

LAPORAN PRAKTIKUM
Pertemuan 1
MODUL 1 : PENGENALAN PBO



Disusun Oleh :
Ahmad Saiful Huda
(19102158)

Dosen Pengampu :
Merlinda Wibowo, S.T., M. Phil.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2021

MODUL I

Pengenalan Environment OOP

I. TUJUAN

- a. Memahami lingkungan dasar NetBeans IDE serta cara meng-compile dan menjalankan program
- b. Membandingkan pemrograman terstruktur dengan pemrograman berorientasi objek dengan membuat program dalam bahasa Java.

II. TOOL

- a. NetBeans IDE 8.1 atau
- b. NetBeans IDE 12.3
- c. Java SE Development Kit 8

III. DASAR TEORI

A. Pengantar Pemrograman Berorientasi Objek

Apa itu OOP? Merupakan teknik membuat suatu program berdasarkan objek dan apa yang bisa dilakukan objek tersebut. Object-oriented program terdiri dari objek-objek yang berinteraksi satu samalain untuk menyelesaikan sebuah tugas. Kenapa menggunakan OOP? Kode-kode di-breakdown agar lebih mudah di-manage. Breakdown berdasarkan objek-objek yang ada pada program tersebut. Dianjurkan diimplementasikan untuk program dengan berbagai ukuran karena lebih mudah untuk men-debug. Pemrograman procedural mengatur program dalam barisan-barisan linier yang bekerja dari atas ke bawah. Kumpulan tahapan yang dijalankan setelah yang lain berjalan. Baik untuk program kecil yang berisi sedikit code. Tidak dianjurkan diimplementasikan pada program berukuran besar, karena susah untk dimanage dan di-debug OOP mempunyai karakteristik sebagai berikut :

1. Abstraksi Menemukan hal-hal yang penting pada suatu objek dan mengabaikan hal-hal yang sifatnya insidental.
2. Enkapsulasi Pengkapsulan adalah proses pemaketan data objek bersama method-methodnya

3. Pewarisan (Inheritance) Proses penciptaan kelas baru (subclass/kelas turunan) dengan mewarisi karakteristik dari kelas yang sudah ada (superclass/kelas induk), ditambah karakteristik unik kelas baru itu.
4. Reuseability Reuseability adalah kemampuan untuk menggunakan kembali kelas yang sudah ada.
5. Polymorphism Polymorphism berasal dari bahasa Yunani yang berarti banyak bentuk.

B. PENGENALAN CLASS, OBJECT & ENKAPSULASI

a. Class

Class merupakan cetak biru (blue print) dari objek atau dengan kata lain sebuah Class menggambarkan ciri-ciri objek secara umum. Sebagai contoh Suzuki Smash, Yamaha VegaR, Honda SupraFit, dan Kawasaki KazeR merupakan objek dari Class sepeda motor. Suzuki Smash dan objek lainnya juga memiliki kesamaan atribut (merk, tipe, berat, kapasitas bensin, tipe mesin, warna, harga) dan method untuk mengakses data pada atributnya (misal fungsi untuk menginputkan data merk, tipe, berat, dsb serta fungsi untuk mencetak data merk, tipe, berat, dsb).

Contoh :

```
1 public class SepedaMotor {
2     private String merk;
3     private long harga;
4
5     public void setMerk(String merkMotor) {
6         merk = merkMotor;
7     }
8
9     public String getMerk(){
10        return merk;
11    }
12
13    public long Harga(long hargaMotor) {
14        return harga = hargaMotor;
15    }
16 }
```

b. Object

Objek (Object) merupakan segala sesuatu yang ada di dunia ini, yaitu manusia, hewan, tumbuhan, rumah, kendaraan, dan lain sebagainya. Contoh-contoh objek yang telah disebutkan diatas merupakan contoh objek nyata pada kehidupan kita. Pada pemrograman berorientasi objek, kita akan belajar bagaimana membawa konsep objek dalam kehidupan nyata menjadi objek dalam dunia pemrograman. Setiap objek dalam

