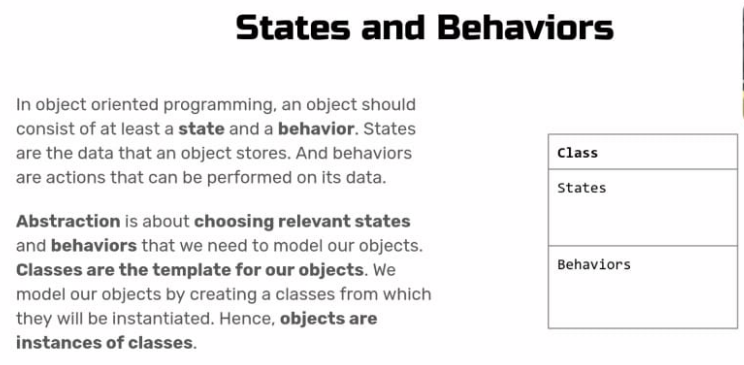
Procedural oop

Urutan bikin objek yaitu new

Perubahan berkali kali perubahan hanya didalam classnya aja



Abstarkasi, menyembukan ditel yang tidak berguna, kareana bagi pengguna dia cuman pegnen tau hasil, rumus kan gaperlu.

sebuah objek/class harus punya sebuah state dan behavior

initialize itu kshus yang bikin objek dari kelas person

Ketika permainan dimulai,

hero dibuat dengan hp

to\_s konversi ke text

to\_i konversi ke integer

to\_s cuman 1, tar nanti ada overloading

def to\_s, puts maka langsung kepanggil

setiap bikin kelas, initialize dulu vaiable nya

baru pake di method getpanjang, setpanjang

class Box

# constructor method

def initialize(w,h)

@width, @height = w, h

end

# accessor methods

def getWidth

@width

end

def getHeight

@height

end

# setter methods

def setWidth=(value)

@width = value

end

def setHeight=(value)

@height = value

end

end

# create an object

box = Box.new(10, 20)

# use setter methods

box.setWidth = 30

box.setHeight = 50

# use accessor methods

x = box.getWidth()

y = box.getHeight()

puts "Width of the box is : #{x}"

puts "Height of the box is : #{y}"

in person.rb

class Person

    def initialize(name, hp, atk) //ini state

        @name = name

        @hp = hp

        @atk = atk

    end

    def to\_s //ini behaviour

        "#{@name} has #{@hp}"

    end

end

in hero.rb

require\_relative 'person'

jin = Person.new("Jin",100,50)

kotun = Person.new("Kotun",500,50)

jin.atk = 1

kotun.atk = 0

puts jin

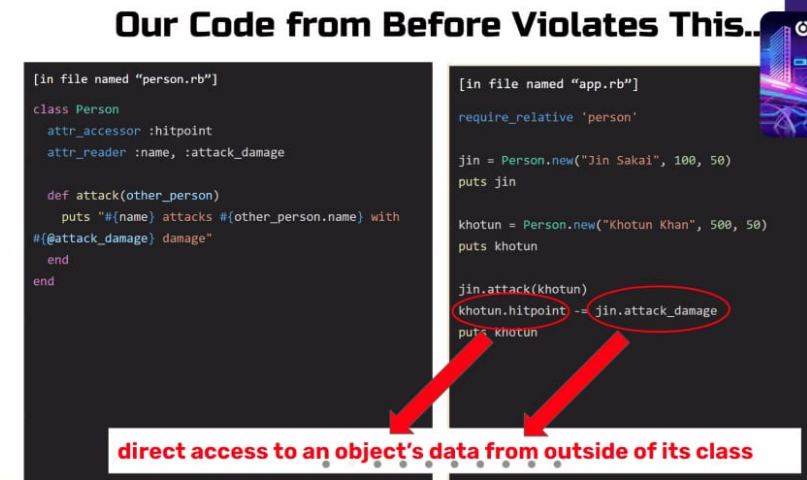
puts kotun

attr\_header :name, :hp, :atk

untuk menghindari geter dan seter

attr\_accessor :hitpoint

kaya get set, gaperlu pake reader lagi



Ini hitpoint berkurang secara langsung dengan accessor, gabole akses langsung brarti ga abstraksi

