



— CLASS & OBJECT —

KELOMPOK 5

UNIVERSITAS DIPA MAKASSAR

Our Team

KELOMPOK 5



RADEN AHMAD



WIDYA APRILIANI RIDWAN



DAMAR SURYA ANATA



ELMA GRACELIA



BERTOLD MASPA



ABD HAMID JUMAIN



AHMAD FUAD YULHAIDIR



DELI PANGINGNGI



MIGHEL CHRISTIAN



ANDI AIDIL FATRA.R



01

Pengertian Object-Oriented
Programming (OOP)

02

Pengertian Class dan Object

03

Karakteristik Class dan Object

04

Bagian Class dan Macam Access
Modifier

05

Contoh Penggunaan Class dan Object



```
        error' => $quote['sort_order'],
    }
}

$sort_order = array();

foreach ($quotes as $key => $value) {
    $sort_order[$key] = $value['sort_order'];
}

array_multisort($sort_order, SORT_ASC, $quotes);

$this->session->data['lpa']['shipping_methods'] = $quotes;
$this->session->data['lpa']['address'] = $address;

if (empty($quotes)) {
    $json['error'] = $this->language->get('
        error_no_shipping_methods');
} else {
    $json['quotes'] = $quotes;
}

if (isset($this->session->data['lpa']['shipping_method']) && !
empty($this->session->data['lpa']['shipping_method']) &&
isset($this->session->data['lpa']['shipping_method']['code'])
) {
    $json['selected'] = $this->session->data['lpa']['
        shipping_method']['code'];
} else {
    $json['selected'] = '';
}

{
    $json['error'] = $this->language->get('error_shipping_methods');

}

$this->response->addHeader('Content-Type: application/json');
```

```
        || (this.paused = function (e) {
    if (this.$element.find('.next, .prev').length && $.support.transition.end)
        this.$element.trigger($.support.transition.end)
    this.cycle(true)
}

this.$element.on('slide.bs.carousel', {
    next: function (e) {
        var target = $next[0]
        var slideEvent = $.Event('slide.bs.carousel', {
            relatedTarget: relatedTarget,
            direction: direction
        })
        this.$element.trigger(slideEvent)
    },
    prev: function (e) {
        var target = $prev[0]
        var slideEvent = $.Event('slide.bs.carousel', {
            relatedTarget: relatedTarget,
            direction: direction
        })
        this.$element.trigger(slideEvent)
    }
})
```

OBJECT-ORIENTED PROGRAMMING

Pemrograman berorientasi objek atau Object-Oriented Programming (OOP) merupakan suatu pendekatan pemrograman yang menggunakan object dan class. OOP adalah paradigma atau Teknik pemrograman di mana semua hal dalam program dimodelkan seperti objek dalam dunia nyata. OOP merupakan cara pandang dalam menganalisa sistem dan permasalahan pemrograman. Dalam OOP, setiap bagian dari program adalah object.