dunia nyata pasti memiliki 2 elemen penyusunnya, yaitu keadaan (state) dan perilaku/sifat (behaviour). Sebagai contoh, sepeda memiliki keadaan yaitu warna, merk, jumlah roda, ukuran roda. Dan perilaku/sifat sepeda adalah berjalan, berhenti, belok, menambah kecepatan, mengerem. Pada saat objek diterjemahkan ke dalam konsep PBO, maka elemen penyusunnya juga terdiri atas 2 bagian, yaitu : Atribut, merupakan ciri-ciri yang melekat pada suatu objek (state). Method, merupakan fungsi-fungsi yang digunakan untuk memanipulasi nilai-nilai pada atribut atau untuk melakukan hal-hal yang dapat dilakukan suatu objek (behaviour). Objek dalam konsep PBO memiliki keadaan dan perilaku yang sama seperti halnya objek di dunia nyata, karena objek dalam konsep PBO merupakan representasi objek dari dunia nyata. Objek dalam PBO merepresentasikan keadaan melalui variabel-variabel (Atribut), sedangkan perilakunya direpresentasikan dengan method (yang merupakan suatu fungsi yang berhubungan dengan perilaku objek tersebut maupun berhubungan dengan atribut dari objek tersebut). Objek yang memiliki kesamaan atribut dan method dapat dikelompokkan menjadi sebuah Class. Dan objek-objek yang dibuat dari suatu class itulah yang disebut dengan Instant of class.

Untuk menginstansi (membuat) objek dari class, gunakan operator new. Sintaks membuat objek dari suatu class :

```
namaClass namaObjek = new namaClass()
```

c. Enkapsulasi

Enkapsulasi (encapsulation) merupakan cara untuk melindungi property (atribut) / method tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi oleh suatu bagian program. Cara untuk melindungi data yaitu dengan menggunakan access modifiers (hak akses). Ada 4 hak akses yang tersedia, yaitu default, public, protected, private.

Untuk lebih jelasnya, silahkan lihat kedua table berikut ini :

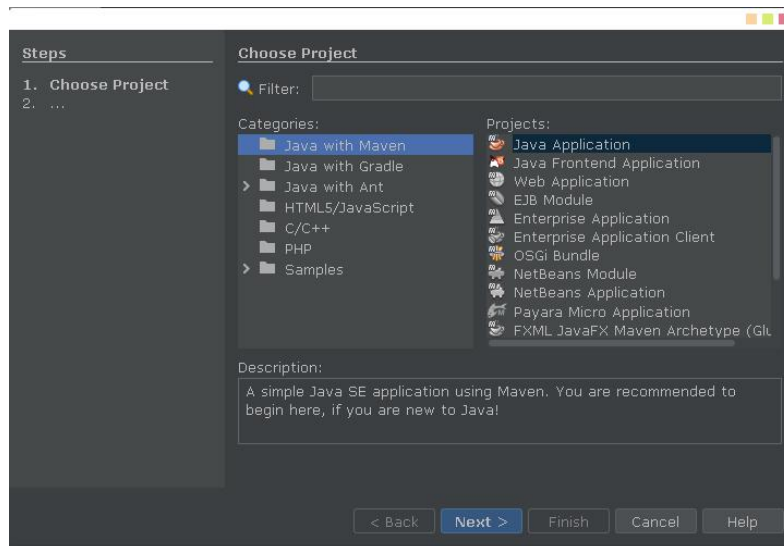
No	Modifier	Pada class dan interface	Pada method dan variabel
1	Default (tidak ada modifier)	Dapat diakses oleh yang sepaket	Diwarisi oleh subkelas dipaket yang sama, dapat diakses oleh method-method yang sepaket
2	Public	Dapat diakses dimanapun	Diwarisi oleh subkelasnya, dapat diakses dimanapun
3	Protected	Tidak bisa diterapkan	Diwarisi oleh subkelasnya, dapat diakses oleh method-method yang sepaket
4	private	Tidak bisa diterapkan	Tidak dapat diakses dimanapun kecuali oleh method-method yang ada dalam kelas itu sendiri

Aksesabilitas	public	private	protected	default
Dari kelas yang sama	Ya	Ya	Ya	Ya
Dari sembarang kelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari sembarang kelas di luar paket	Ya	Tidak	Tidak	Tidak
Dari subkelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari subkelas di luar paket	Ya	Tidak	Ya	Tidak

