



# Pemrograman Berorientasi Objek (PBO)

# Capaian Pembelajaran Mata Kuliah

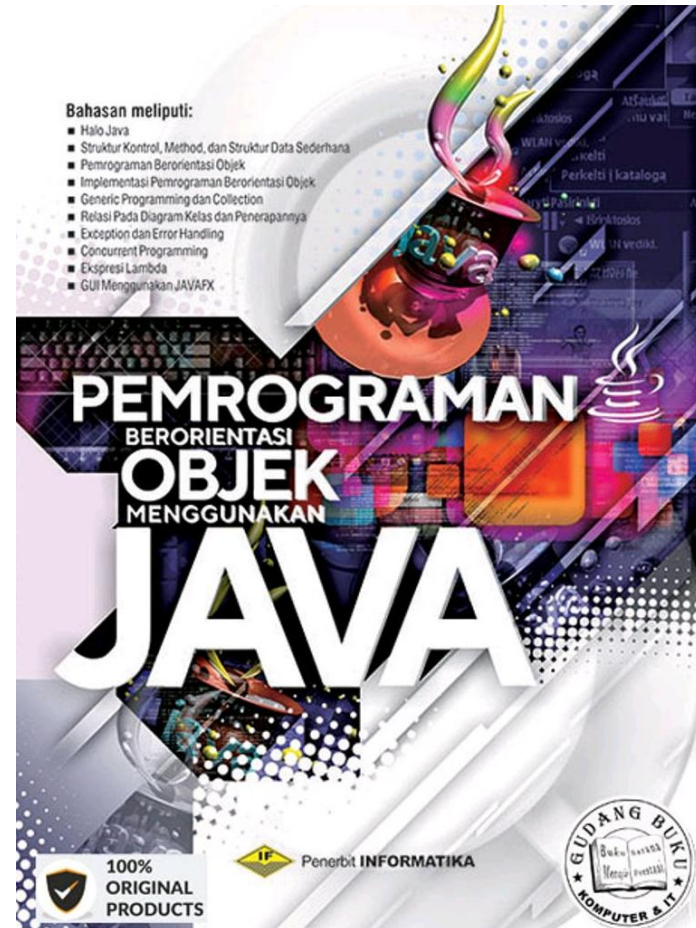
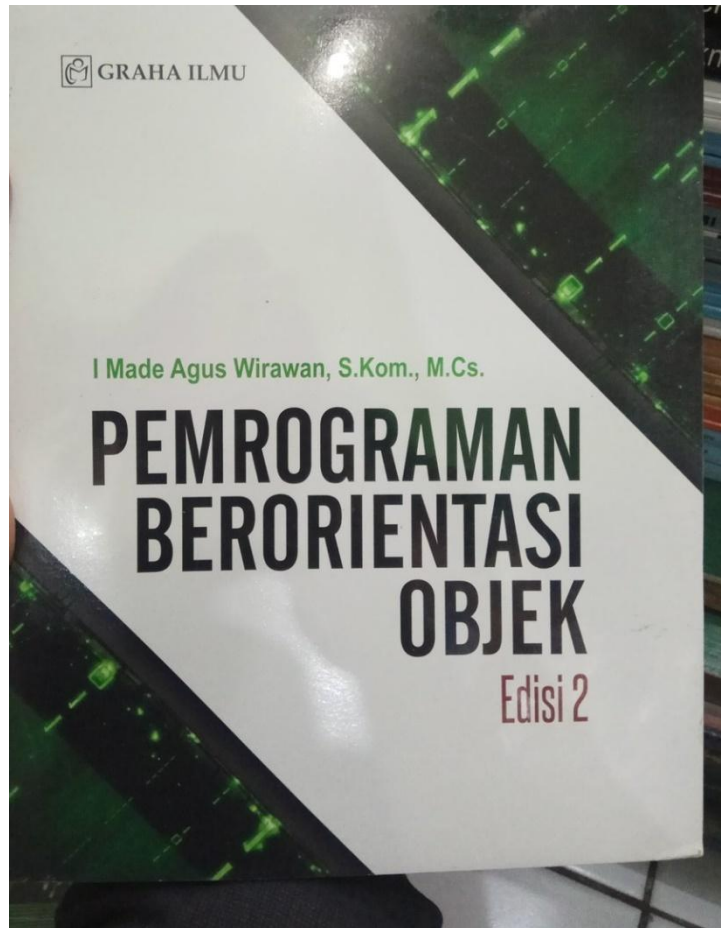
- Mahasiswa mengerti akan manfaat, fungsi, peranan konsep pemrograman berorientasi objek dalam pengembangan aplikasi di bidang TI (Teknik Informatika)/ SI (Sistem Informasi) dan Bisnis Digital untuk menjebatani kebutuhan bisnis, perusahaan (enterprise) dalam era industri 4.0.