PENGERTIAN **CLASS**

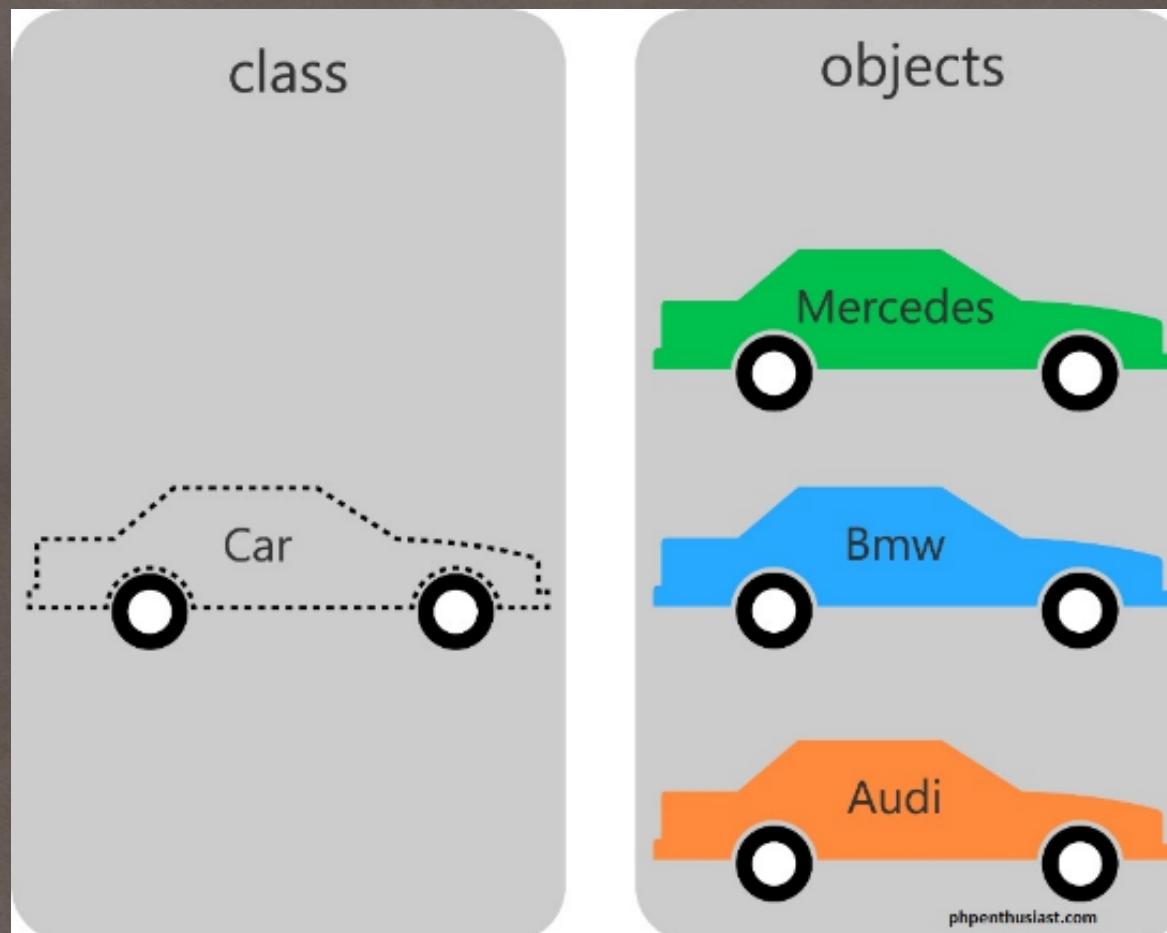
Class merupakan gambaran dari sebuah object atau bisa dikatakan output dari sebuah object. Pada bahasa pemrograman, class merupakan sekumpulan kode yang dituliskan untuk mendefinisikan property dan metod yang ada pada sebuah object. Class didefinisikan dengan menampung nilai property dan metod, dimana properti adalah sebuah data yang menjelaskan tentang class dan metode adalah tingkah laku yang bisa dilakukan oleh object.

OBJECT

Apa itu Object?

Object adalah hasil instansiasi dari class, dan mengandung seluruh resource yang telah didefinisikan pada class. Dikarenakan class merupakan cetakan dari object, maka object hasil instansiasi juga mempunyai resource seperti class.

KARAKTERISTIK CLASS DAN OBJECT



- Class merupakan wadah yang akan digunakan untuk menciptakan objek, maka jelaslah bahwa kita harus membuat class terlebih dahulu sebelum membuat objek.
- Class dan Object merupakan fitur yang sangat membantu untuk mendirikan sebuah program besar, menjadikan sebuah code program yang ditulis oleh programmer mudah untuk dimengerti, lebih terstruktur, dan juga mudah dalam pemeliharaan program.

Karakteristik Class

- a. Tidak menggunakan spasi atau menggunakan spasi dengan diganti “_”
(misal class Sarjana_Mipa)
- b. Tidak menggunakan kata-kata yang telah dipakai oleh compiler (reserved Word, lihat bab sebelumnya)



Kelompok 5

Karakteristik Object

Objek-objek ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu.



CLASS

01

VARIABEL / PROPERTI / ATRIBUT

karakteristik yang dimiliki oleh objek. Dibagian itu kita dapat mendeklarasikan atribut-atribut yang dibutuhkan kelas untuk membentuk objek. Hal ini biasa juga disebut Access Modifier

02

KONSTRUKTOR

method yang berfungsi untuk menginisialisasi variabel-variabel instans yang akan dimiliki oleh objek. Method konstruktor harus memiliki nama yang sama dengan nama kelas. Kegunaan konstruktor yaitu Mengalokasikan ruang bagi sebuah kelas dalam objek, Memberikan nilai awal terhadap anggota data suatu objek dan Membentuk tugas-tugas umum lainnya.

03

METHOD

Method merupakan tingkah laku yang dimiliki oleh objek. Method ini bisa berbentuk prosedur maupun fungsi.

MACAM-MACAM ACCESS MODIFIER (ATRIBUT)



Private

Memberikan hak akses hanya kepada anggota class tersebut untuk menggunakan dan/atau mengubah nilai dari property atau method tersebut.