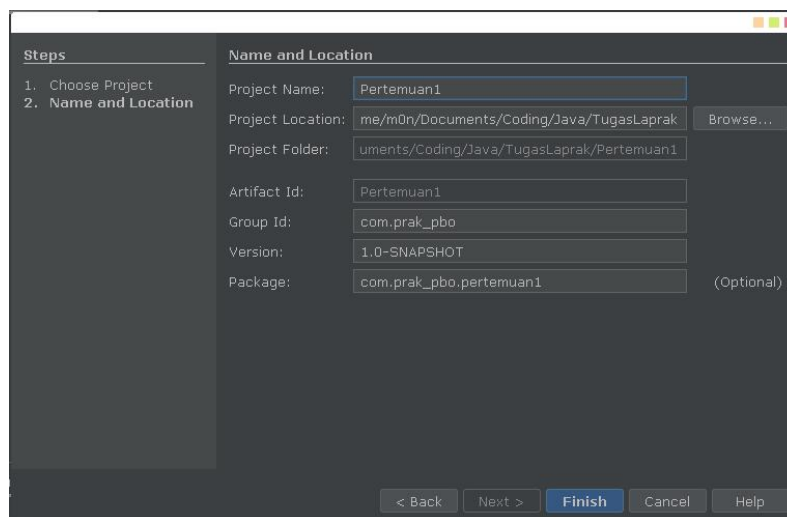
IV. Praktikum

A. GUIDED

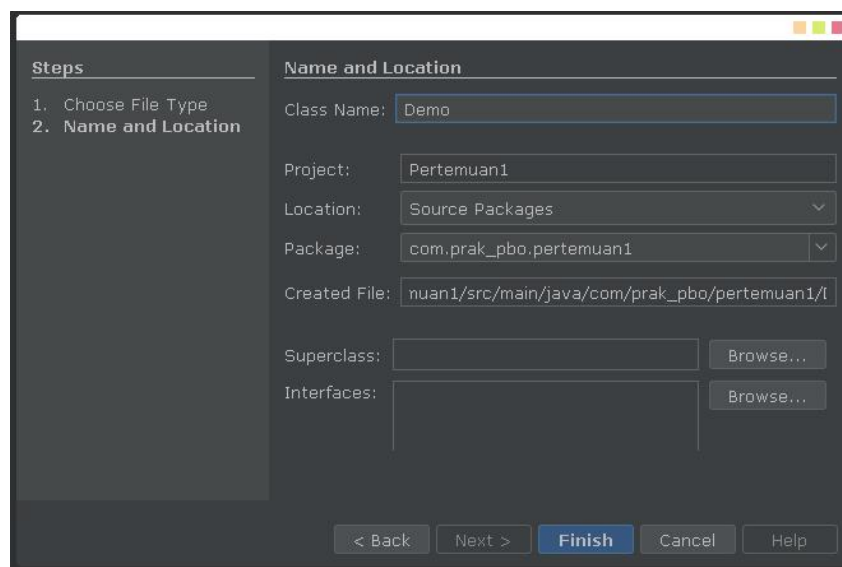
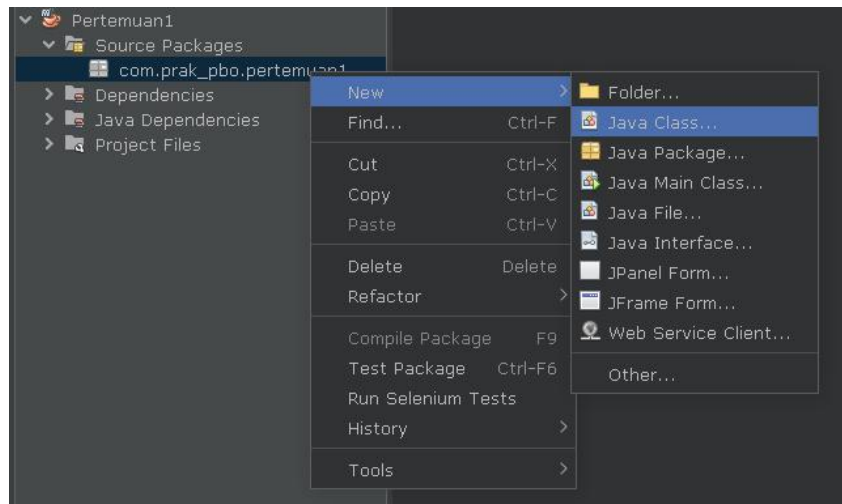
Pertama buat project terlebih dahulu, disini pilih project Java Application pada kategori 'Java with Maven'



Selanjutnya isi nama project dan lokasi.



