



Desain dan
Pemrograman
Berorientasi Objek

Pendahuluan Kelas dan Relasinya

Rosa A. S.

Rosa Ariani Sukamto

- Blog: <http://hariiniadalahhadiah.wordpress.com>
- Facebook: <https://www.facebook.com/rosa.ariani.sukamto>
- Email: rosa.ariani@upi.edu
- Website: <https://rosa-as.id>
- Youtube: <https://www.youtube.com/c/RosaArianiSukamto>

Terstruktur vs Objek

dalam satu file

Struktur NilaiMatKul

Struktur Elemen

Struktur List

Program Utama

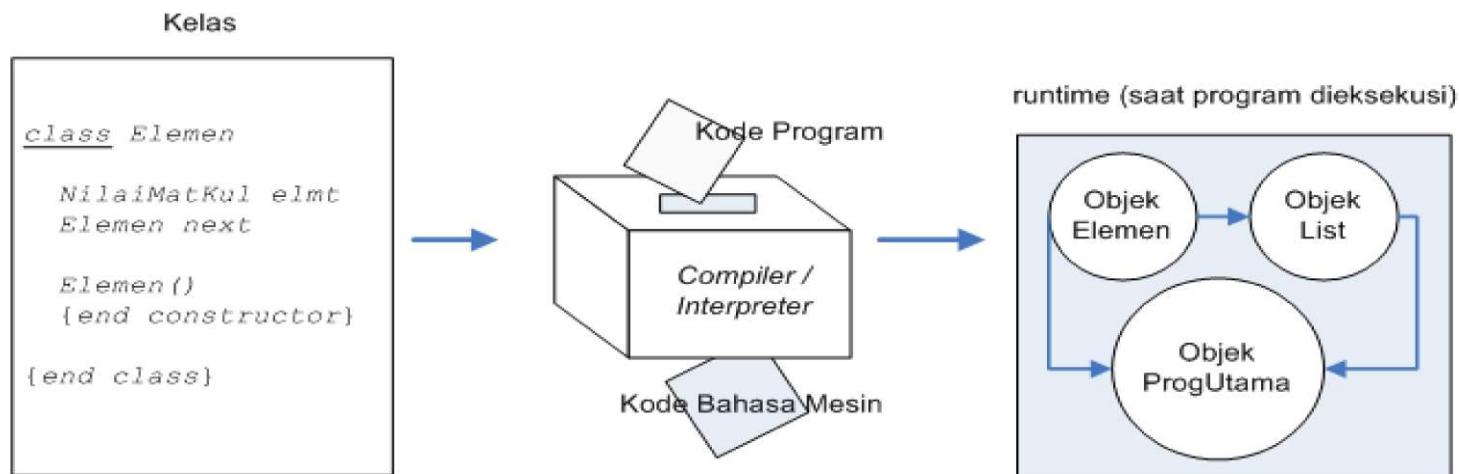
dalam satu direktori



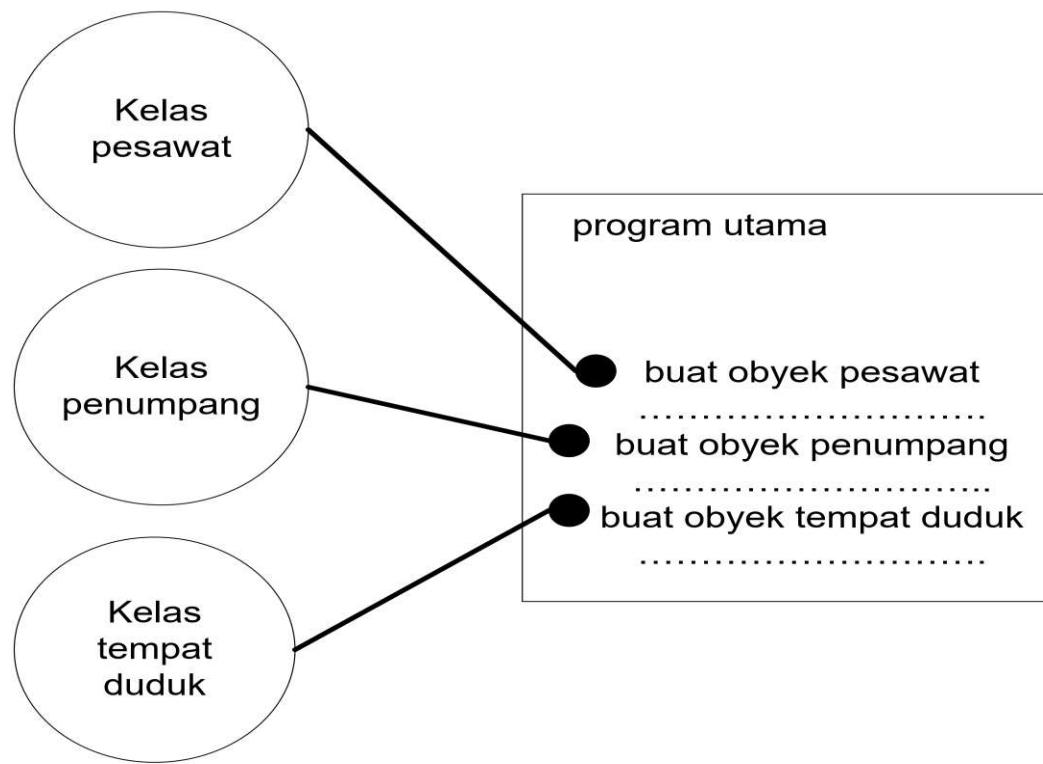
Terstruktur vs Objek



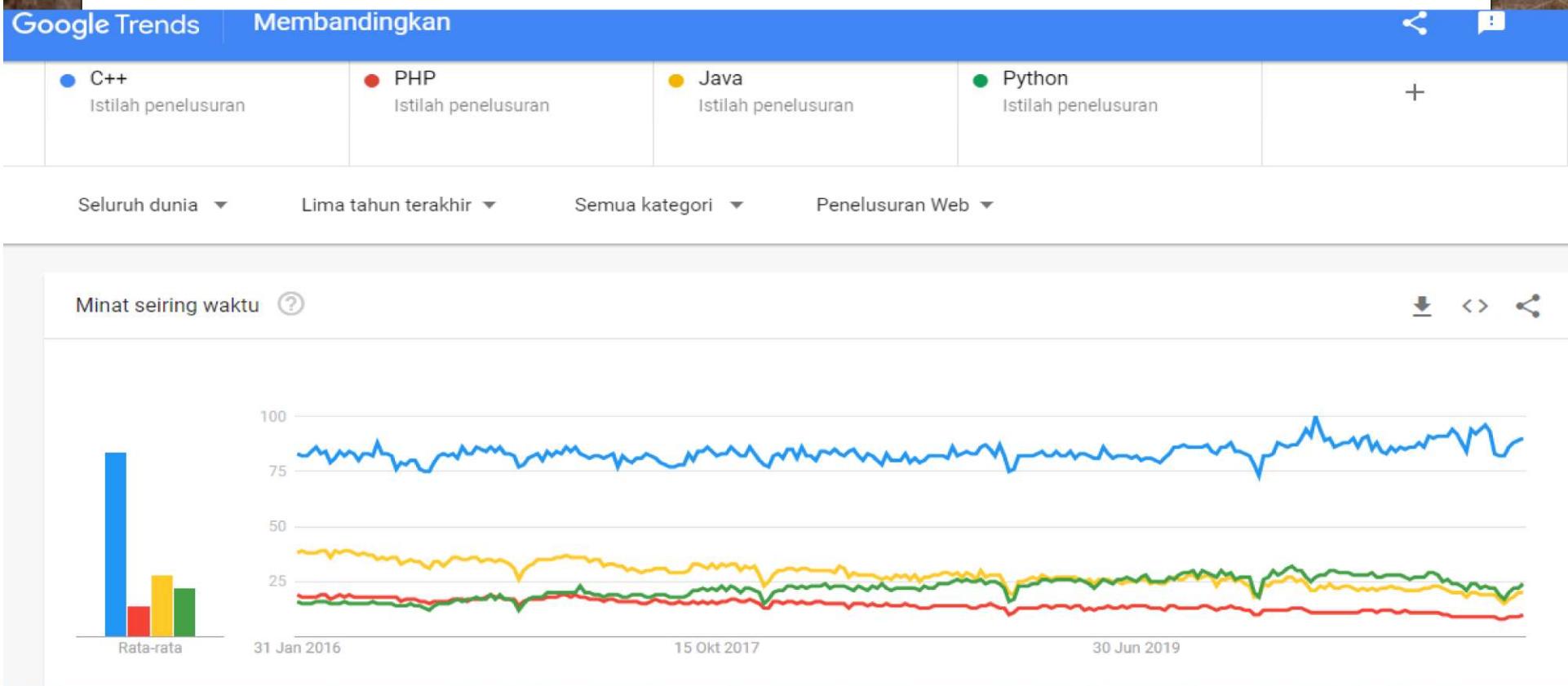
Pemrograman Berorientasi Objek (1)