-enkapsulasi

adalah car akita membatasi akes langsung ke data2 suatu objek, seperti atributnya, kasih cara untuk memanipulasi data2 tersebut, menyembunyikan cara kerja internal, jadi ga kasih akses langsung, mungkin sperti geter ster

ruby itu duck typing

selama variable yang dipasang parameter maka yang diapke parameter itu

ext vscode: ruby solargraph

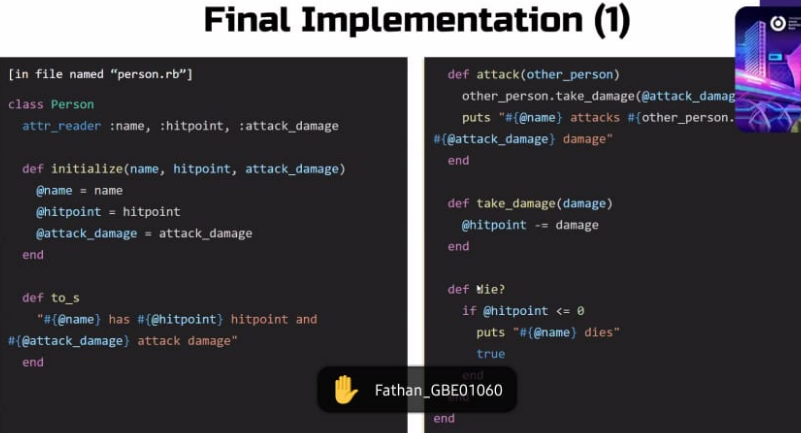
puts self nge refer ke objek diri sendiri

