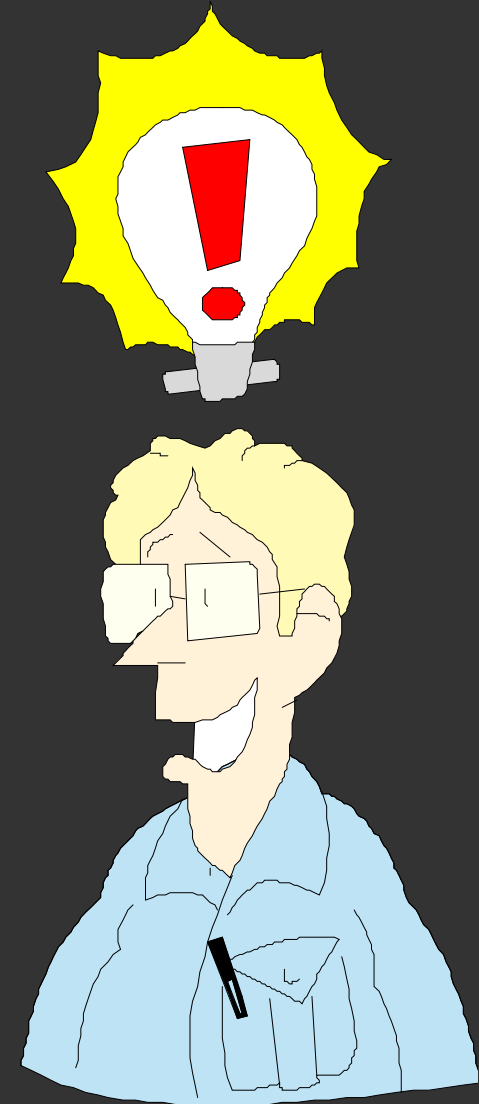


Object  
Oriented  
Programming

# Why OOP ?

- Kemampuan untuk memelihara dan mengimplementasikan perubahan di dalam program lebih efektif dan cepat
- Memungkinkan spesialis (programmer) bekerja secara parsial
- Memungkinkan dengan mudah penggunaan kembali coding program yang digunakan oleh program lain



# Object

“an object is a structure for incorporating data and the procedures for working with that data”

(Daniel R.Clark)

Terdiri dari 3 komponen dasar :

Properties, Method dan Event

## Properties

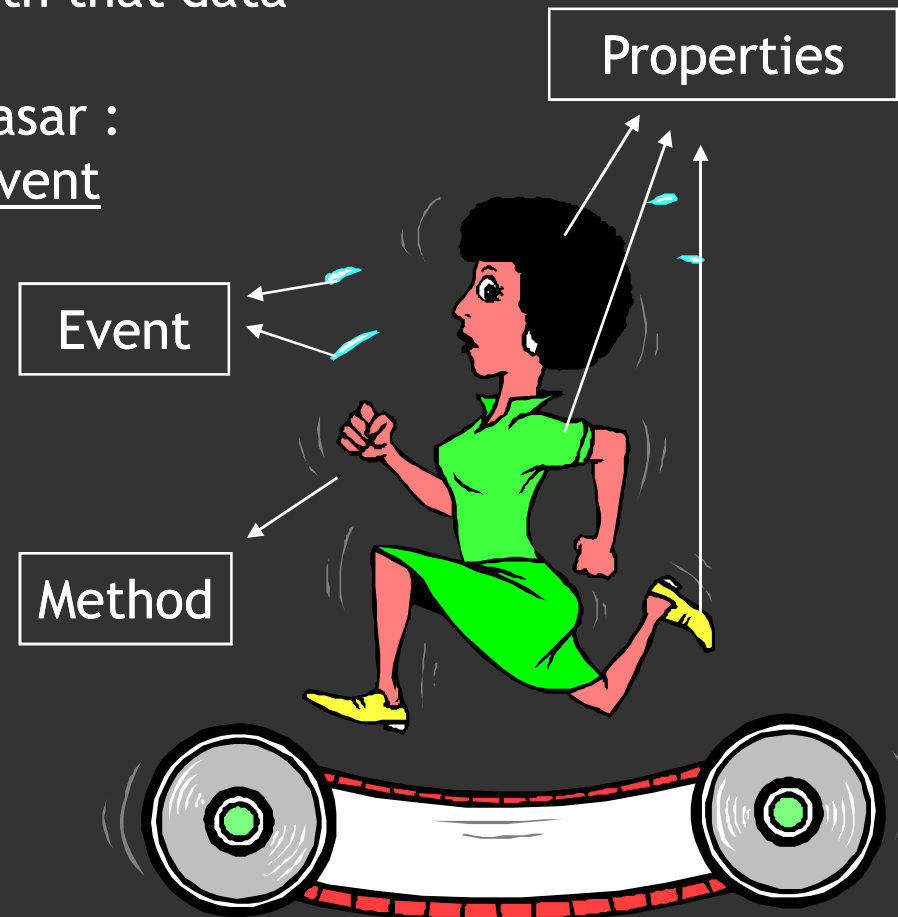
Nilai/karakteristik yang dimiliki sebuah objek

