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

`name`	Variabel lokal
`@name`	Variabel instance
`@@count`	Variabel class
`$debug`	Variabel global
`MAX_SIZE`	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

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

Array & Hash

Array

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

Metode Array

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

`.keys / .values`	Array key / nilai
`.each { |k, v| }`	Iterasi pasangan key-value
`.merge(other)`	Gabungkan dua hash
`.key?(k) / .value?(v)`	Periksa keberadaan
`.select { |k, v| }`	Filter pasangan
`.transform_values { |v| }`	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 Kembalian

```
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

`method?`	Mengembalikan boolean (predikat)
`method!`	Mengubah receiver (bang method)
`self.method`	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

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

Include vs Extend

`include ModName`	Tambahkan sebagai instance method
`extend ModName`	Tambahkan sebagai class method
`prepend ModName`	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

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

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

`File.exist?(path)`	Periksa apakah file ada
`File.directory?(path)`	Periksa apakah path adalah direktori
`File.basename(path)`	Nama file tanpa direktori
`File.extname(path)`	Ekstensi file
`File.size(path)`	Ukuran file dalam byte
`File.delete(path)`	Hapus file
`Dir.glob('*.rb')`	Temukan file yang cocok pola
`FileUtils.mkdir_p(path)`	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"] }
```