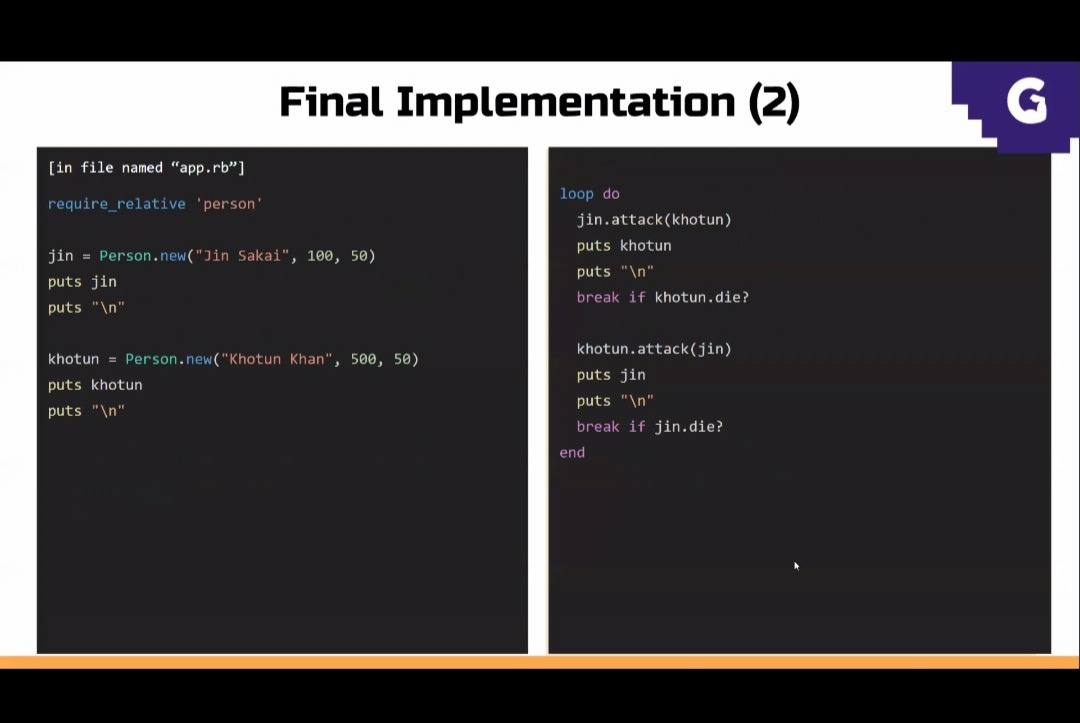
puts itu method

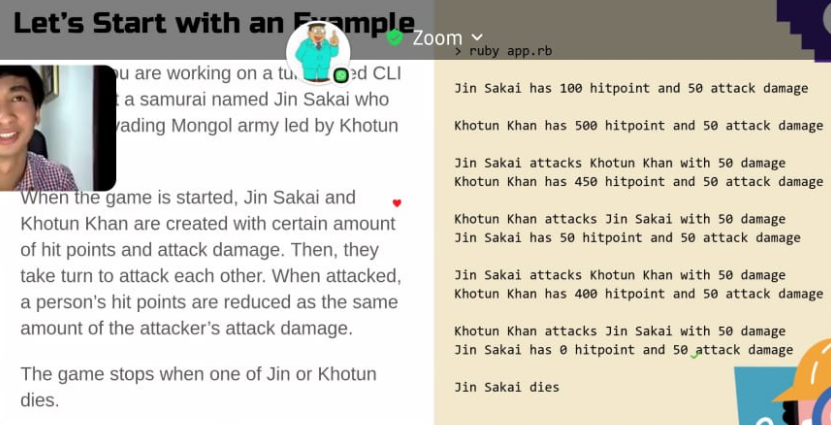
@ itu nandain untuk atrbiut kelas

Ruby minds mirip jetbrains untuk debug

Dia nyari to\_s yang return string tapi adanya puts makanya keluar hash hash







Ada chance 80% hero bisa nangkis

class Person

    attr\_reader :name, :hitpoint, :attack\_damage #kaya geter seternya

    # dalam kasus ini supaya other\_person bisa akses namenya

                     #parameter

    def initialize(name, hitpoint, attack\_damage) #inisialisasi (state)

      @name = name #atribut

      @hitpoint = hitpoint #atribut

      @attack\_damage = attack\_damage #atribut

    end

    def attack(other\_person)

        if other\_person.name == "Jin"

            if rand() <= 0.8 #rand() nilainya (0-1), kalo kurang dari sama dengan 0.8 maka ketangkis

                puts "#{@name} attacks #{other\_person.name} with #{@attack\_damage} damage"

                      # namenya si person bukan other person

                puts "#{other\_person.name} menangkis serangan #{@name}"

            else #kalo hasilnya lebih dari 0.8 maka tidak ketangkis

                other\_person.take\_damage(@attack\_damage)

                puts "#{@name} attacks #{other\_person.name} with #{@attack\_damage} damage"

            end

        else #nah ini berarti jin yang serang khotun, khotun kan gabisa nangkis

            other\_person.take\_damage(@attack\_damage) #parameternya attack\_damage si person bukan other person

                         #method take\_damage yang dibawah

            puts "#{@name} attacks #{other\_person.name} with #{@attack\_damage} damage"

        end

    end

    def take\_damage(damage) #kena damage

                   #parameter attack damage si person

        @hitpoint -= damage #hp nya si other person, karena dipanggil dgn other\_person.take\_damage

    end

    def to\_s #to string, akan muncul kalo pake puts

        "#{@name} has #{@hitpoint} hitpoint and #{@attack\_damage} attack damage"

    end

    def die?

        if @hitpoint <= 0

            puts "#{@name} dies"

            true #supaya keluar dari loop dengan menghasilkan true, true nilainya 1 (bool)

        end

    end

end

require\_relative 'person'

jin = Person.new("Jin",100,50) #urutannya harus sesuai dengan parameter state di class

                 #nama,hp,atk

puts jin #cari to s atau balikannya objek

puts "\n"

khotun = Person.new("Khotun",500,50)

puts khotun

puts "\n"

loop do

    jin.attack(khotun)

       #method #other\_person

       #attack

    puts khotun

    puts "\n"

    break if khotun.die? #break if maksutnya break jika khotun.die ini hasilnya

                    #method die?

    khotun.attack(jin)

    puts jin

    puts "\n"

    break if jin.die?

end