## Method

Serangkaian perintah didalam sebuah objek (Sub/Function)

## Event

Kejadian yang dialami oleh sebuah objek



# Abstraction

Membatasi agar tidak semua pihak  
luar boleh mengakses member  
dari sebuah objek  
Implementasi : Access Modifier





# Access Modifier

## Friend

Member hanya dapat diakses dari “Current Project” (class itu sendiri / class lain)

## Private

Member hanya dapat diakses oleh class itu sendiri

## Protected

Member hanya dapat diakses oleh class itu sendiri dan class turunannya

## Protected Friend

Member dapat diakses dari “Current Project”, class itu sendiri dan class turunannya

## Public

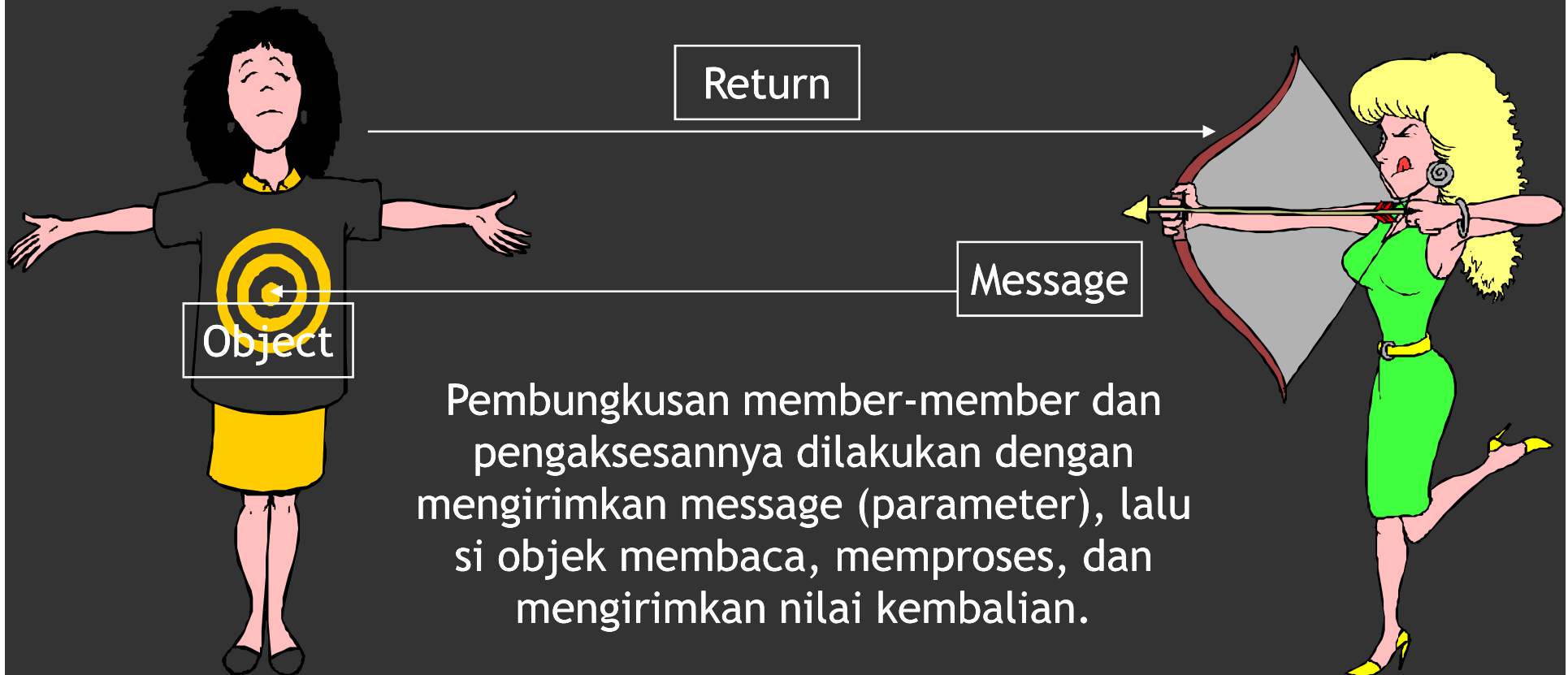
Member dapat diakses oleh siapa saja baik class itu sendiri atau class lain baik di dalam aplikasi maupun di luar aplikasi