# **PENGANTAR PEMROGRAMAN BERORIENTASI OBJEK**

**Pertemuan ke-1**

**U N I V E R S I T A S   B U N D A   M U L I A**



# Sub-CPMK

- *Mahasiswa mampu menjelaskan konsep struktur (prosedural) dan konsep PBO dan perbandingan konsep struktur Vs konsep PBO. (C2, A2).*

## Materi

1. Sekilas Konsep Prosedural / Struktur (Non Objek).
2. Sejarah Singkat Konsep Pemrograman Berorientasi Objek.
3. Konsep Pemrograman Berorientasi Objek.
4. Konsep Non Objek Vs Konsep Berorientasi Objek.



# 1. Sekilas Konsep Prosedural / Struktur (Non Objek)

- Dalam membuat program dapat menggunakan konsep prosedural (struktur) / non objek dan dengan konsep berorientasi objek.
- Konsep berorientasi objek semakin berkembang dalam pengembang aplikasi, hal ini dikarenakan seiring dengan didukungnya konsep pemrograman berorientasi objek secara penuh oleh bahasa-bahasa pemrograman saat ini.

# 1.1 Konsep Prosedural

- Pada konsep pemrograman prosedural penelusuran program bersifat dari atas ke bawah yang dikenal dengan top-down, dan menggunakan teknik dekomposisi yaitu: dimulai dari gambaran global kemudian program dipecah-pecah ke dalam prosedur (sub prosedur) dan fungsi.
- Teknik dekomposisi yang menjadi ciri dari konsep pemrograman prosedural.



## 1.2 Cara Pandang Prosedural

- Pada konsep pemrograman prosedural, melihat sebuah program sebagai suatu prosedur besar yang kemudian dapat dipecah-pecah (dekomposisi) menjadi prosedur-prosedur (sub prosedur) yang lebih kecil dan saling berinteraksi satu sama lainnya.
- Bahasa pemrograman yang mendukung konsep prosedural seperti: COBOL, FORTRAN, C, BASIC, PASCAL, ADA, dan yang lainnya.



## 2. Sejarah Singkat Pemrograman Berorientasi Objek

## 2.1 Sejarah Singkat OOP

- Konsep pemrograman berorientasi objek sudah ada sejak sekitar tahun 1960-an.
- Pertama kali konsep berorientasi objek diperkenalkan oleh Ole Johan Dahl dan Kristen Nygaard dari Universitas Oslo, Norwegia pada tahun 1966 dalam sebuah jurnal yang berjudul “SIMULA An Algol Based Simulation Language”.

## 2.1 Sejarah Singkat OOP (Lanj...)

- Pada tahun 1970-an SMALLTALK dikembangkan oleh XEROX PARC. Pada tahun 1970-an banyak pengembang aplikasi menggunakan bahasa pemrograman COBOL dan FORTRAN untuk membuat program.
- Kedua bahasa pemrograman ini menggunakan konsep dekomposisi dan fungsional.

## 2.1 Sejarah Singkat OOP (Lanj...)

- Pada tahun-tahun tersebut konsep pemrograman dekomposisi (prosedural) sedang trend.
- Bahasa pemrograman yang ada saat itu kebanyakan belum bisa mendukung secara penuh konsep berorientasi objek.
- Konsep berorientasi objek mulai populer dimana analisis dan desain menjadi lebih diperhatikan daripada program (coding).

