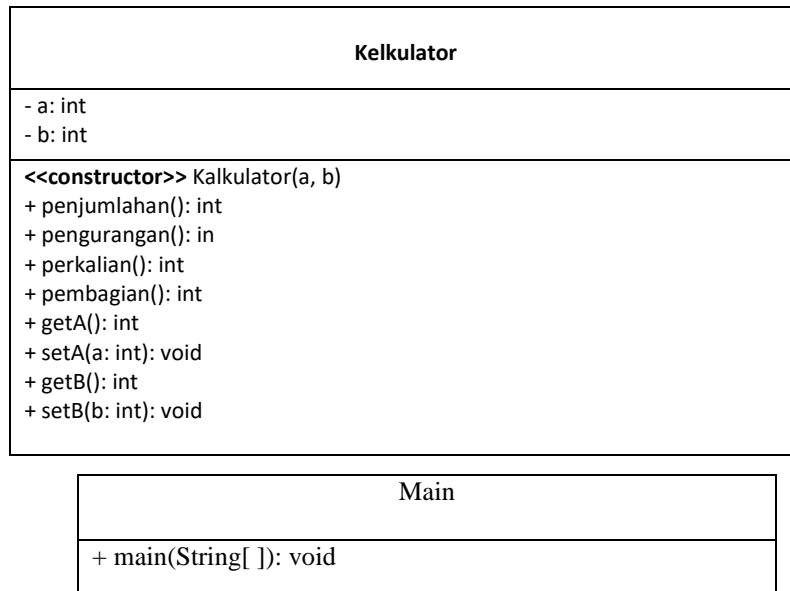
D. ANALISA

No	Syntax	penjelasan
1	public class SegiTiga2 {	// deklarasikan kelas SegiTiga2, yang merepresentasikan segitiga dengan tiga titik koordinat.
2	private double x1, y1;	// deklarasikan atribut x1 dan y1 sebagai titik pertama segitiga.
3	private double x2, y2;	// deklarasikan atribut x2 dan y2 sebagai titik kedua segitiga.
4	private double x3, y3;	// deklarasikan atribut x3 dan y3 sebagai titik ketiga segitiga.
5	public SegiTiga2(double x1, double y1, double x2, double y2, double x3, double y3) {	// Konstruktor untuk menginisialisasi nilai koordinat segitiga saat objek dibuat.
6	this.x1 = x1; this.y1 = y1;	// Mengisi nilai titik pertama dari parameter ke atribut kelas.
7	this.x2 = x2; this.y2 = y2;	// Mengisi nilai titik kedua dari parameter ke atribut kelas.

8	<code>this.x3 = x3; this.y3 = y3;</code>	// Mengisi nilai titik ketiga dari parameter ke atribut kelas.
9	<code>public double getX1() { return x1; }</code>	// Getter untuk mendapatkan nilai x1.
10	<code>public void setX1(double x1) { this.x1 = x1; }</code>	// Setter untuk mengubah nilai x1.
11	<code>public double getY1() { return y1; }</code>	// Getter untuk mendapatkan nilai y1.
12	<code>public void setY1(double y1) { this.y1 = y1; }</code>	// Setter untuk mengubah nilai y1.
13	[Getter dan Setter untuk x2, y2, x3, y3 dibuat dengan pola yang sama...]	// Getter dan setter untuk koordinat lainnya dibuat dengan pola yang sama seperti x1 dan y1.
14	<code>public double hitungSisi(double xa, double ya, double xb, double yb) {</code>	// Method untuk menghitung panjang sisi antara dua titik berdasarkan rumus Euclidean.
15	<code>return Math.sqrt(Math.pow(xa - xb, 2) + Math.pow(ya - yb, 2));</code>	// Menggunakan rumus $\sqrt{(xa - xb)^2 + (ya - yb)^2}$ untuk menghitung panjang sisi segitiga.
16	<code>return hitungSisi(x1, y1, x2, y2) + hitungSisi(x2, y2, x3, y3) + hitungSisi(x3, y3, x1, y1);</code>	// hitungSisi() untuk menjumlahkan panjang ketiga sisi segitiga.
17	<code>public double hitungLuas() {</code>	// untuk menghitung luas segitiga menggunakan rumus Heron.
18	<code>double s = hitungKeliling() / 2;</code>	// Menghitung semi-perimeter (setengah dari keliling segitiga).
19	<code>return Math.sqrt(s * (s - hitungSisi(x1, y1, x2, y2)) * (s - hitungSisi(x2, y2, x3, y3)) * (s - hitungSisi(x3, y3, x1, y1)));</code>	// rumus Heron untuk menghitung luas segitiga: $\sqrt{s(s-a)(s-b)(s-c)}$