# Encapsulation

“Encapsulation is the process in which no direct access is granted to the data; instead, it is hidden”

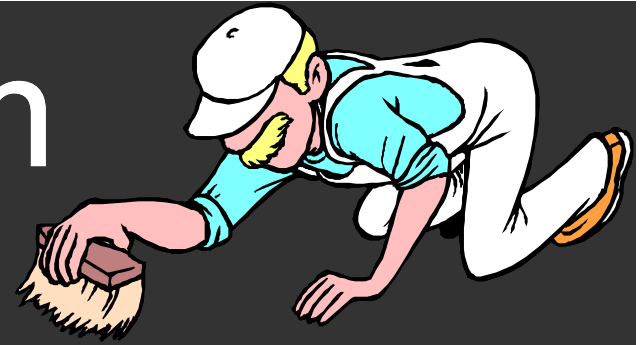
(Daniel R.Clark)

Implementasi : File DLL (ByRef & ByVal) Keyword Shared



# Polymorphism

Memungkinkan method-method yang berbeda memiliki nama yang sama, namun memproses data dan hasil kembalian yang sama ataupun berbeda



Sapu

Sikat

Membersihkan

Lap, Sapu

Lap, Sikat



# Implementing Polymorphism

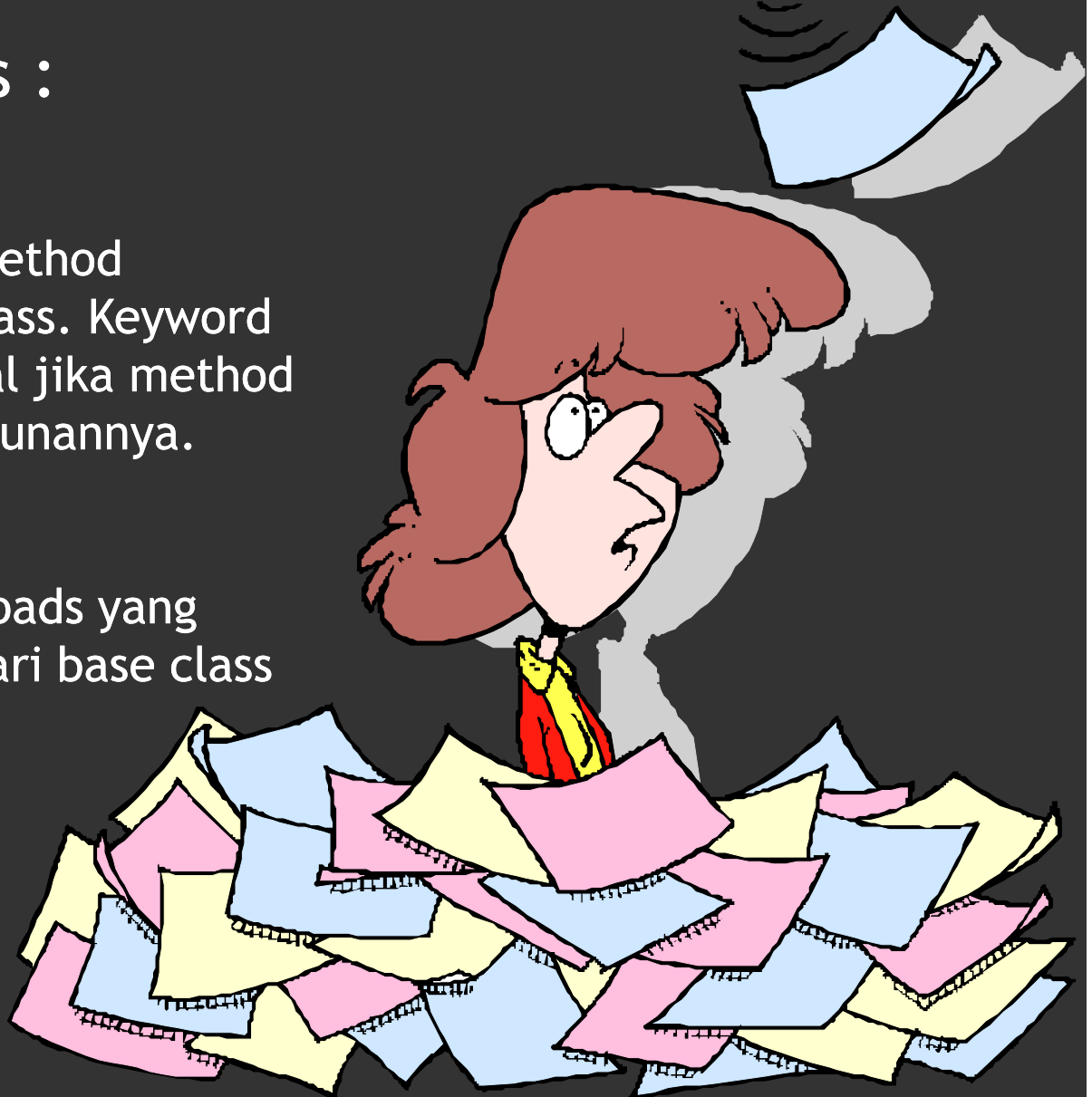
## Overloading Methods :

### Overloads Keyword

Untuk membentuk sebuah method overloading di dalam satu class. Keyword “Overloads” bersifat optional jika method tidak diturunkan ke class turunannya.

### Shadows Keyword

Merupakan lawan dari Overloads yang mencegah sebuah method dari base class untuk di-Overloads.