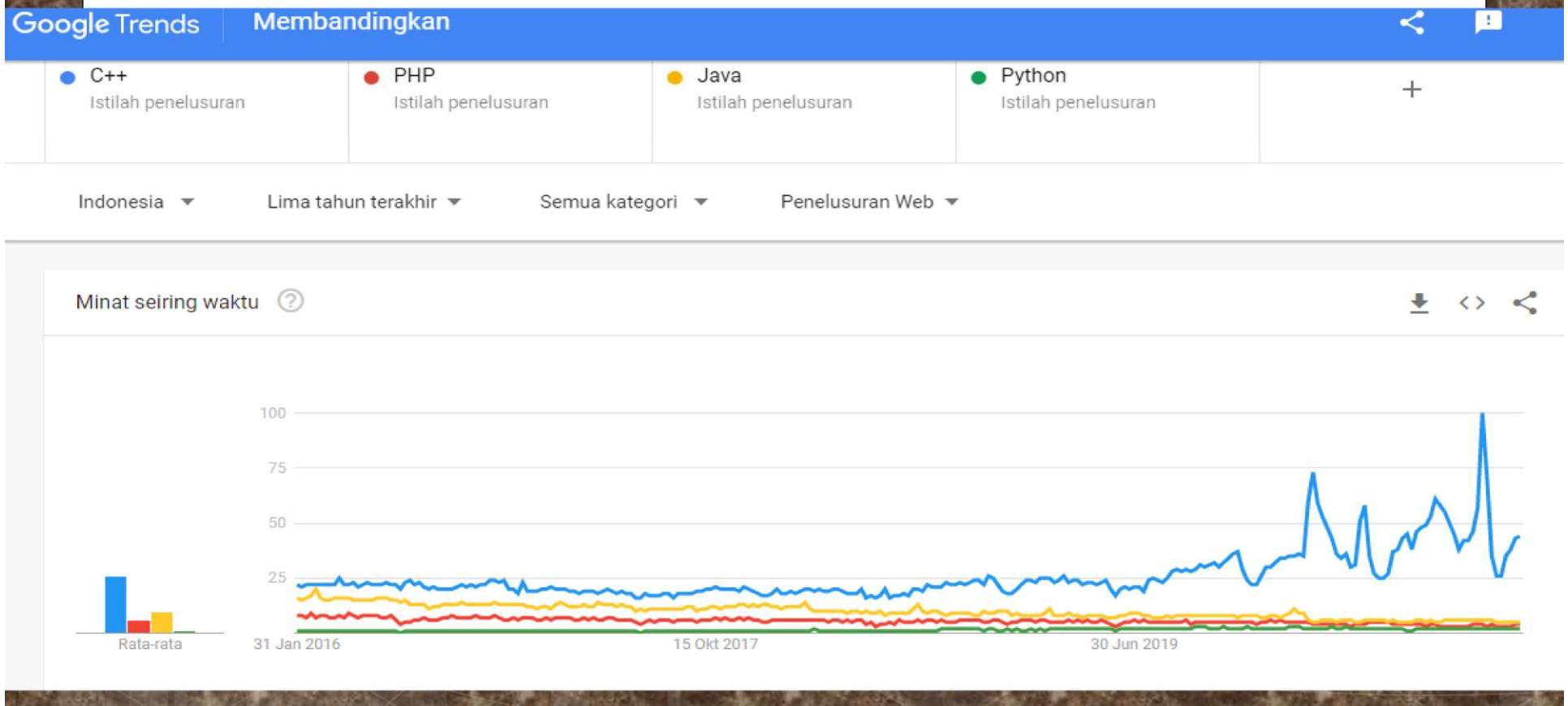
Pemrograman Berorientasi Objek (2)



Trend – Seluruh Dunia



Trend - Indonesia



The screenshot shows a Java-based IDE interface with the following components:

- Top Bar:** File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, ?
- Toolbar:** Includes icons for file operations like Open, Save, Print, and various toolbars.
- File List:** common.jury.php, judgedaemon.main.php, submission.php, Halo.cpp, main.cpp
- Function List:** Shows the structure of the Halo.cpp file, including the class definition and member functions.
- Code Editor:** Displays the following C++ code:

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 class Halo{
7     private:
8         string kata;
9
10    public:
11        Halo(){
12            kata = "Halo Dunia";
13        }
14        void tulis(){
15            cout << kata << endl;
16        }
17        ~Halo(){
18    };
19
20
```
- Search Results:** Find result - (1 hit)
Search "similarity detail" (1 hit in 1 file of 5692 searched)
C++ sou length : 264 lines : 20 Ln: 20 Col: 1 Pos: 265 Windows (CR LF) UTF-8 INS
D:\
Halo Dunia
D:\>
- System Tray:** Shows icons for various applications like Task Manager, File Explorer, and others, along with the date and time (11:42, 31/01/2021).

C++

A screenshot of a C++ code editor interface. The menu bar includes File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, and Window. The toolbar contains various icons for file operations like Open, Save, Print, and Undo/Redo. The tab bar shows several files: common.jury.php, judgedaemon.main.php, submission.php, Halo.cpp, and main.cpp. The main editor area displays the following C++ code:

```
1 #include "Halo.cpp"
2
3 int main() {
4     Halo objekHalo;
5     objekHalo.tulis();
6
7     return 0;
8 }
9
```

A red bracket is placed around the entire `main()` function block, starting from the opening brace at line 3 and ending at the closing brace at line 8.

C++

- *File* kelas dan main.cpp berada di dalam satu direktori
- Dalam satu direktori, hanya boleh ada 1 main
- Kompilasi pada MinGW
- g++ -c *.cpp
- Eksekusi pada MinGW
- g++ main.cpp -o halo.exe

```
1 <?php
2 class Halo{
3     private $kata = "";
4
5     function __construct(){
6         $this->kata = "Halo Dunia";
7     }
8
9     function tulis(){
10        echo $this->kata."<br/>";
11    }
12
13    function __destruct(){
14    }
15 }
16 ?>
```

The screenshot shows the 'Function List' window with the title bar 'Function List'. The main area displays the class structure of 'Halo.php'. At the top level, there is a file icon followed by 'Halo.php'. Below it, there is a class icon followed by 'Halo'. Under the 'Halo' class, three methods are listed: '_construct', 'tulis', and '_destruct'. Each method name is preceded by a purple diamond icon representing a magic method.

Find result - (1 hit)

File Edit View Bookmarks Tools Window Help

127.0.0.1/dpbo/

 127.0.0.1/dpbo/

Search Google

Halo Dunia



PHP

The screenshot shows a software interface with a title bar "PHP" and a menu bar including File, Edit, Search, View, Encoding, Language, Settings, Tools, Macro, Run, Plugins, Window, and ?. Below the menu is a toolbar with various icons. The main workspace contains several tabs: "judgedaemon.main.php", "submission.php", "Halo.cpp", "main.cpp", "Halo.php", and "index.php". The "index.php" tab is currently selected. The code editor displays the following PHP script:

```
1 <?php
2 include "Halo.php";
3
4 $halo = new Halo();
5 $halo->tulis();
6
7 ?>
```

To the right of the code editor is a "Function List" panel containing a single entry: "index.php".

Java

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

judgedaemon.main.php submission.php Halo.cpp main.cpp Halo.php index.php Ha Function List

```
1 class Halo{
2
3     public static void main(String[] args){
4
5         String kata;
6
7         kata = "Halo Dunia";
8         System.out.println(kata);
9
10    }
11
12 }
```

Main.java
Halo
main

Find result - (1 hit)

Command Prompt

```
D:\ >javac Halo.java
D:\ >java Halo
Halo Dunia
D:\ >
```

12:19
31/01/2021

- Path pada JDK
- Path="C:\Program Files\Java\jdk\bin";%path%
- Kompilasi
- javac *.java
- Eksekusi
- java Halo

Python

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

submission.php Halo.cpp main.cpp Halo.php index.php Halo.java Halo.py

```
1 class Halo:  
2     kata = 'Halo Dunia' #komentar  
3     print(kata)  
4  
5  
6  
7  
8  
9
```

Function List

..... Halo.py

Command Prompt

```
D:\>python Halo.py  
Halo Dunia
```

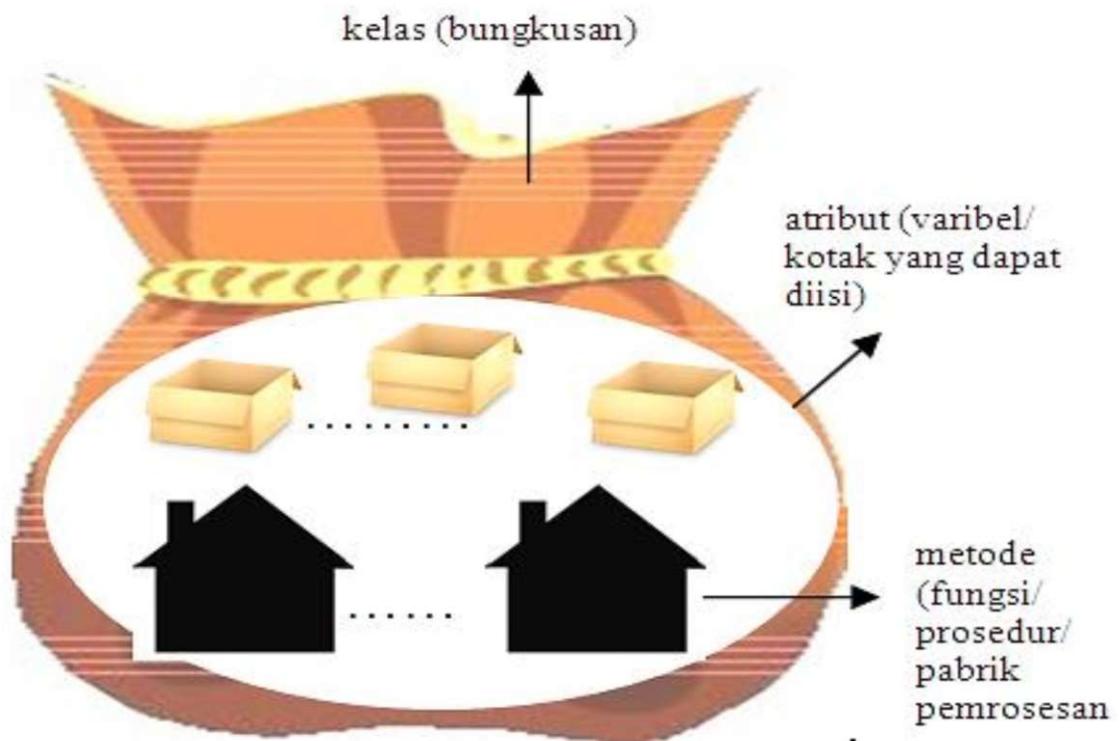
```
D:\>
```

12:25
31/01/2021

Python

- Path pada Kompilator Python dari Python.Org
- `python *.py`

Enkapsulasi



Atribut

```
#include <iostream>
#include <string>

using namespace std;

class Halo{
    private:
        string kata;
    public:
        -----
        ~Halo () {
    }
};
```

```
<?php
class Halo{

    private $kata = "";

    function __construct() {
        $this->kata = "Halo Dunia";
    }

    function tulis() {
        echo $this->kata."<br/>";
    }

    function __destruct() {
    }
?>
```

Akses atribut private (1)

- Atribut tidak dapat diakses langsung oleh kelas lain yang menggunakan kelas pemilik atribut

Atribut private (2) - C++

