

# **41STUDIO**

## **Training Sessions**

# TIPE DATA

- 1. Integer**
- 2. Float**
- 3. String**
- 4. Boolean**
- 5. Array**
- 6. Hash**
- 7. Time**
- 8. Date**
- 9. File**

# INTEGER

**Ex: 0, 1, 2, 1000, - 20**

# FLOAT

**Ex: 3.14, -200.5, 1002. 50**

# STRING

**Ex: "41studio"**

# BOOLEAN

**Ex: true, false**



# ARRAY

**Ex: [1, "Bandung", false, 3.14, {}]**

# HASH

**Ex: {student1: "foo", student2: "bar"}**



# VARIABLE

**variable == Tempat penampungan / penyimpanan data sementara**

**Ex:**

**name = "John Doe"**

# **ATURAN PENAMAAN VARIABLE**

- 1. Harus huruf kecil semua**
- 2. Harus dimulai dengan character (a-z) diikuti oleh numerik jika dibutuhkan**
- 3. Harus deskriptif**
- 4. Harus menggunakan bahasa inggris**
- 5. Gunakan character underscore “\_” untuk memisahkan setiap kata. Ex: “student\_name”**

# MACAM-MACAM VARIABLE

**Variable local**

**ex: name = "david"**

**instant variable**

**ex: @name = "david"**

**class variable**

**ex: @@name = "david"**

**global variable**

**ex : \$name = "david"**

# CONSTANT

**Konstanta == tempat penampungan data yang hanya bisa satu kali diisi. Nilainya akan tetap selama program berjalan**

# **ATURAN PENAMAAN KONSTANTA**

- 1. Harus dimulai / diawali dengan character (A-Z) diikuti oleh numerik jika dibutuhkan**
- 2. Harus huruf besar semua**
- 3. Harus deskriptif**
- 4. Harus menggunakan bahasa inggris**
- 5. Gunakan character underscore “\_” untuk memisahkan setiap kata. Ex: “STUDENT\_NAME”**



# EXAMPLE

**PHI = 3.14**

**NAME = “john doe”**

**STUDENT\_NAME = “stevent spielberg”**



# FUNGSI

**Semua fungsi di Ruby akan mengembalikan nilai balik.**

**Ex:**

```
def say_hello  
  puts "hello world!"  
end
```

# **ATURAN PENAMAAN FUNGSI**

- 1. Harus dimulai / diawali dengan character (a-z) diikuti oleh numerik jika dibutuhkan**
- 2. Harus huruf kecil semua**
- 3. Harus deskriptif**
- 4. Harus menggunakan bahasa inggris**
- 5. Gunakan character underscore “\_” untuk memisahkan setiap kata. Ex:  
“say\_hello”**

# CLASS

**Aturan penamaan class :**

- 1. Harus dimulai / diawali dengan character (A-Z) diikuti oleh numerik jika dibutuhkan**
- 2. Harus dimulai / diawali dengan huruf besar lalu diikuti oleh huruf kecil**
- 3. Harus descriptif**
- 4. Harus menggunakan bahasa inggris**
- 5. Gunakan character huruf besar untuk memisahkan setiap kata. Ex: "HelloWorld"**

# EXAMPLE

```
class HelloWorld  
end
```

# CONTOH PENGGUNAAN CLASS #1

```
class Student  
  def name=(n)  
    @name = n  
end
```

```
  def name  
    @name  
end  
end
```

```
# instantiate  
student = Student.new  
student.name="john"  
student.name #=> "john"
```



# CONTOH PENGGUNAAN CLASS #2

```
class Student  
  attr_accessor :name, :age, :address  
end
```

```
student = Student.new  
# instantiate  
student = Student.new  
student.name="john"  
student.name #=> "john"
```



# SETTER & GETTER

```
class Student  
  # Setter  
  def name=(n)  
    @name = n  
  end  
  
  # Getter  
  def name  
    @name  
  end  
end
```

# ATTRIBUTE ACCESSOR, WRITER & READER

```
class Student  
  attr_accessor :name # Setter & Getter  
  attr_reader :address # Getter  
  attr_writer :gender # Setter  
end
```

# CONSTRUCTOR DALAM CLASS

**Constructor == method di dalam class yang akan diinvoke pertama kali saat membuat object dari suatu class.**

**Constructor di Ruby nama methodnya adalah “initialize”**

# EXAMPLE

```
class Student  
  attr_accessor :name  
  
  def initialize  
    @name = "john"  
  end  
end  
  
student = Student.new  
student.name #=> john
```