## 2.2 Pelopor OOP



Ole Johan Dahl  
12-10-1931 s/d  
29-06-2002

- Ole Johan Dahl, Ole Johan Dahl (12 Oktober 1931 - 29 Juni 2002) adalah ilmuwan komputer berkebangsaan Norwegia dan merupakan salah satu tokoh bahasa pemrograman SIMULA dan konsep pemrograman berorientasi objek bersama-sama dengan rekannya yaitu Kristen Nygaard.

## 2.2 Pelopor OOP (Lanj...)



Kristen Nygaard  
27-08-1926 s/d  
10-08-2002.

- Kristen Nygaard, Kristen Nygaard (27 Agustus 1926 - 10 Agustus 2002), merupakan ilmuwan komputer berkebangsaan Norwegia dan merupakan salah satu tokoh bahasa pemrograman SIMULA dan konsep pemrograman berorientasi objek bersama-sama dengan rekannya yaitu Ole Johan Dahl.



### 3. Konsep Pemrograman Berorientasi Objek



- Pada konsep pemrograman berorientasi objek, membagi program menjadi objek-objek yang saling berinteraksi satu sama lain.
- Objek dapat berupa benda, dapat berwujud nyata maupun tidak berwujud nyata.

**Objek mobil**



# Lanj...

- Bahasa-bahasa pemrograman yang mendukung konsep pemrograman berorientasi objek seperti: C++, Visual Basic.Net, Visual C++, Visual C#, Java, Delphi, Visual Foxpro, PHP, Simula, Smalltalk, Ruby, Python, Eiffel, Perl, Adobe Flash AS 3.0, dan yang lainnya.

# Lanj...

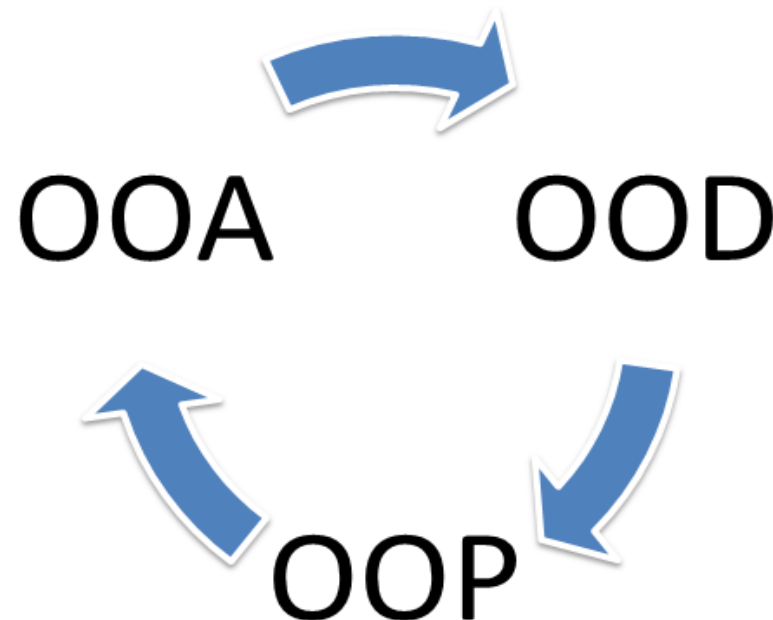
- Sebelum membuat aplikasi dilakukan analisis dan desain sistem. Pada konsep berorientasi objek dilakukan **OOA** (*Object Oriented Analysis*), **OOD** (*Object Oriented Design*).
- Diagram yang digunakan untuk desain sistem pada berorientasi objek seperti: *use case diagram, class diagram, statechart diagram, activity diagram, sequence diagram, collaboration diagram, component diagram, deployment diagram.*

## Lanj...

- Sehingga OOA, OOD dan **OOP** (*Object Oriented Programming*) merupakan satu kesatuan dalam pengembangan aplikasi dengan pendekatan berorientasi objek.
- Selama pengembangan sistem dapat menggunakan salah satu tahapan pengembangan sistem, seperti: Waterfall, SCRUM, XP, dan yang lainnya. Sehingga membentuk daur hidup pengembangan sistem secara utuh (holistik).