```
#include <iostream>
#include <string>

class Buku{
    private:
        string judul;
        string pengarang;

    public:
        Buku() {
        }

        Buku(string j, string p) {
            judul = j;
            pengarang = p;
        }

        void setJudul(string j) {
            judul = j;
        }

        void setPengarang (string p) {
            pengarang = p;
        }

        string getJudul () {
            return judul;
        }

        string getPengarang () {
            return pengarang;
        }

        ~Buku () {
        }
};
```

Atribut private (3) - C++

```
#include "Buku.cpp"
using namespace std;
int main() {
    Buku b1;
    b1.setJudul("J2ME");
    b1.setPengarang("Orang_1");

    cout << b1.getJudul() << endl;
    cout << b1.getPengarang() << endl;

    Buku b2 ("J2ME", "Orang_2");
    cout << b2.getJudul() << endl;
    cout << b2.getPengarang() << endl;

    return 0;
}
```

Atribut private (4) - PHP

```
<?php  
  
class Buku{  
  
    private $judul = "";  
    private $pengarang = "";  
  
    function __construct($j="",  
                        $p="") {  
        $this->judul = $j;  
        $this->pengarang = $p;  
    }  
  
    function setJudul($j="") {  
        $this->judul = $j;  
    }  
}
```

```
function setPengarang($p="") {  
    $this->pengarang = $p;  
}  
  
function getJudul () {  
    return $this->judul;  
}  
  
function getPengarang () {  
    return $this->pengarang;  
}  
?  
?
```

Atribut private (5) - PHP

```
<?php
    include "Buku.php";

    $b2 = new Buku("J2EE", "Orang_1");
    echo $b2->getJudul() . "<br/>";
    echo $b2->getPengarang() . "<br/>";

?>
```

Atribut private (6) - Java

```
class Buku{  
    private String judul;  
    private String pengarang;  
  
    Buku() {  
    }  
  
    Buku(String j, String p){  
        this.judul = j;  
        this.pengarang = p;  
    }  
  
    void setJudul(String j){  
        this.judul = j;  
    }  
  
    void setPengarang(String p){  
        this.pengarang = p;  
    }  
  
    String getJudul(){  
        return this.judul;  
    }  
  
    String getPengarang(){  
        return this.pengarang;  
    }  
}
```

Atribut private (7) - Java

```
class Main{  
    public static void main(String[] args){  
        Buku b1;  
        Buku b2;  
  
        b1 = new Buku();  
        b1.setJudul("J2ME");  
        b1.setPengarang("Orang_1");  
        System.out.println(b1.getJudul());  
        System.out.println(b1.getPengarang());  
  
        b2 = new Buku("J2EE", "Orang_2");  
        System.out.println(b2.getJudul());  
        System.out.println(b2.getPengarang());  
    }  
}
```

Akses atribut public

- Atribut dapat diakses langsung oleh kelas lain yang menggunakan kelas pemilik atribut

METODE

- Metode atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada pemrograman prosedural
- Pada sebuah metode di dalam sebuah kelas juga memiliki ijin akses seperti halnya atribut pada kelas, ijin akses itu antara lain private, public, dan protected yang memiliki arti sama pada ijin akses atribut yang telah dibahas sebelumnya.

konstruktor

- Sebuah kelas harus memiliki sebuah metode yang disebut sebagai konstruktor.
- Nama sebuah konstruktor harus sama dengan nama dari sebuah kelas, misalkan kelas Buku maka konstruktornya adalah Buku().
- Sebuah konstruktor juga dapat menerima sebuah masukan seperti halnya prosedur pada pemrograman prosedural. Fungsi dari sebuah konstruktor adalah :
 - mengalokasikan sebuah objek saat program dieksekusi (memerintahkan dibuatnya alokasi objek di memori saat program dijalankan)
 - memberikan nilai awal sebagai inisialisasi dari semua atribut yang perlu diinisialisasi
 - mengerjakan proses-proses yang diperlukan saat sebuah objek dibuat
- namun pada kenyataannya sebuah konstruktor dapat tidak berisi apa-apa, hal ini jika memang tidak diperlukan adanya inisialisasi atau proses yang dikerjakan ketika sebuah objek dibuat.
- Konstruktor harus bersifat public karena sebuah konstruktor akan diakses oleh kelas lain untuk membuat objek suatu kelas.

destruktor

- Destruktor adalah metode yang dipanggil secara otomatis ketika objek dihancurkan.
- Sebuah destruktor tidak harus ada pada kode program sebuah kelas jika *compiler* atau *interpreter* tidak memiliki *garbage collection* (mekanisme membersihkan alokasi objek-objek yang sudah tidak terpakai dari memori).

kelas (1) - C++

```
class Titik{
    /*kelas yang digunakan untuk
     mengimplementasikan sebuah tipe
     titik*/

    private:
        int x; /*koordinat x*/
        int y; /*koordinat y*/

    public:
        Titik(){
            /*konstruktor*/
            x = 0;
            y = 0;
        }

        Titik(int xp, int yp){
            /*konstruktor*/
            x = xp;
            y = yp;
        }

        void setX(int xp){
            /*mengeset nilai koordinat x*/
            x = xp;
        }
}
```

