

# Referensi Cepat Ruby

Objek, block, iterator, regex, dan file I/O esensial

## Dasar

### Hello World

```
puts "Hello, World!"
print "no newLine"
p [1, 2, 3] # output inspect: [1, 2, 3]
```

### Jalankan Ruby

```
ruby script.rb # jalankan file
ruby -e 'puts "hi"' # jalankan inline
irb # REPL interaktif
```

## Variabel

<b>name</b>	Variabel lokal
<b>@name</b>	Variabel instance
<b>@@count</b>	Variabel class
<b>\$debug</b>	Variabel global
<b>MAX_SIZE</b>	Konstanta (huruf besar menurut konvensi)

## Tipe

42.class	# Integer
3.14.class	# Float
"hello".class	# String
true.class	# TrueClass
nil.class	# NilClass
:symbol.class	# Symbol

## String

### Dasar String

```
name = "World"
puts "Hello, #{name}!" # interpolasi (double quotes)
puts 'No #{interpolation}' # literal (single quotes)
multi = <<-HEREDOC
heredoc dengan indent
HEREDOC
```

## Metode String

<b>.length / .size</b>	Jumlah karakter
<b>.upcase / .downcase</b>	Konversi huruf besar/kecil
<b>.strip</b>	Hapus whitespace awal/akhir
<b>.split(' ,')</b>	Pecah menjadi array
<b>.gsub(/pat/, 'rep')</b>	Substitusi global
<b>.include?('sub')</b>	Periksa apakah mengandung substring
<b>.start_with?('pre')</b>	Periksa prefix
<b>.chars / .bytes</b>	Array karakter / byte
<b>.to_i / .to_f</b>	Konversi ke integer / float
<b>.freeze</b>	Buat string tak dapat diubah

## Array & Hash

### Array

```
arr = [1, "two", :three]
arr << 4 # push (tambahkan)
# 1
arr[0] # 1
arr[-1] # 4 (elemen terakhir)
arr[1..2] # ["two", :three] (slice)
```

## Metode Array

<b>.push / .pop</b>	Tambah/hapus dari akhir
<b>.shift / .unshift</b>	Hapus/tambah dari awal
<b>.flatten</b>	Ratakan array bersarang
<b>.compact</b>	Hapus nilai nil
<b>.uniq</b>	Hapus duplikat
<b>.sort / .reverse</b>	Urutkan / balik urutan
<b>.map {  x  x * 2 }</b>	Transformasi tiap elemen
<b>.select {  x  x &gt; 0 }</b>	Filter elemen
<b>.reduce(0) {  sum, x  sum + x }</b>	Akumulasikan menjadi satu nilai

## Hash

```
user = { name: "Alice", age: 30 } # key symbol
old = { "key" => "value" } # key string
user[:name] # "Alice"
user[:email] = "a@b.com" # tambah pasangan
user.fetch(:name, "default") # dengan default
```

## Metode Hash

<b>.keys / .values</b>	Array key / nilai
<b>.each {  k, v  }</b>	Iterasi pasangan key-value
<b>.merge(other)</b>	Gabungkan dua hash
<b>.key?(k) / .value?(v)</b>	Periksa keberadaan
<b>.select {  k, v  }</b>	Filter pasangan
<b>.transform_values {  v  }</b>	Transformasi semua nilai

## Alur Kontrol

### Kondisional

```
if score >= 90 then "A"
elsif score >= 80 then "B"
else "C"
end
puts "adult" if age >= 18 # if inline
puts "minor" unless age >= 18 # unless inline
```

### Case / When

```
case status
when :ok then puts "success"
when :error then puts "failed"
when 400..499 then puts "client error"
else puts "unknown"
end
```

## Loop

```
5.times { |i| puts i }
(1..10).each { |n| puts n }
while condition do end
until condition do end
loop { break if done }
```

## Ternary & Logika

```
status = age >= 18 ? "adult" : "minor"
name = input || "default" # or-assign
name ||= "fallback" # efek sama
```

## Metode

### Mendefinisikan Metode

```
def greet(name, greeting = "Hello")
  "#{greeting}, #{name}!"
end
greet("Alice") # "Hello, Alice!"
greet("Bob", "Hi") # "Hi, Bob!"
```

## Nilai Kembali

```
def add(a, b)
  a + b # ekspresi terakhir adalah return implisit
end
def divide(a, b)
  return nil if b == 0
  a.to_f / b
end
```

## Argumen Keyword & Splat

```
def connect(host:, port: 80, **opts)
  puts "#{host}:#{port} #{opts}"
end
def log(*messages)
  messages.each { |m| puts m }
end
```