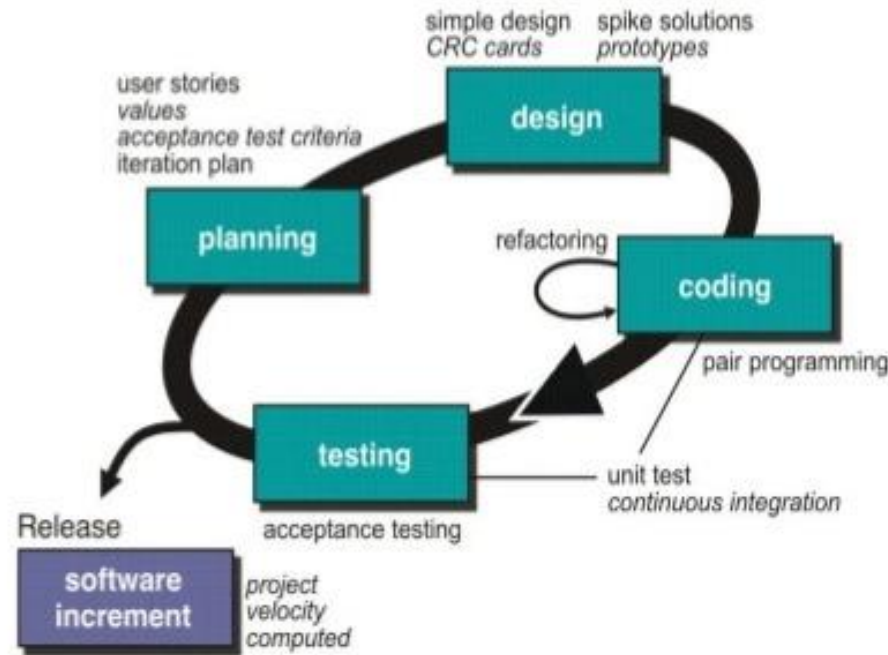
# Lanj...

- Keterkaitan OOA, OOD, dan OOP dalam pengembangan aplikasi dengan konsep berorientasi objek.



# Lanj...

- Salah satu contoh tahapan pengembangan sistem seperti eXtreme Programming (XP).



Sumber: I Gusti Ngurah Suryantara, 2017

## 3.1 Pengertian Objek

- Objek adalah benda (dalam dunia nyata), baik yang berwujud nyata maupun yang tidak nyata (seperti sistem).
- Objek dalam dunia nyata dapat berupa: komputer, televisi, mobil, motor, manusia, kuda, dan lainnya.
- Pada program objek dapat berupa: form, basis data, report, dan lainnya.

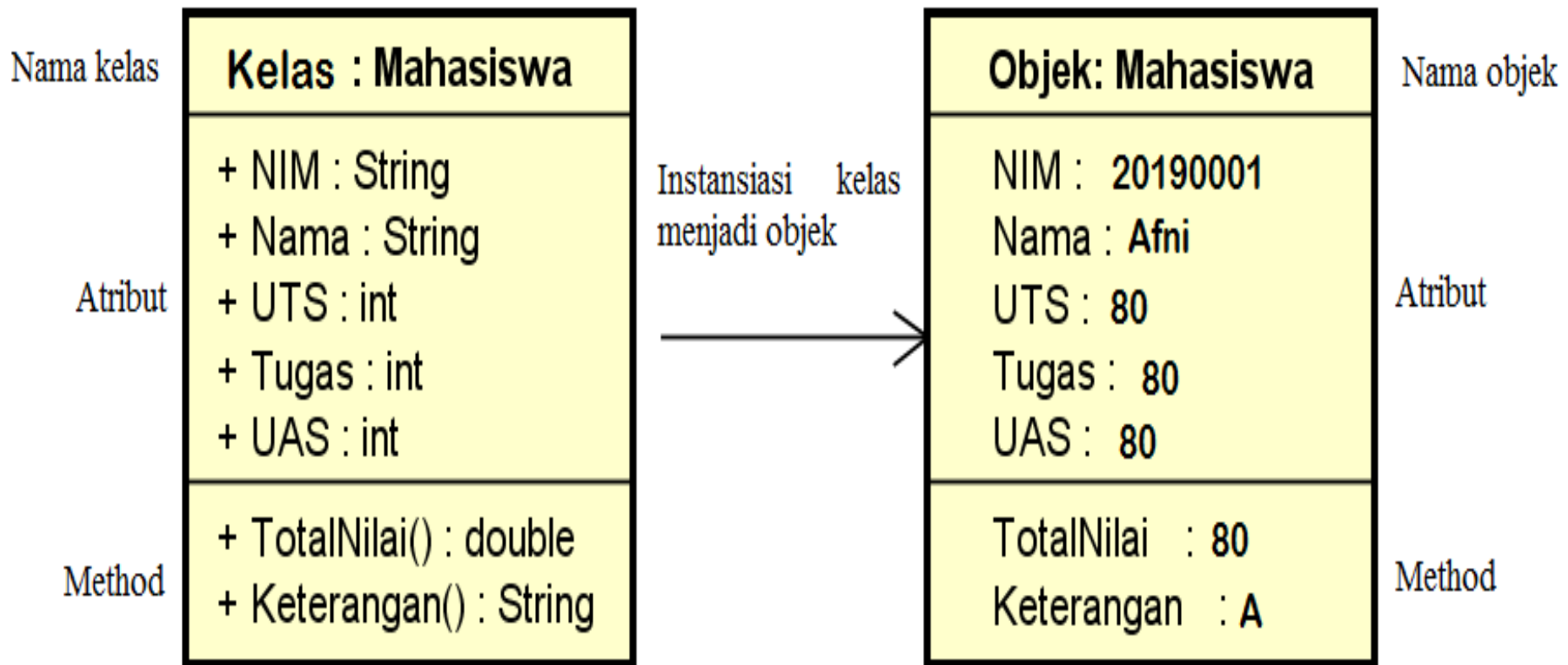
## 3.1 Pengertian Objek (Lanj...)