```
int getX(){
    /*mengembalikan nilai
     koordinat x*/
    return x;
}

void setY(int yp){
    /*mengeset nilai
     koordinat y*/
    y = yp;
}

int getY(){
    /*mengembalikan nilai
     koordinat y*/
    return y;
}

~Titik(){
    /*destruktur*/
}
};
```

kelas (2) - C++

```
#include <iostream>
#include <Titik.cpp>

using namespace std;
/*fungsi main untuk mengetes kelas Titik*/

int main() {

    Titik t1;
    Titik t2(11, 9);

    t1.setX(18);
    t1.setY(28);

    cout << "t1 : nilai X :" << t1.getX() << endl;
    cout << "t1 : nilai Y :" << t1.getY() << endl;

    cout << "t2 : nilai X :" << t2.getX() << endl;
    cout << "t2 : nilai Y :" << t2.getY() << endl;

    return 0;
}
```

kelas (3) - PHP

KELAS (3) - PHP

```
<?php
class Titik{
    /*kelas yang digunakan untuk
     mengimplementasikan sebuah
     tipe titik*/
    private $x; /*koordinat x*/
    private $y; /*koordinat y*/
    public function __construct(){
        /*konstruktor*/
        $this->x = 0;
        $this->y = 0;
    }
    public function setX($xp){
        /*mengeset nilai koordinat
         x*/
        $this->x = $xp;
    }
}
```

```
public function getX(){
    /*mengembalikan nilai
     koordinat x*/
    return $this->x;
}
public function setY($yp){
    /*mengeset nilai
     koordinat y*/
    $this->y = $yp;
}
public function getY(){
    /*mengembalikan nilai
     koordinat y*/
    return $this->y;
}
function __destruct(){
    /*destruktur*/
}
?>
```

kelas (4) - PHP

```
<?php
    include "Titik.php";

/*main untuk mengetes kelas Titik*/
$t1 = new Titik();

$t1->setX(18);
$t1->setY(28);

echo "t1 : nilai X : "
     . $t1->getX() . "<br/>";
echo "t1 : nilai Y : "
     . $t1->getY() . "<br/>";

?>
```

kelas (5) - Java

```
class Titik{  
    /*kelas yang digunakan untuk  
     mengimplementasikan sebuah  
     tipe titik*/  
  
    private int x; /*koordinat x*/  
    private int y; /*koordinat  
     y*/  
  
    Titik(){  
        /*konstruktor*/  
        x = 0;  
        y = 0;  
    }  
  
    Titik(int xp, int yp){  
        /*konstruktor*/  
        x = xp;  
        y = yp;  
    }  
  
    public void setX(int xp){  
        /*mengeset nilai koordinat  
         x*/  
        x = xp;  
    }
```

```
    public int getX(){  
        /*mengembalikan nilai  
         koordinat x*/  
        return x;  
    }  
  
    public void setY(int yp){  
  
        /*mengeset nilai koordinat  
         y*/  
        y = yp;  
    }  
  
    public int getY(){  
        /*mengembalikan nilai  
         koordinat y*/  
        return y;  
    }  
}
```

kelas (6) - Java

```
class Main{
/*metode main untuk mengetes kelas Titik*/

    public static void main(String[] args) {
        Titik t1 = new Titik();
        Titik t2 = new Titik(11, 9);

        t1.setX(18);
        t1.setY(28);

        System.out.println("t1 : nilai X : " + t1.getX());
        System.out.println("t1 : nilai Y : " + t1.getY());

        System.out.println("t2 : nilai X : " + t2.getX());
        System.out.println("t2 : nilai Y : " + t2.getY());
    }
}
```

kelas (6) - PYTHON

```
class Titik:  
#kelas yang digunakan untuk  
mengimplementasikan sebuah tipe titik  
  
    def __init__(self, x, y):  
        #konstruktor  
        self.x = x  
        self.y = y  
  
    def setX(self, x):  
        #mengeset nilai koordinat x  
        self.x = x  
  
    def getX(self):  
        #mengembalikan nilai koordinat x  
        return self.x  
  
    def setY(self, y):  
        #mengeset nilai koordinat x  
        self.y = y  
  
    def getY(self):  
        #mengembalikan nilai koordinat x  
        return self.y
```

kelas (7) - PYTHON

```
from Titik import Titik

t1 = Titik(18, 28)

print("t1 : nilai X : " + str(t1.getX()))
print("t1 : nilai Y : " + str(t1.getY()))
```

Inner class - Java (1)

```
1 import java.util.Scanner;
2
3 class Calculator{
4
5     private Operasi opr;
6
7     Calculator(){
8         opr = new Operasi();
9     }
10
11    Operasi getopr(){
12        return opr;
13    }
14
15    class Operasi{
16
17        Operasi(){
18
19        }
20
21        int tambah(int x, int y){
22
23            return (x + y);
24        }
25
26        int kurang(int x, int y){
27
28            return (x - y);
29        }
30
31        int kali(int x, int y){
32
33            return (x * y);
34        }
35
36        int bagi(int x, int y){
37
38            if(y > 0){
39                return (x / y);
40            } else{
41                return -999999;
42            }
43        }
44
45    }
46
47}
48
```

Inner class - Java (2)