Public

Memberikan hak akses kepada property dan method agar dapat digunakan diluar class tersebut.



Protected

Memberikan hak akses kepada anggota class nya dan anggota class hasil inheritance (penurunan sifat) dari class tersebut.

```

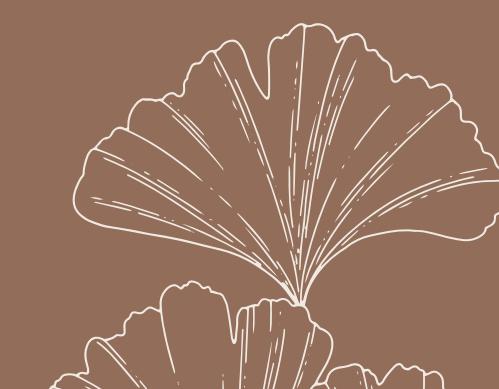
4 // Buat class Car dengan atribut
5 class Car {
6     public :
7         string brand;
8         string model;
9         string year;
10    };
11
12 int main()
13 {
14
15 // Buat objek Car
16 Car carObj1;
17 carObj1.brand = "Brand : BMW\n";
18
19 carObj1.model = "Model : X5\n";
20
21 carObj1.year = "tahun keluaran : 1999\n";
22
23 // Buat objek Car yang Lain
24 Car carObj2;
25 carObj2.brand = "Brand : Mercendes\n";
26
27 carObj2.model = "Model : Peugeot 407 SW\n";
28
29 carObj2.year = "tahun keluaran : 2004\n";
30
31 // Buat objek Car yang Lain
32 Car carObj3;
33 carObj3.brand = "Brand : AUDI\n";
34
35 carObj3.model = "Model : R8\n";
36
37 carObj3.year = "tahun keluaran : 2020\n";
38
39 // Cetak nilai atribut
40 cout << carObj1.brand << " - " << carObj1.model << " - " << carObj1.year << "\n";
41
42 cout << carObj2.brand << " - " << carObj2.model << " - " << carObj2.year << "\n";
43
44 cout << carObj3.brand << " - " << carObj3.model << " - " << carObj3.year << "\n";
45
46 return 0;

```

CONTOH PENGGUNAAN CLASS DAN OBJECT

1. Class Access Private

Public dalam artian yang lebih mudah dipahami adalah sebuah akses yang diperlukan dalam program class. Dan tanpa public data program yang ada dalam satu class itu tidak dapat dijalankan atau di running.



2. Class Access Private



Private dalam artian yang lebih mudah di pahami adalah sebuah akses yang ada dalam Class, yang jika dimunculkan atau diketik dalam program Class maka program



tersebut akan besifat tertutup (tidak dapat dijalankan). Private



merupakan atribut dari class yang membutuhkan Public agar dapat dijalankan pada int main.

```
1 #include <iostream>
2
3 using namespace std;
4
5 class J {
6     private:
7     public:
8     string Namakelas;
9     string Jumlahsiswa;
10    string Ketuakelas;
11
12    void Detail_Kelas(){
13        cout<<Namakelas<<endl;
14        cout<<Jumlahsiswa<<endl;
15        cout<<Ketuakelas<<endl;
16    }
17 }
18
19 int main()
20 {
21     //Object Kelas J
22     J kelasJ;
23     kelasJ.Namakelas = "Kelas J";
24     kelasJ.Jumlahsiswa = "\tJumlah Siswa = 34 orang";
25     kelasJ.Ketuakelas = "\tKetua Kelas = Raden Ahmad";
26     kelasJ.Detail_Kelas();
27
28     return 0;
29 }
```

3. CLASS ACCESS PROTECTED

Protected dalam artian yang lebih mudah dipahami yaitu dia memiliki konsep seperti Private, tetapi yang membedakannya dengan private yaitu protected bisa diakses ke class yang lain sedangkan private hanya bisa di akses dalam classnya sendiri.

```
1 #include <iostream>
2
3 using namespace std;
4 class kuliah{
5     protected:
6     string nama = "UNIVERSITAS DIPA MAKASSAR";
7 }
8 class Kampus:protected kuliah {
9     protected:
10    string nama = "UNIVERSITAS DIPA MAKASSAR";
11
12 // buat hak akses public
13 public:
14     string namaKuliah() {
15         return nama;
16     }
17 }
18 int main()
19 {
20     Kampus dipa ;
21     cout<<dipa.namaKuliah();
22 }
23
```

KESIMPULAN

CLASS DAN OBJECT

DALAM PROSES PERCOBAAN PEMBUATAN CLASS DAN
OBJECT DAPAT KITA SIMPULKAN :



Nama class dan nama object harus sama dan nama object tersebut dapat diubah dengan sesuai yang di inginkan.