- Objek memiliki dua karakteristik yaitu:
  - Ciri-ciri (state / atribut).
  - Perilaku (behavior / method).
- Misalkan objek manusia memiliki state (nama, tinggi, berat, warna kulit, dan lainnya), behavior (menangis, tertawa, tersenyum, berjalan, dan lainnya).



## 3.1 Pengertian Objek (Lanj...)

- Perhatikan diagram kelas dan objek berikut.





## 4. Konsep Non Objek Vs Berorientasi Objek

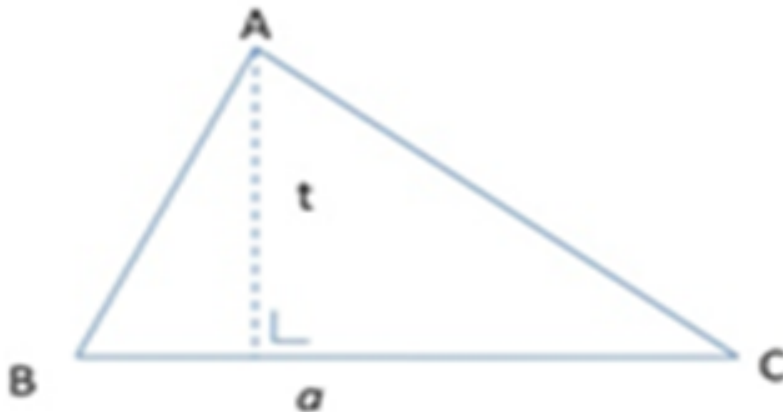
- Pada pendekatan non objek yang menjadi perhatian adalah: pembuatan spesifikasi dan dekomposisi sistem secara fungsional.
- Sedangkan pada pendekatan berorientasi objek menekankan pada saat melakukan indentifikasi objek dari domain aplikasi, kemudian pembuatan method yang sesuai.

# Lanj...

- Keuntungan konsep PBO adalah:
  - Natural.
  - Reliable.
  - Modularity.
  - Information-Hiding.
  - Maintainability.
  - Extensibility.
  - Reusability.
  - Pluggability & Debugging Ease.
  - Efisiensi.

## 4.1 Analisis Kasus

- Berikut menghitung luas segitiga dengan konsep prosedural dan berorientasi objek, sehingga memiliki pemahaman pada kedua konsep tersebut.



$$L = \frac{1}{2} at$$

## 4.2 Non Objek

Berikut menghitung luas segitiga dengan konsep prosedural.