```
class Main{  
  
    public static void main(String[] args){  
  
        int menu = 0;  
        int x = 0;  
        int y = 0;  
  
        Calculator cal = new Calculator();  
  
        System.out.println("masukkan menu");
```

```
Scanner sc = new Scanner(System.in);  
try{  
  
    menu = sc.nextInt();  
  
}catch(Exception e){  
}  
System.out.println("masukkan x");  
try{  
  
    x = sc.nextInt();  
  
}catch(Exception e){  
}  
System.out.println("masukkan y");  
try{  
  
    y = sc.nextInt();  
  
}catch(Exception e){  
}
```

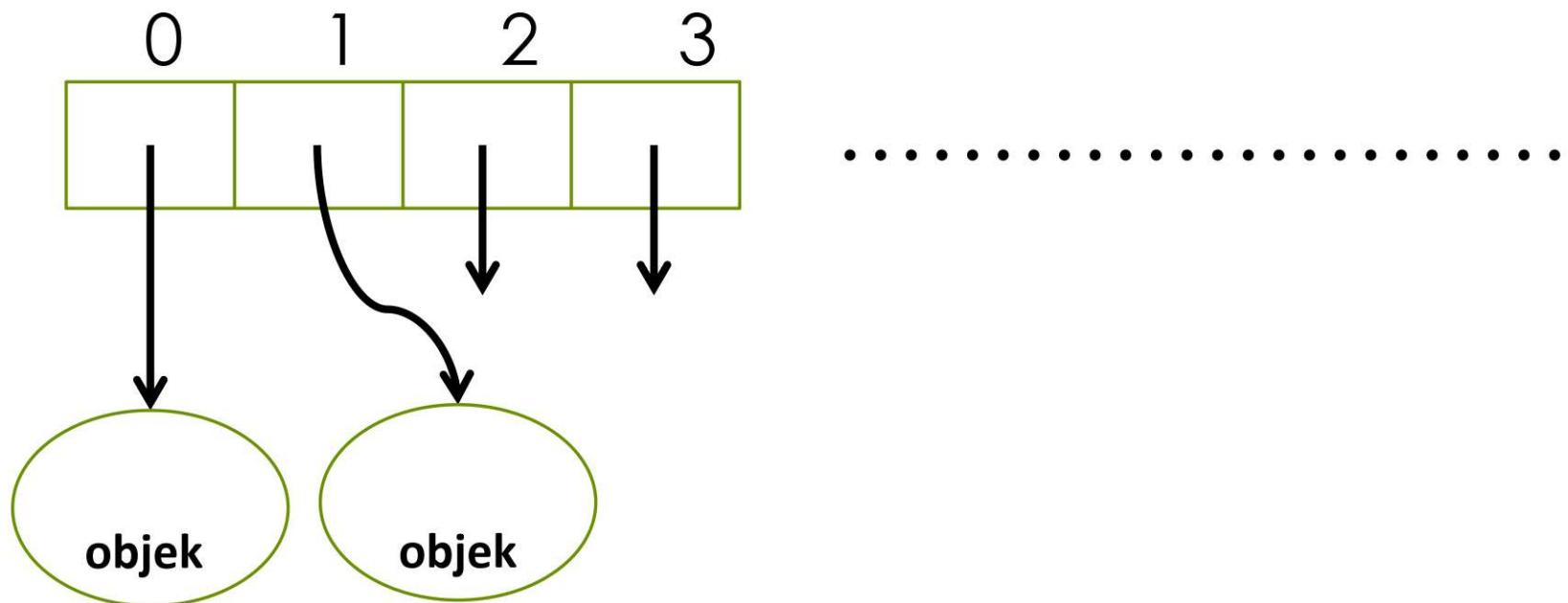
Inner class - Java (3)

```
switch(menu) {  
    ...  
    case 1 :  
        System.out.println(cal.getOpr().tambah(x,y));  
        break;  
    case 2 :  
        System.out.println(cal.getOpr().kurang(x,y));  
        break;  
    case 3 :  
        System.out.println(cal.getOpr().kali(x,y));  
        break;  
    case 4 :  
        System.out.println(cal.getOpr().bagi(x,y));  
        break;  
    }  
}
```

Array di Java - 1 dimensi

```
1 import java.util.Scanner;
2
3 public class CobaArrayString{
4     public static void main(String args[]){
5         String arr[] = new String[5];
6         arr[0] = "satu";
7
8         System.out.println("isi array ke-0 " + arr[0]);
9
10        int i = 0; // di java versi baru harus ada inisialisasi
11        int e = 0;
12        Scanner scan = new Scanner(System.in);
13
14        for(i=0;i<5;i++){
15            arr[i] = scan.next();
16            System.out.println("Isi array ke: " + i + " adalah " + arr[i]);
17        }
18    }
19
20 }
```

Ilustrasi Array Kelas



Array kelas di Java - 1 dimensi

```
1 import java.awt.Point;
2 import java.util.Scanner;
3
4 public class ArrayPoint{
5     public static void main(String args[]){
6         Point arrPoint[] = new Point[2];
7
8         int i = 0;
9         int x = 0;
10        int y = 0;
11
12        Scanner scan = new Scanner(System.in);
13
14        for(i=0;i<arrPoint.length;i++){
15            x = scan.nextInt();
16            y = scan.nextInt();
17            arrPoint[i] = new Point(x,y);
18            System.out.println("x: " + arrPoint[i].x + " y: " + arrPoint[i].y);
19        }
20    }
21 }
22 }
```

Array di Java - 2 dimensi