Praktikum kasus 4

A. DIAGRAM CLASS



B. LISTING PROGRAM

```
9  * @author Calvin Pati
10  */
11  public class Kalkulator {
12      // Deklarasi variabel private untuk menyimpan 2 bilangan
13      private int a; // Menyimpan bilangan pertama
14      private int b; // Menyimpan bilangan kedua
15
16      // Constructor untuk inialisasi nilai awal
17      public Kalkulator(int a, int b) {
18          this.a = a; // Mengisi nilai a dari parameter ke variabel a
19          this.b = b; // Mengisi nilai b dari parameter ke variabel b
20      }
21
22      // Method untuk operasi penjumlahan
23      public int penjumlahan() {
24          return a + b; // Mengembalikan hasil penjumlahan a dan b
25      }
26
27      // Method untuk operasi pengurangan
28      public int pengurangan() {
29          return a - b; // Mengembalikan hasil pengurangan a dan b
30      }
31
32      // Method untuk operasi perkalian
33      public int perkalian() {
34          return a * b; // Mengembalikan hasil perkalian a dan b
35      }
36
37      // Method untuk operasi pembagian
38      public int pembagian() {
39          return a / b; // Mengembalikan hasil pembagian a oleh b
40      }
41  }
```

```
11  * @author Calvin Pati
12  */
13  public class KelkulatorMain {
14      // Method utama sebagai entry point program
15      public static void main(String[] args) {
16          // Membuat objek Scanner untuk membaca input dari keyboard
17          Scanner keyboard = new Scanner(System.in);
18
19          // Meminta input bilangan pertama dari user
20          System.out.print("Masukkan Bilangan a: ");
21          // Membaca input integer dan menyimpannya ke variabel a
22          int a = keyboard.nextInt();
23
24          // Meminta input bilangan kedua dari user
25          System.out.print("Masukkan Bilangan b: ");
26          // Membaca input integer dan menyimpannya ke variabel b
27          int b = keyboard.nextInt();
28
29          // Membuat objek kalkulator dengan input a dan b
30          Kalkulator kalkulator = new Kalkulator(a, b);
31
32          // Menampilkan hasil penjumlahan
33          System.out.println("Hasil penjumlahan: " + kalkulator.penjumlahan());
34
35          // Menampilkan hasil pengurangan
36          System.out.println("Hasil pengurangan: " + kalkulator.pengurangan());
37
38          // Menampilkan hasil perkalian
39          System.out.println("Hasil perkalian : " + kalkulator.perkalian());
40
41          // Menampilkan hasil pembagian
42          System.out.println("Hasil pembagian : " + kalkulator.pembagian());
43      }
```