**# constructor dengan parameter**

**class Student**  
  **attr\_accessor :name**

**def initialize(n)**  
    **@name = n**  
  **end**  
**end**

**student = Student.new("john")**  
**student.name**



## **Instant Method**

**instant method == method yang hanya bisa dipanggil dari object**

## **Class Method**

**class method == method yang hanya bisa dipanggil dari classnya**

## **Class Variable**

**class variable == variable yang data / isinya available dan sama di setiap object atau dengan kata lain digunakan untuk sharing data antar object**



# EXAMPLE

```
class Student  
  #class variable  
  @@counter
```

```
  def initialize  
    @@counter || = 0  
    @@counter += 1  
  end
```

```
#instant method  
def name  
  puts "john"  
end
```

```
#class method  
def self.say_hello  
  puts "Hello World!"  
end
```

```
def show_counter  
  puts @@counter  
end  
end
```

**Student.say\_hello # calling class method**

**student = Student.new**

**student.name # calling instant method**

**student.show\_counter #=> 1**

**student2 = Student.new**

**student2.show\_counter #=> 2**

**student.show\_counter #=> 2**

# INHERITANCE

**inheritance == penurunan / pewarisan**

```
class Parent  
  def say_hello  
    puts "Hello World"  
  end  
end
```

```
class Children < Parent  
end
```

```
child = Children.new  
child.say_hello
```

# OVERRIDING

```
class Parent  
  def say_hello  
    puts "Hello from parent"  
  end  
  
  def say_hi  
    puts "Hi There"  
  end  
end  
  
class Children < Parent  
  def say_hello  
    puts "Holla from children"  
  end
```



```
# call super for calling parent method  
# with the same name as current  
def say_hi  
  super  
end  
end
```

```
child = Children.new  
child.say_hello  
child.say_hi
```



# OVERLOADING

```
class Parent  
  def say_hello  
    puts "Hello World!"  
  end  
end
```

```
class Children < Parent  
  def say_hello(s)  
    puts s  
  end  
end
```

```
child = Children.new  
child.say_hello("Hi there")
```

# CONDITIONS

## If statement

```
if <condition>  
  #block code in here  
end
```

```
if <condition>  
  #block code in here  
else  
  #block code in here  
end
```

```
if <condition>  
  #block code in here  
elsif <condition>  
  #block code in here  
else  
  #block code in here  
end
```

# CASE STATEMENT

**gender = “male”**

**case gender**

**when “male”**

**puts “you are man..”**

**when “female”**

**puts “You are girl”**

**else**

**puts “oops.. shemale aren’t you ??”**

**end**

# OPERATOR PEMBANDING

**|| or**

**&& and**

**== .eq? (object)**

# LOOPING

```
cities = ["bandung","jakarta","cimahi","denpasar","makasar","palembang"]
```

```
# Using "for" statement  
for city in cities  
  puts city  
end
```

```
# Using "each" statement  
cities.each do |city|  
  puts city  
end
```

```
# it will loop 10 times  
10.times do  
  puts "hello #{counter}"  
end
```



```
1.upto(10) do  
  puts “hello world!”  
end
```

```
1.upto(10) do |counter|  
  puts “hello #{counter}”  
end
```

```
10.downto(1)  
  puts “hello world!”  
end
```

```
10.downto(1) do |counter|  
  puts “hello #{counter}”  
end
```

# MODULE

**Module digunakan untuk mengelompokkan methods, classes dll yang sering dipake dibanyak tempat sehingga mengurangi penggunaan fungsi yang berulang-ulang.**

# **ATURAN PENAMAAN MODULE**

- 1. Harus dimulai / diawali dengan character (A-Z) diikuti oleh numerik jika dibutuhkan**
- 2. Harus dimulai / diawali dengan huruf besar lalu diikuti oleh huruf kecil**
- 3. Harus descriptif**
- 4. Harus menggunakan bahasa inggris**
- 5. Gunakan character huruf besar untuk memisahkan setiap kata. Ex: "HelloWorld"**

# EXAMPLE

```
module HelloWorld  
  def say_hello  
    puts "Hello World"  
  end  
end
```



# **MIXIN MODULE**

**Gabungan functions, classes dan modules di dalam satu buah module.**

```
module HelloWorld  
  def say_hello  
    puts "Hello World"  
  end
```

```
class Something  
  # block class in here  
end
```

```
module AnotherModule  
  # block module in here  
end  
end
```

**Next yang harus dipelajari methods untuk classes :**

**1. Integer**

**2. Float**

**3. String**

**4. Boolean**

**5. Array**

**6. Hash**

**7. Time**

**8. Date**

**9. File**

**Bisa baca di ruby api doc <http://www.ruby-doc.org/core-1.9.3/>**

**THANKS**