```
1 import java.util.Scanner;
2
3 public class Array2Dimensi{
4     public static void main(String args[]){
5         int arr[][]= new int[2][3];
6         int i = 0, j = 0;
7
8         Scanner scan = new Scanner(System.in);
9
10        for(i=0;i<arr.length;i++){
11            for(j=0;j<arr[i].length;j++){
12                arr[i][j] = scan.nextInt();
13                System.out.println(arr[i][j]);
14            }
15        }
16    }
17 }
18 }
```

Bagaimana membuat program dengan relasi kelas

- Menampilkan tabel untuk beberapa ukuran balok
 - Kelas Balok
 - Kelas Tabel
 - Kelas Main

```
Masukkan banyak balok: 2
Masukkan panjang, lebar, tinggi balok ke 1 :
1
2
3
Masukkan panjang, lebar, tinggi balok ke 2 :
2
3
4
-----
| 1 | 1 | 2 | 1 | 3 | 1 | 6 | 1 | 22 | 1 |
-----
| 2 | 1 | 3 | 1 | 4 | 1 | 24 | 1 | 52 | 1 |
```

Kelas Balok

```
public class Balok{  
    private int panjang;  
    //panjang balok  
    private int lebar;  
    //lebar balok  
    private int tinggi;  
    //tinggi balok  
  
    Balok(){  
        //konstruktor kosong  
    }  
  
    Balok(int panjang, int lebar, int  
          tinggi){  
        //konstruktor langsung isi  
        //atribut  
        this.panjang = panjang;  
        this.lebar = lebar;  
        this.tinggi = tinggi;  
    }  
}
```

```
//get set  
.....  
  
public int volume(){  
    //menghitung volume balok  
    return (panjang * lebar *  
tinggi);  
}  
  
public int luas(){  
    //menghitung luas balok  
    return ((2 * panjang *  
lebar) + (2 * panjang * tinggi) +  
(2 * lebar * tinggi));  
}
```

Kelas Tabel (1)

```
public class Tabel{  
    private int baris;  
    //banyaknya baris  
    private int kolom;  
    //banyaknya kolom  
  
    Tabel(){  
        //konstruktor kosong  
    }  
  
    Tabel(int baris, int kolom){  
        //konstruktor langsung mengisi  
        //atribut  
        this.baris = baris;  
        this.kolom = kolom;  
    }  
    public void setBaris(int baris){  
        //mengeset baris  
        this.baris = baris;  
    }
```

```
//get set  
.....  
public void buatBaris(String[] isi,  
                      int add){  
    int i = 0, j = 0; //inisialisasi  
    variabel di java  
    //buat tabel bagian atas  
    for(i=0;i<isi.length;i++){  
        for(j=0;j<(isi[i].length() +  
add) ;j++) {  
            System.out.print("-");  
        }  
    }  
    System.out.println("");
```

Kelas Tabel (2)

```
//buat baris isi
for(i=0;i<isi.length;i++) {
    System.out.print("| ");
    System.out.print(isi[i]);
    for(j=0;j<(add-3);j++) {
        System.out.print(" ");
    }
    System.out.print("| ");
}
System.out.println("");
```

```
//buat tabel bagian bawah
for(i=0;i<isi.length;i++) {
    for(j=0;j<(isi[i].length() + add);j++) {
        System.out.print("-");
    }
    System.out.println("");
}
```

Kelas Main (1)

```
import java.util.Scanner;
public class Main{
    public static void main(String[]
    args){
        int n = 0; //banyaknya balok
        System.out.println("Masukkan
banyak balok:");
        Scanner sc = new
Scanner(System.in);

        try{
            n = sc.nextInt();
        }catch(Exception e){
        }
        //array balok
        Balok[] arrbalok = new
Balok[n];
    }
}
```

```
//mengisi array balok
int i = 0, p = 0, l = 0, t = 0;
for(i=0;i<n;i++){
    System.out.println("Masukkan
panjang, lebar, tinggi balok ke "
+ (i+1) + " :");
    try{
        p = sc.nextInt();
    }catch(Exception e){
    }
    .....
    arrbalok[i] = new Balok(p,
l, t);
}
}
```

Kelas Main (2)

```
//menampilkan array balok
Tabel tab = new Tabel(n, 5);
for(i=0;i<n;i++) {
    String[] arrstr = new String[5];
    arrstr[0] = "" + arrbalok[i].getPanjang();
    arrstr[1] = "" + arrbalok[i].getLebar();
    arrstr[2] = "" + arrbalok[i].getTinggi();
    arrstr[3] = "" + arrbalok[i].volume();
    arrstr[4] = "" + arrbalok[i].luas();
    tab.buatBaris(arrstr, 5);
}
}
```

Mari Mencoba

- Ubah Kode program ke bahasa C++ dan Python
- Buat program untuk membuat tabel nama nasabah bank
 - Nama
 - Saldo
 - Transaksi terakhir

Daftar Pustaka