C. OUTPUT

- Setelah program nya saya perbaiki dan saya maka tidak ada lagi yang error dari program

```
--- resources:3.3.1:resources (default-resources) @ Modul2FixPBO ---
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\Ne

--- compiler:3.13.0:compile (default-compile) @ Modul2FixPBO ---
Recompiling the module because of changed source code.
Compiling 8 source files with javac [debug release 23] to target\classes

--- exec:3.1.0:exec (default-cli) @ Modul2FixPBO ---
Masukkan Bilangan a: 10
Masukkan Bilangan b: 5
Hasil penjumlahan: 15
Hasil pengurangan: 5
Hasil perkalian : 50
Hasil pembagian : 2

-----
BUILD SUCCESS
-----

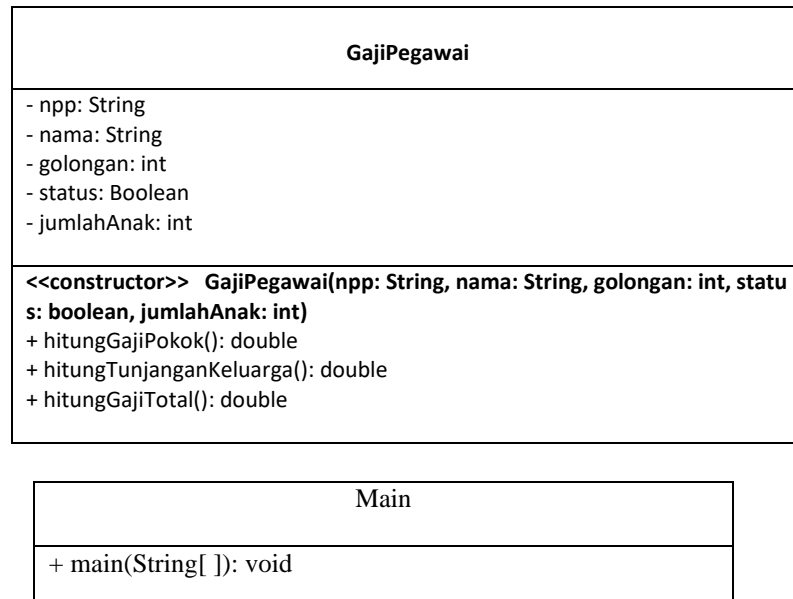
Total time: 8.622 s
Finished at: 2025-03-25T23:00:42+07:00
-----
```

D. ANALISA

No	Syntax	penjelasan
1	public class Kalkulator {	// kelas Kalkulator, yang merepresentasikan operasi dasar kalkulator.
2	private int a;	// variabel privat a untuk menyimpan bilangan pertama.
3	private int b	// Mendeklarasikan variabel privat b untuk menyimpan bilangan kedua.
4	public Kalkulator(int a, int b) {	// Konstruktor untuk menginisialisasi nilai a dan b saat objek dibuat.
5	this.a = a	// Mengisi nilai a dari parameter ke atribut a dalam kelas.
6	this.b = b;	//Mengisi nilai b dari parameter ke atribut b dalam kelas.
7	public int penjumlahan() {	// Method untuk melakukan operasi penjumlahan antara a dan b.
8	return a + b; public int pengurangan() {	//Mengembalikan hasil penjumlahan a dan b //Method untuk melakukan operasi pengurangan antara a dan b.
9	return a - b; public int perkalian() {	//Mengembalikan hasil pengurangan a dan b. //Method untuk melakukan operasi perkalian antara a dan b.
10	return a * b; public int pembagian() { return a / b;	//Mengembalikan hasil perkalian a dan b. //Method untuk melakukan operasi pembagian antara a dan b //engembalikan hasil pembagian a oleh b. Catatan: Perlu dicek agar b tidak bernilai nol untuk menghindari error.

