

# **LAPORAN PRAKTIKUM ALGORITMA DAN PEMOGRAMAN**

(Dosen pengampu : Sherly Gina Supratman, M.Kom.)



**Nama : Muhammad Rizky**  
**NIM : 20240810023**  
**Kelas : TINFC-2024-02**

**TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS KUNINGAN**  
**2024**

## A. Latar Belakang

Dalam Perkembangan Teknologi yang berkembang sangat cepat ini, algoritma dan pemrograman merupakan salah dua penopang perkembangan teknologi yang begitu cepat, dengan demikian jika kita ingin ikut berkontribusi dalam perkembangan teknologi, kita perlu memahami apa itu algoritma dan programan, dan untuk memahaminya kita bisa mulai dari mempelajari programan dasar yang di implementasikan melalui salah satu matakuliah di semester satu ini.

## B. Tujuan

Tujuan di laksanakan praktikum ini, sebagian sebagai penerapan dari materi object oriented programming (oop).

## C. Landasan Teori

Menurut sebuah artikel dari [rpl.upi.media](http://rpl.upi.media) OOP adalah paradigma pemrograman yang berfokus pada pemodelan program sebagai kumpulan objek yang memiliki atribut (data) dan metode (fungsi atau perilaku). Objek dalam OOP merepresentasikan entitas nyata atau konseptual dalam sistem yang sedang dikembangkan. Konsep OOP terdiri dari empat prinsip utama: Abstraksi, Pewarisan, Polimorfisme, dan Enkapsulasi.

## D. Alat, Bahan dan Perangkat



Alat, bahan, prangkat kersa, dan perangkat lunak yang digunakan:

1. Laptop
2. Visual Studio Code

## E. Prosuder Kerja

1. Kembangkanlah code program sesuai studi kasus posttest 1 di atas, kita dapat menambahkan konsep Abstraksi dengan membuat method `makeSound()` pada kelas `Animal` menjadi abstrak, yaitu dengan menambahkan tanda "=" setelah tipe data method dan nilai "O" di dalamnya. Dengan menandai method sebagai abstrak, kita memberitahu compiler bahwa kelas tersebut adalah abstrak dan tidak dapat diinisialisasi, sehingga hanya dapat digunakan pada kelas turunannya.

## F. Hasil

```
nineth >  tugas.cpp >  main()
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  class Animal {
7      public:
8          string name;
9          int age;
10
11         virtual void makeSound() = 0;
12 };
13
14 class Dog : public Animal {
15     public:
16         void makeSound() override {
17             cout << name << " barks!" << endl;
18         }
19 };
20
21 class Cat : public Animal {
22     public:
23         void makeSound() override {
24             cout << name << " meows!" << endl;
25         }
26 };
```

Kode diatas meruakan sebuah kode yang berisi objek animal dan juga turunan dari objek animal yakni dog dan cat

```

28  int main () {
29
30      Dog dog;
31      dog.name = "buddy";
32      dog.age = 5;
33
34      Cat cat;
35      cat.name = "mittens";
36      cat.age = 3;
37
38      Animal* animal1 = &dog;
39      animal1->makeSound();
40      Animal* animal2 = &cat;
41      animal2->makeSound();
42
43  }

```

Dan kode diatas merupakan implementasi dari objek yang telah kita buat.

## G. Kesimpulan

Tugas diatas kita dapat mengambil kesimpulan, bahawasanya dengan menggunakan objek kita dapat menuliskan sekali kode tersebut, tapi kita bebas dapat memamnggilnya kapan saja semau kita, tanpa harus menuliskan kode serupa, dan kita hanya perlu memanggil objeknya saja

## H. Daftar Pustaka

- <https://rpl.upi.edu/mendalami-konsep-oop-object-oriented-programming-dalam-pengembangan-perangkat-lunak/#:~:text=Apa%20itu%20OOP?,Pewarisan%2C%20Polimorfisme%2C%20dan%20Enkapsulasi.>