### Program: Lat\_prosedural

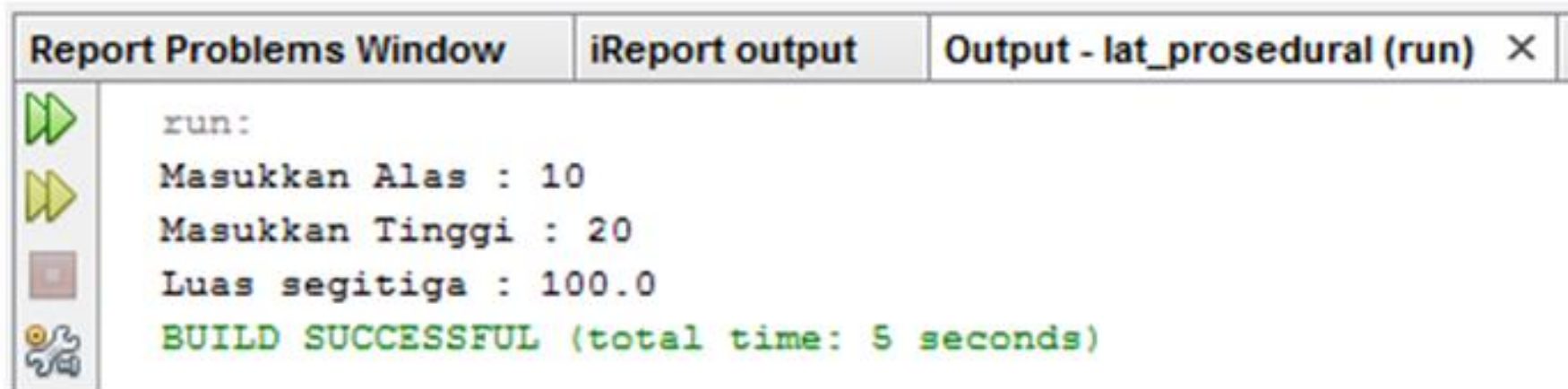
```
1. package lat_prosedural;
2. import java.util.Scanner;
3.
4. public class Lat_prosedural {
5.     /**
6.      * @param args the command line arguments
7.      */
8.
9.     //membuat fungsi
10.    static double LuasSegitiga(int mAlas, int mTinggi)
11.    {
12.        return 0.5 * mAlas * mTinggi;
13.    }
14.
```

## 4.2 Non Objek (Lanj...)

```
15.     public static void main(String[] args)
16.     {
17.         //mendeklarasikan variabel
18.         int Tinggi;
19.         int Alas;
20.         Double Luas;
21.
22.         // TODO code application logic here
23.         Scanner input = new Scanner (System.in);
24.
25.         System.out.printf("Masukkan Alas : ");
26.         Alas = input.nextInt();
27.         System.out.printf("Masukkan Tinggi : ");
28.         Tinggi = input.nextInt();
29.         Luas = LuasSegitiga(Alas, Tinggi);
30.         System.out.println("Luas segitiga : "+ Luas);
31.     }
32. }
```

## 4.2 Non Objek (Lanj...)

- Contoh keluaran program



The screenshot shows a window titled "Output - lat\_prosedural (run)" with a close button (X). The window contains the following text:

```
run:
Masukkan Alas : 10
Masukkan Tinggi : 20
Luas segitiga : 100.0
BUILD SUCCESSFUL (total time: 5 seconds)
```

On the left side of the window, there are four icons: a green double arrow, a yellow double arrow, a red square, and a blue circular icon with a white 'e'.



## 4.3 Konsep PBO

- Berikut menghitung luas segitiga dengan konsep berorientasi objek.
- Membuat kelas dan diinstansiasi menjadi objek.