Praktikum kasus 5

A. Diagram Kelas



B. LISTING PROGRAM

```
Source History
11 public class GajiPegawai {
12     // Atribut untuk menyimpan NPP pegawai
13     private String npp;
14     // Atribut untuk menyimpan nama pegawai
15     private String nama;
16     // Atribut untuk menyimpan golongan pegawai
17     private int golongan;
18     // Atribut untuk menyimpan status menikah pegawai
19     private boolean status; // true jika menikah, false jika tidak
20     // Atribut untuk menyimpan jumlah anak pegawai
21     private int jumlahAnak;
22
23     // Konstruktor untuk menginisialisasi atribut
24     public GajiPegawai(String npp, String nama, String status, int golongan, int jumlahAnak) {
25         this.npp = npp; // Mengatur NPP pegawai
26         this.nama = nama; // Mengatur nama pegawai
27         this.golongan = golongan; // Mengatur golongan pegawai
28         this.status = status; // Mengatur status pegawai
29         this.jumlahAnak = jumlahAnak; // Mengatur jumlah anak pegawai
30     }
31
32     // Metode untuk menghitung gaji pokok berdasarkan golongan
33     public double hitungGajiPokok() {
34         // Variabel untuk menyimpan gaji pokok
35         double gajiPokok = 0;
36
37         // Menentukan gaji pokok berdasarkan golongan
38         switch (golongan) {
39             case 1:
40                 gajiPokok = 1300000; // Gaji pokok untuk golongan 1
41                 break;
42             case 2:
43                 gajiPokok = 1750000; // Gaji pokok untuk golongan 2
44                 break;
45             case 3:
46                 gajiPokok = 2000000; // Gaji pokok untuk golongan 3
47                 break;
48             default:
49                 // Jika golongan tidak valid, kembalikan 0
50                 gajiPokok = 0;
51                 break;
52         }
53
54         return gajiPokok; // Mengembalikan nilai gaji pokok
55     }
56
57     // Metode untuk menghitung tunjangan keluarga
58     public double hitungTunjanganKeluarga() {
59         // Jika pegawai menikah, hitung tunjangan
60         if (status) {
61             return 300000 + (jumlahAnak * 100000); // Tunjangan keluarga
62         } else {
63             return 0; // Tidak ada tunjangan jika tidak menikah
64         }
65     }
66
67     // Metode untuk menghitung gaji total
68     public double hitungGajiTotal() {
69         // Menghitung gaji total dengan menjumlahkan gaji pokok dan tunjangan keluarga
70         return hitungGajiPokok() + hitungTunjanganKeluarga();
71     }
72 }
73
74
```

```
public class GajiPegawaiMain {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);

        System.out.print("Masukkan NPP anda:");
        String NPP = keyboard.nextLine();
        System.out.print("Masukkan nama anda:");
        String nama = keyboard.nextLine();
        System.out.print("Masukkan status anda:");
        String status = keyboard.nextLine();
        System.out.print("Masukkan golongan 1,2,3 anda:");
        int golongan = keyboard.nextInt();
        System.out.print("Masukkan Jumlah anak anda:");
        int jumlahanak = keyboard.nextInt();

        GajiPegawai gaji = new GajiPegawai(NPP, nama, status, golongan, jumlahanak);

        System.out.println("Gaji pokok : " + gaji.hitungGajiPokok());
        System.out.println("Tunjangan Keluarga : " + gaji.hitungTunjanganKeluarga());
        System.out.println("Gaji Total : " + gaji.hitungGajiTotal());
    }
}
```

C. Output

- Setelah program nya saya perbaiki dan saya maka tidak ada lagi yang error dari program

```
--- resources:3.3.1:resources (default-resources) @ Modul2FixPBO ---
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\Ne

--- compiler:3.13.0:compile (default-compile) @ Modul2FixPBO ---
Recompiling the module because of changed source code.
Compiling 10 source files with javac [debug release 23] to target\classes

--- exec:3.1.0:exec (default-cli) @ Modul2FixPBO ---
Masukkan NPP anda:200
Masukkan nama anda:celvin
Masukkan status anda:menikah
Masukkan golongan 1,2,3 anda:1
Masukkan Jumlah anak anda:2
Gaji pokok           : 1300000.0
Tunjangan Keluarga  : 0.0
Gaji Total           : 1300000.0

-----
BUILD SUCCESS
-----

Total time:  19.840 s
Finished at: 2025-03-25T23:36:40+07:00
-----
█
```


E. ANALISA

No	Syntax	penjelasan
1	<pre>private String npp; private String nama; private int golongan; private boolean status; private int jumlahAnak;</pre>	// Menyimpan informasi pegawai seperti NPP, nama, golongan, status menikah, dan jumlah anak.
2	<pre>public GajiPegawai(String npp, String nama, int golongan, boolean status, int jumlahAnak) { ... }</pre>	// Menginisialisasi atribut dengan nilai yang diberikan saat objek dibuat.
3	<pre>public double hitungGajiPokok() { ... }</pre>	// Mengembalikan gaji pokok berdasarkan golongan pegawai.
4	<pre>public double hitungTunjanganKeluarga() { ... }</pre>	// Mengembalikan tunjangan keluarga jika pegawai menikah.
5	<pre>public double hitungGajiTotal() { ... }</pre>	// Mengembalikan total gaji dengan menjumlahkan gaji pokok dan tunjangan keluarga.