# Implementing Polymorphism

## Overloading Methods :

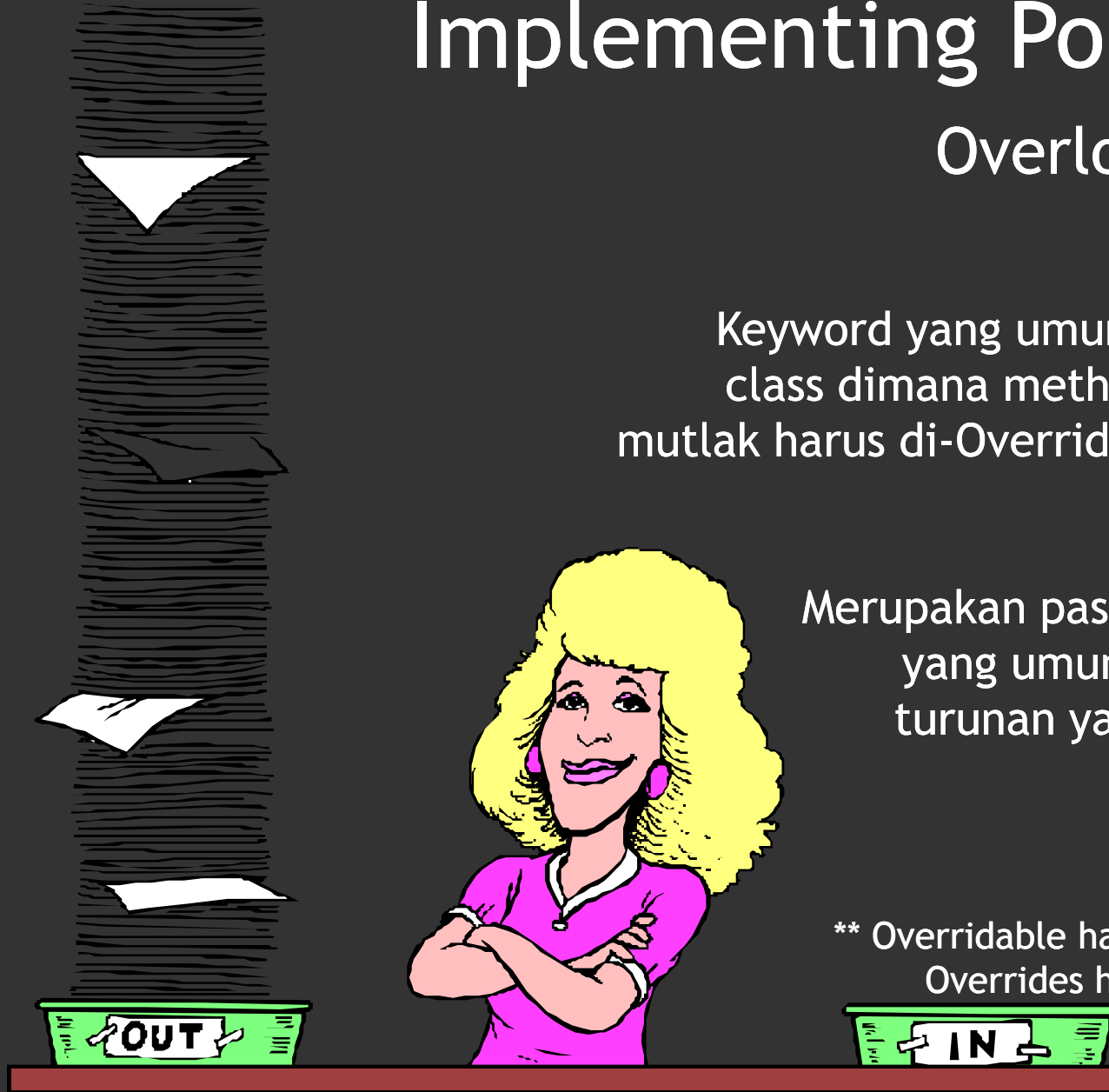
### Overridable Keyword

Keyword yang umumnya dibuat pada base class dimana method yang di-Overridable mutlak harus di-Overrides di class turunannya.

### Overrides Keyword

Merupakan pasangan dari Overridable yang umumnya dibuat pada class turunan yang merupakan turunan dari method base class yang di-Overridable.

\*\* Overridable harus berada di base class dan Overrides harus berada di class turunan



# Inheritance

Mengelompokkan dan menggabungkan member-member base class dan mewariskannya ke class turunannya.

Implementasi : Keyword Inherits



# Aggregation

Suatu kondisi dimana sebuah objek yang terdiri dari berbagai objek bekerja bersama-sama

