|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| OOP PHP #23 | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. OOP (Object Oriented Programming) merukapan metode menulis program dengan berbasis object. Jadi hampir semua hal di dalam code dianggap sebagai object
2. Class merupakan blueprint dari sebuah object
3. Object merupakan instansi dari sebuah class
4. Jika di ibaratkan ke dunia nyata: Terdapat class hewan karnivora, maka objectnya adalah harimau, macan, singa, dll
5. Satu class dapat menghasilkan lebih dari satu object
6. Variable didalam class disebut property
7. Function di dalam class disebut method
8. Setiap property dan method harus di beri visibility (public, private, protected, dll)
9. Cara menginstansi class: $object = new Class\_name;
10. Nama class biasanya diawali dengan huruf kapital
11. Cara menjalankan method dalam sebuah object: $object->method();
12. Jika suatu property atau method memiliki visibility private maka property atau method tersebut tidak dapat diakses dari luar class
13. Nama file umumnya disamakan dengan nama class
14. Penulisan nama method umunya menggunakan camelCase
15. Kata kunci $this mengacu pada object itu sendiri
16. Setelah panah tidak boleh ada tanda $
17. Method \_\_construct() akan langsung dijalankan Ketika class di inisialisasikan atau dibuatkan object
18. Visibility static membuat suatu method atau property dapat dipanggil tanpa inisialisasi object terlebih dahulu
19. Cara memenaggil method atau property static: Class\_Name::method();

**Saya Belum Mengerti**

1.

2.

|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| Database Class #24 | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Kita dapat membuat method untuk mengambil semua data dalam table category
2. Kita juga dapat membuat membuat method untuk mengambil salah satu data dari table category
3. Kita dapat membuat method untuk menghitung jumlah data
4. Kita dapat membuat method untuk menjalankan syntax SQL
5. Kita dapat membuat method untuk memunculkan pesan

**Saya Belum Mengerti**

1.

2.