# LAPORAN PRAKTIKUM

# PEMROGRAMAN BERORIENTASI OBJECT

# SLIDE 02



# Sahrul Ramadhani

# 244107020058

# 2H

# D-IV Teknik Informatika

# Teknologi Informasi

# Politeknik Negeri Malang

# 2025

# Pertanyaan

## Apa tujuan utama penggunaan enkapsulasi?

## Berikan contoh kasus sebuah objek di dunia nyata dimana kita harus menerapkan konsep enkapsulasi antara lain:

o Menggunakan access modifier private pada atribut-atributnya

o Menggunakan method khusus untuk mengubah nilai atribut-atributnya

## Berikan contoh kasus sebuah objek di dunia nyata dimana kita harus membuat konstruktor pada sebuah class

# Jawaban

## Untuk menyembunyikan sebuah atribut di dalam class, yaa tujuannya agar program terstruktur, Selain Terstruktur kegunaan keamanan juga termasuk di dalamnya seperti agar methode hanya bisa di akses oleh class lain yang disediakan, implementasi di dunia nyata ya agar user tidak bisa menjalankan sebuah method yang hanya bisa di Read Only oleh User, namun juga bisa di edit melalui method publik lain yang emang di sediakan untuk mengedit Method yang di enkapsulasi tersebut, Misalnya Yang sudah di jelaskan tadi method yang di enkapsulasi adalah Mobil, Kecepatan, dan power mobil di enkapsulasi, namun user masih bisa memberi input perintah, power = on, Tambah Kecepatan, Kurangi Kecepatan.

## Access Modifier pada Mesin Cuci

* - power ; Boolean
* - washTimer ; INT
* - dryTimer ; INT
* - Buang Air ; Boolean
* - isiAir ; Boolean
* + nyalakanPower (): void
* + matikanPower (): void
* + tambahWashTimer ():void
* + turangiWashTimer ():void
* + tambahDryTimer (): void
* + turangiDryTimer (): void
* + mulaiBuangAir ():void
* + stopBuangAir ():void
* + mulaiIsiAir (): void
* + stopisiAir (): void

## Contoh Konstruktor Jadwal Kuliah

* + - kodeMk : String
  + - namaMk : String
  + - hari : String
  + - jamMulai : String
  + -jamSelesai : String
  + - ruang : String
* + JadwalKuliah(kodeAwal: String, namaAwal: String, hariAwal: String,

jamMulaiAwal: String, jamSelesaiAwal: String, ruangAwal: String)

* + setKodeMk(): void
* + getKodeMk(): String
* + setNamaMk(): void
* + getNamaMk(): String
* + setHari(): void
* + getHari(): String
* + setJamMulai(): void
* + getJamMulai(): String
* + setJamSelesai(): void
* + getJamSelesai(): String
* + setRuang(): void
* + getRuang(): String
* + getDurasiMenit(): int
* + tampilJadwal ():int