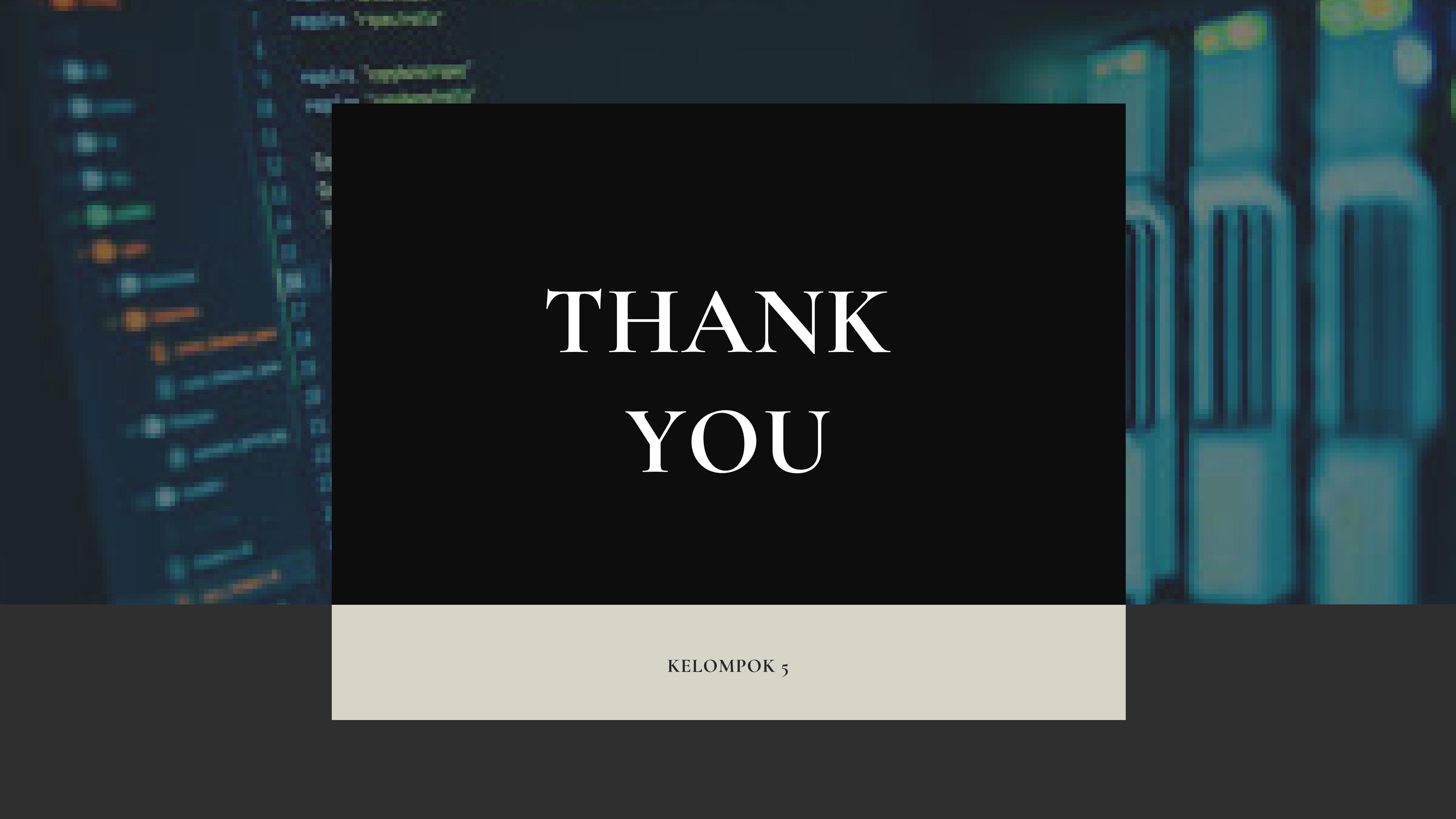
Dapat membuat dua object dengan nama yang berbeda yang berasal dari sebuah object yang sama..



Untuk mengakses atribut dari sebuah class object harus menggunakan nama object yang telah diubah, tanda titik(.) dan diikuti dengan atribut yang hendak di akses



penggunaan method harus digunakan dalam sebuah class



THANK
YOU

KELOMPOK 5