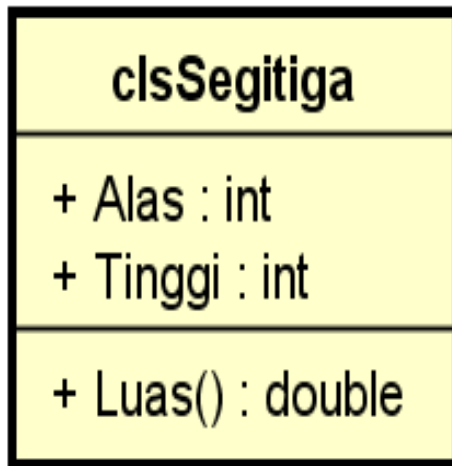


Diagram Kelas / Class Diagram



Diagram objek / Object Diagram

## 4.3 Konsep PBO (Lanj...)

**Nama class: clsSegitiga**

```
1.  package segitiga;
2.  import java.util.Scanner;
3.
4.  //menciptakan kelas
5.  public class clsSegitiga
6.  {
7.      //membuat atribut
8.      int Alas;
9.      int Tinggi;
10.
11.     //membuat method
12.     public double Luas()
13.     {
14.         return 0.5 * Alas * Tinggi;
15.     }
16. }
```






## 4.3 Konsep PBO (Lanj...)

### Main Program: Segitiga

```
1.  package segitiga;
2.
3.  import java.util.Scanner;
4.
5.  public class Segitiga {
6.
7.  public static void main(String[] args)
8.  {
9.      // TODO code application logic here
10.     Scanner input = new Scanner (System.in);
11.
12.     //membuat objek dari clsSegitiga
13.     clsSegitiga objSegitiga = new clsSegitiga();
14.
```

## 4.3 Konsep PBO (Lanj...)

```
15.         //masukkan nilai alas dan tinggi
16.         System.out.printf("Masukkan Alas : ");
17.         objSegitiga.Alas = input.nextInt();
18.         System.out.printf("Masukkan Tinggi : ");
19.         objSegitiga.Tinggi = input.nextInt();
20.
21.         //Cetak luas segitiga
22.         System.out.printf("Luas segitiga : %.2f\n",
    objSegitiga.Luas());
23.     }
24. }
```

Report Problems Window	iReport output	Output - lat_prosedural (run) X
	run:	
	Masukkan Alas : 10	
	Masukkan Tinggi : 20	
	Luas segitiga : 100.0	
	BUILD SUCCESSFUL (total time: 5 seconds)	

# Ringkasan:

- Konsep pemrograman berorientasi objek terus berkembang dengan pesat, seiring dengan didukungnya secara penuh oleh bahasa-bahasa pemrograman yang ada dewasa ini.
- Konsep pemrograman berorientasi objek pertama kali diperkenalkan oleh **Ole Johan Dahl** dan **Kristen Nygaard**.

# Ringkasan: (Lanj...)

- Empat pilar utama konsep pemrograman berorientasi objek yaitu: **Abstraksi, Pewarisan, Pembungkusan, dan Banyak bentuk.**
- Pada konsep pemrograman berorientasi objek menekankan pembuatan kelas dan menginstansiasi kelas menjadi objek, hal ini berdasarkan abstraksi dunia nyata.

# Ringkasan: (Lanj...)

- Pada pendekatan berorientasi objek menekankan pada saat melakukan indentifikasi objek dari domain aplikasi, kemudian pembuatan prosedur dan fungsi yang sesuai.

# Latihan Mandiri

1. Jelaskan pendekatan pada konsep pemrograman prosedural atau terstruktur (non objek) dan pada konsep berorientasi objek.
2. Jelaskan secara singkat mulai kapan muncul paradigma pemrograman berorientasi objek.
3. Jelaskan apa yang dimaksud dengan kelas.
4. Jelaskan apa yang dimaksud dengan objek.



# Latihan Mandiri (Lanj...)

5. Jelaskan empat hal (pilar) utama konsep pemrograman berorientasi objek.
6. Jelaskan secara singkat perbedaan pada non objek dan berorientasi objek.
7. Keuntungan apa yang didapatkan dengan konsep berorientasi objek.
8. Jelaskan secara singkat tingkatan bahasa pemrograman dan generasi bahasa pemrograman.

# Latihan Mandiri (Lanj...)

9. Jelaskan apa yang dimaksud dengan atribut.
10. Jelaskan apa yang dimaksud dengan method.
11. Buatlah program dengan konsep berorientasi objek untuk menghitung luas lingkaran.
12. Buatlah program dengan konsep berorientasi objek untuk menghitung volume tabung.



**TERIMA KASIH**

**U N I V E R S I T A S   B U N D A   M U L I A**