Buat package baru, beri nama com.prak_pbo.pertemuan1, lalu buat class baru dan beri nama Demo.java



1. Program HelloWorld

Source Code

```

1  /**
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package com.m0n.pertemuan1;
7
8   /**
9    *
10   * @author m0n
11   */
12  public class DemoPrint {
13      public static void main(String[] args) {
14          System.out.println("Hello");
15          System.out.println("World");
16
17          System.out.print("Hello");
18          System.out.print("Dunia");
19      }
20  }
21
22

```

Output :

```

--- maven-compiler-plugin:3.1:compile (default-compile) @ pertemuan1 ---
Changes detected - recompiling the module!
Compiling 6 source files to /home/m0n/Documents/Coding/Java/pertemuan1/target/classes

--- exec-maven-plugin:3.0.0:exec (default-cli) @ pertemuan1 ---
Hello
World
HelloDunia
-----
BUILD SUCCESS
-----
Total time: 3.136 s
Finished at: 2021-04-13T11:18:12+07:00
-----

```

2. Program Bilangan Positif dan Negatif

Source Code

```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package com.m0n.pertemuan1;
7
8   import java.util.Scanner;
9
10  /**
11   *
12   * @author m0n
13   */
14  public class Demo2 {
15      public static void main(String[] args) {
16          Scanner input = new Scanner(System.in);
17
18          System.out.println("Masukan bilangan: ");
19          int bil = input.nextInt();
20
21          if(bil > 0)
22              System.out.println(bil + " adalah bilangan positif");
23          else if (bil < 0)
24              System.out.println(bil + " adalah bilangan negatif");
25          else
26              System.out.println(bil + " adalah nol!");
27      }
28  }
29
30

```

Output :

```

--- maven-compiler-plugin:3.1:compile (default-compile) @ pertemuan1 ---
Nothing to compile - all classes are up to date

--- exec-maven-plugin:3.0.0:exec (default-cli) @ pertemuan1 ---
Masukan bilangan:
4
4 adalah bilangan positif
-----
BUILD SUCCESS
-----
Total time: 6.587 s
Finished at: 2021-04-13T11:23:28+07:00
-----

```

3. Program Profil

Source Code


```

10  /**
11   *
12   * @author m0n
13   */
14  public class DemoScanner {
15      public static void main(String[] args) {
16          Scanner input = new Scanner(System.in);
17
18          System.out.println("==== Edit Profile =====");
19          System.out.println("");
20
21          System.out.println("Masukan nama: ");
22          String nama = input.nextLine();
23
24          System.out.println("Masukan inisial: ");
25          char initial = input.nextLine().charAt(0);
26
27          System.out.println("Masukan usia: ");
28          int usia = input.nextInt();
29
30          System.out.println("Masukan tinggi badan: ");
31          float tinggi = input.nextFloat();
32
33          System.out.println();
34          System.out.println("====> Profil Saya <====");
35          System.out.println();
36          System.out.println("Nama      : " + nama);
37          System.out.println("Inisial  : " + initial);
38          System.out.println("Usia     : " + usia + " tahun");
39          System.out.println("Tinggi   : " + tinggi + " cm");
40      }
41  }
42

```

Output :

```

--- exec-maven-plugin:3.0.0:exec (default-cli) @ pertemuan1 ---
==== Edit Profile =====

Masukan nama:
Adm0n
Masukan inisial:
A
Masukan usia:
21
Masukan tinggi badan:
174

====> Profil Saya <====

Nama      :Adm0n
Inisial   :A
Usia      :21 tahun
Tinggi    :174.0 cm
-----
BUILD SUCCESS
-----
Total time: 18.682 s
Finished at: 2021-04-13T11:29:30+07:00
-----