## Konvensi Metode

<b>method?</b>	Mengembalikan boolean (predikat)
<b>method!</b>	Mengubah receiver (bang method)
<b>self.method</b>	Definisi class method

## Class

### Definisi Class

```
class User
  attr_accessor :name, :email
  def initialize(name, email)
    @name = name
    @email = email
  end
end
```

## Inheritance

```
class Admin < User
  def initialize(name, email, level)
    super(name, email)
    @level = level
  end
end
```

## Kontrol Akses

<b>public</b>	Default; dapat diakses dari mana saja
<b>private</b>	Hanya dapat diakses di dalam class
<b>protected</b>	Dapat diakses dalam class dan subclass
<b>attr_reader</b>	Buat metode getter
<b>attr_writer</b>	Buat metode setter
<b>attr_accessor</b>	Buat getter dan setter

## Modul

### Mixin

```
module Greetable
  def greet
    "Hello, I'm #{name}"
  end
end
class User; include Greetable; end
```

## Namespace

```
module Payment
  class Processor
    def charge(amount) end
  end
end
p = Payment::Processor.new
```

# Referensi Cepat Ruby

## Include vs Extend

<b>include ModName</b>	Tambahkan sebagai instance method
<b>extend ModName</b>	Tambahkan sebagai class method
<b>prepend ModName</b>	Sisipkan sebelum class dalam pencarian metode

## Block & Iterator

### Sintaks Block

```
[1, 2, 3].each { |n| puts n }      # block satu baris
[1, 2, 3].each do |n|
  puts n                          # block multi-baris
end
```

### Yield

```
def with_logging
  puts "start"
  result = yield
  puts "end"
  result
end
with_logging { expensive_operation }
```

## Proc & Lambda

```
square = Proc.new { |x| x ** 2 }
square.call(5)          # 25
double = ->(x) { x * 2 } # lambda
double.call(3)         # 6
[1, 2, 3].map(&square)  # [1, 4, 9]
```

## Iterator Umum

<b>.each</b>	Iterasi tiap elemen
<b>.map / .collect</b>	Transformasi tiap elemen
<b>.select / .filter</b>	Pertahankan elemen yang cocok
<b>.reject</b>	Hapus elemen yang cocok
<b>.reduce / .inject</b>	Akumulasikan menjadi satu nilai
<b>.each_with_index</b>	Iterasi dengan indeks
<b>.flat_map</b>	Map dan ratakan satu level
<b>.any? / .all? / .none?</b>	Pemeriksaan boolean pada koleksi

## Regex

### Pencocokan

```
"hello 42" =~ /\d+/      # 6 (posisi kecocokan)
"hello" =~ /\d+/        # nil (tidak cocok)
"hello".match?(/ell/)   # true
md = "age: 30".match(/(\d+)/)
md[1]                   # "30"
```

### Pola Umum

<b>/^start/</b>	Dianchor di awal
<b>/end\$/</b>	Dianchor di akhir
<b>/\d+/</b>	Satu atau lebih digit
<b>/\w+/</b>	Karakter kata
<b>/\s+/</b>	Whitespace
<b>/[a-z]+/i</b>	Case-insensitive
<b>/(group)/</b>	Capture group

### Substitusi

```
"hello world".sub(/world/, "Ruby") # kecocokan pertama
"aabba".gsub(/a/, "x")            # semua: "xxbbx"
"foo bar".gsub(/(\w+)/) { $1.upcase } # "FOO BAR"
```

## File I/O

### Baca & Tulis

```
content = File.read("data.txt")
lines = File.readlines("data.txt", chomp: true)
File.write("out.txt", "hello\n")
File.open("log.txt", "a") { |f| f.puts "entry" }
```

### Operasi File

<b>File.exist?(path)</b>	Periksa apakah file ada
<b>File.directory?(path)</b>	Periksa apakah path adalah direktori
<b>File.basename(path)</b>	Nama file tanpa direktori
<b>File.extname(path)</b>	Ekstensi file
<b>File.size(path)</b>	Ukuran file dalam byte
<b>File.delete(path)</b>	Hapus file
<b>Dir.glob('*.*rb')</b>	Temukan file yang cocok pola
<b>FileUtils.mkdir_p(path)</b>	Buat direktori secara rekursif

### CSV & JSON

```
require "json"
data = JSON.parse(File.read("data.json"))
File.write("out.json", JSON.pretty_generate(data))
require "csv"
CSV.foreach("data.csv", headers: true) { |row| puts row["name"] }
```