

1. Karena dengan OOP, kita dapat dengan mudah menambahkan fitur-fitur baru dengan cara menambahkan object object tanpa harus mengganggu fungsi lainnya.
2. Object Oriented Programming merupakan paradigma pemrograman yang mana memanfaatkan konsep objek , yang merupakan suatu entitas yang dapat berasosiasi dan berkolaborasi dengan object lainnya, atau pun program utama. Pada kasus PHP, object object yang sudah dibuat sedemikian rupa dapat dideklarasikan dengan tanda \$, dan dipanggil sebagai suatu elemen atau pun entitas lainnya yang dapat berinteraksi dengan aplikasi web.
3. Dengan OOP, para developer dapat dengan mudah melihat gambaran aplikasi secara menyeluruh dan menambahkan fitur-fitur baru dengan menambahkan object baru dibandingkan harus menyelipkannya ke dalam suatu baris kode dalam fungsi, sehingga meningkatkan skalabilitas dan efisiensi dari aplikasi.
4. hal tersebut dapat dilakukan dengan menarik isi dari suatu class ke class lainnya. Contoh yang dapat dilakukan sebagai berikut,

```
class Fruit {
    public $name;
    public $color;
    public function __construct($name, $color) {
        $this->name = $name;
        $this->color = $color;
    }
    public function intro() {
        echo "The fruit is {$this->name} and the color is {$this->color}.";
    }
}

// Strawberry is inherited from Fruit
class Strawberry extends Fruit {
    public function message() {
        echo "Am I a fruit or a berry? ";
    }
}
```

5. Class itu sendiri merupakan kumpulan atribut yang dimiliki oleh suatu tipe, berdasarkan paradigma OOP. Dalam hal ini, sifat-sifat umum biasanya menjadi nama dari class, seperti hewan, makanan, dan lainnya. Sedangkan object adalah suatu variable kompleks yang memiliki atribut dari suatu class, dan dapat memiliki atribut yang lebih spesifik dan unik dari class tersebut yang membedakan objek tersebut dari objek lainnya. Contohnya seperti objek anjing dan kucing, yang dibuat dari class hewan.
6. Class lebih spesifik untuk menyatakan atribut atribut dari suatu hal yang lebih umum, di mana object merupakan variable kompleks yang memiliki atribut atribut dari suatu class, namun dapat memiliki value atribut yang spesifik untuk membedakan dirinya dengan object lain dari class yang sama. Dalam hal PHP ini, object harus dideklarasikan dengan menyertakan class yang ada seperti ini:

```
class Fruit {
    // Properties
    public $name;
```

```

public $color;

// Methods
function set_name($name) {
    $this->name = $name;
}
function get_name() {
    return $this->name;
}
}
// Deklarasi Object apple dari class fruit
$apple = new Fruit();

```

7. Bagaimana cara mengimplementasikan encapsulation dalam PHP OOP, dan mengapa hal ini diperlukan untuk meningkatkan keamanan dan modularitas kode?
8. Jelaskan konsep polymorphism dalam PHP OOP dan berikan contoh penggunaannya dalam konteks pengembangan web.
9. Mengapa konsep abstraksi (abstraction) menjadi kunci dalam paradigma OOP, dan bagaimana hal ini dapat meningkatkan efisiensi dan keterbacaan kode PHP?
10. Apa peran interface dalam PHP OOP, dan bagaimana penggunaannya dapat memfasilitasi pengembangan aplikasi yang lebih modular dan dapat diuji (testable)?