```

4. Program Looping

Source Code

```

10  * @author m0n
11  */
12  public class DemoLoop {
13      public static void main(String[] args) {
14          for (int i = 0; i <= 5; i++) {
15              System.out.println("For loop (" + i + "/5");
16              System.out.println("");
17          }
18
19          int j=1;
20          while(j<=5) {
21              System.out.println("While loop(" + j + "/5");
22              j++;
23          }
24          System.out.println("");
25
26          int k=1;
27          do {
28              System.out.println("While loop(" + k + "/5");
29              k++;
30          } while(k<=5);
31          System.out.println("");
32
33          int x=0;
34          while(x<9999) {
35              System.out.println("Looping... " + x + "/9999");
36              x++;
37
38              if(x == 5) {
39                  System.out.println("Finish..");
40                  break;
41              }
42          }
43      }
44  }
45
46  }

```

Output :

```

--- exec-maven-plugin:3.0.0:exec (default-cli) @ pertemuan1 ---
For loop (0/5

For loop (1/5

For loop (2/5

For loop (3/5

For loop (4/5

For loop (5/5

While loop(1/5
While loop(2/5
While loop(3/5
While loop(4/5
While loop(5/5

While loop(1/5
While loop(2/5
While loop(3/5
While loop(4/5
While loop(5/5

Looping... 0/9999
Looping... 1/9999
Looping... 2/9999
Looping... 3/9999
Looping... 4/9999
Finish..
-----
BUILD SUCCESS
-----
Total time: 1.134 s
Finished at: 2021-04-13T11:32:54+07:00
-----

```

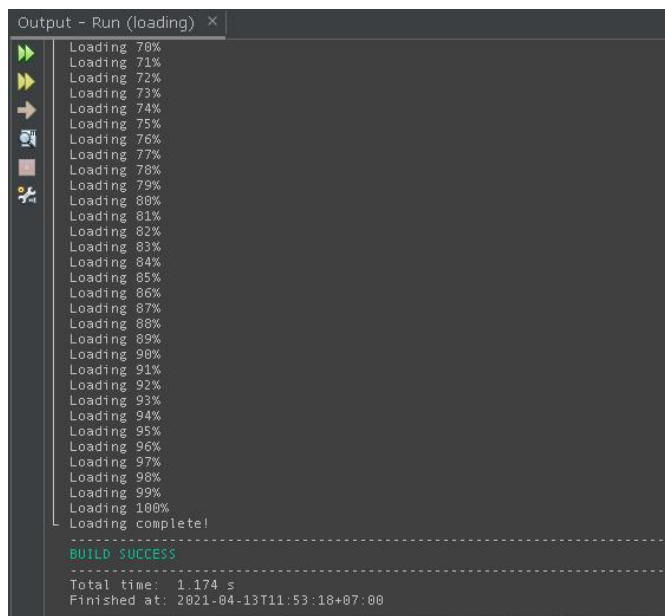
B. UNGUIDED

1. Buatlah looping “Loading x%”, dimana x adalah bilangan 1-100 dengan menggunakan for,while atau do-while! Di akhir looping, print “Loading complete!”

SourceCode :

```
6 package com.prak_pbo.pertemuan1;
7
8 /**
9  *
10  * @author m0n
11  */
12 public class loading {
13     public static void main(String[] args) {
14         System.out.println("==== Loading ====");
15         for (int i = 0; i <= 101; i++){
16             if (i > 100){
17                 System.out.println("Loading complete!");
18             }else {
19                 System.out.println("Loading " +i+ "%");
20             }
21         }
22     }
23 }
24
25
```

Output :



```
Output - Run (loading) X
Loading 70%
Loading 71%
Loading 72%
Loading 73%
Loading 74%
Loading 75%
Loading 76%
Loading 77%
Loading 78%
Loading 79%
Loading 80%
Loading 81%
Loading 82%
Loading 83%
Loading 84%
Loading 85%
Loading 86%
Loading 87%
Loading 88%
Loading 89%
Loading 90%
Loading 91%
Loading 92%
Loading 93%
Loading 94%
Loading 95%
Loading 96%
Loading 97%
Loading 98%
Loading 99%
Loading 100%
Loading complete!
-----
BUILD SUCCESS
-----
Total time: 1.174 s
Finished at: 2021-04-13T11:53:18+07:00
-----
```

Penjelasan :

Program menggunakan perulangan for dan perjabangan if else dimana program akan selesai jika sudah mencapai bilangan 100 dan jika melebihi 100 maka akan mengeluarkan output 'loading complete'.

2. Buatlah sebuah program yang dapat membaca apakah inputan user merupakan bilangan ganjil atau genap!

SourceCode :

```
6 package com.prak_pbo.pertemuan1;
7
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author m0n
13  */
14 public class ganjilgenap {
15
16     public static void main(String[] args) {
17         Scanner input = new Scanner(System.in);
18         System.out.println("==== Program Ganjil Genap ===");
19
20         System.out.println("Masukan bilangan bulat:");
21         int bil = input.nextInt();
22
23         if (bil % 2 == 1) {
24             System.out.println(bil + " adalah bilangan ganjil");
25         } else {
26             System.out.println(bil + " adalah bilangan genap");
27         }
28     }
29 }
30 }
```

Output :

```
Output - Run (ganjilgenap) x
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Loading ---
==== Program Ganjil Genap ====
Masukan bilangan bulat:
3
3 adalah bilangan ganjil
-----
BUILD SUCCESS
-----
Total time: 4.599 s
Finished at: 2021-04-13T12:04:34+07:00
-----
```

Penjelasan :

Program menggunakan perjabangan if else dimana kondisi pertama jika nilai input dioperasikan modulus 2 hasilnya 1 maka bilangan tersebut ganjil dan jika hasilnya tidak 1 maka bilangan tersebut adalah genap.

3. Buatlah sebuah program login sederhana. Jika user memasukan username dan password salah, ulangi hingga user menginput username dan password yang benar atau hingga user salah tiga kali.

SourceCode :

```
10  /**
11  *
12  * @author m0n
13  */
14  public class login {
15      public static void main(String[] args) {
16          Scanner input = new Scanner(System.in);
17
18          String username = "admin";
19          String password = "admin";
20
21          System.out.println("==== Silakan Login ====");
22          for (int i = 1; i <= 3; i++) {
23              System.out.println("");
24              System.out.println("Username :");
25              String user = input.nextLine();
26              System.out.println("Password :");
27              String passwd = input.nextLine();
28
29              if (user.equals(username) && passwd.equals(password)) {
30                  System.out.println("Login successfull!");
31                  break;
32              }
33              else {
34                  if (i == 3){
35                      System.out.println("Login failed! (" + i + "/" + 3 + ")");
36                      System.out.println("You have been blocked!");
37                  }
38                  else {
39                      System.out.println("Login failed! (" + i + "/" + 3 + ")");
40                  }
41              }
42          }
43      }
44  }
45  }
```

Output :

Jika username dan password tidak sesuai

```
Output - Run (login)
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Loading ---
==== Silakan Login ====

Username :
admin
Password :
a
Login failed! (1/3)

Username :
admin
Password :
123
Login failed! (2/3)

Username :
admin
Password :
12345678
Login failed! (3/3)
You have been blocked!
-----
BUILD SUCCESS
-----
Total time: 26.559 s
```

Jika username dan password sesuai

```
Output - Run (login)
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Loading ---
==== Silakan Login ====

Username :
admin
Password :
admin
Login successful!
-----
BUILD SUCCESS
-----
Total time: 6.617 s
Finished at: 2021-04-13T12:30:16+07:00
-----
```

Penjelasan :

Program tersebut menggunakan for untuk melooping 3x jika data yang dimasukan tidak sesuai dan jika sudah 3x salah maka program akan berhenti. Jika data yang dimasukan sesuai maka akan sukses/berhasil login

V. KESIMPULAN

Java Merupakan salah satu bahasa pemrograman yang berorientasi objek. OOP Atau Object Oriented Programing Adalah merupakan model pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam model pemrograman ini dibungkus dalam objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya. Model pemrograman ini merupakan model pemrograman yang mudah untuk dikembangkan dan di maintenance.