



# Struktur

## Data

presented by:

- Muhammad Iqbal K
- Ogi Pambudi



# Contents

It's a summary of what we'll discuss today.

01

Pengertian

02

Struktur data  
pada ruby

03

Macam Macam  
Struktur data

04

Penutup

Struktur data



Struktur data menjadi salah satu bagian penting yang wajib dipahami oleh para programmer. Struktur ini memudahkan pengguna mengakses data yang dibutuhkan secara cepat dan tepat. Ini karena struktur data memiliki format khusus yang berfungsi untuk mengatur, memproses, mengambil, dan menyimpan data.

[www.reallygreatsite.com](http://www.reallygreatsite.com)

# ARRAY 01

Array adalah kumpulan objek yang diatur secara teratur. Dalam Ruby, array dapat berisi objek apa saja dan indeks dimulai dari 0. Untuk membuat array baru, gunakan kurung siku dan pisahkan setiap elemen dengan koma.

```
arr = Array.new(10)
i=0
while i < 10
  arr[i] = i
  i += 1
end
print arr
```



## HASH 02

Hash adalah kumpulan pasangan kunci-nilai yang diatur secara tidak teratur. Dalam Ruby, kunci dan nilai dapat berupa objek apa saja. Untuk membuat hash baru, gunakan kurung kurawal dan pisahkan setiap pasangan kunci-nilai dengan tanda titik dua.

```
person = {name: 'John', age: 30, city: 'New York'}
```



kumpulan objek unik yang diatur secara tidak teratur. Dalam Ruby, set tidak memiliki duplikat dan dapat berisi objek apa saja.

```
require 'set'

set = Set.new
set.add('apple')
set.add('banana')
set.add('orange')
```





Queue adalah struktur data yang menerapkan konsep FIFO (First-In-First-Out). Dalam Ruby, Queue dapat diinisialisasi dengan metode "Queue.new".

```
require 'thread'

queue = Queue.new
queue.push('item 1')
queue.push('item 2')
queue.push('item 3')
```



## Stack 05

Struktur data yang menerapkan konsep LIFO (Last-In-First-Out). Dalam Ruby, Stack dapat diinisialisasi dengan metode "Array.new".

```
stack = []  
stack.push('item 1')  
stack.push('item 2')  
stack.push('item 3')
```





## Tree 06

struktur data yang terdiri dari node yang terhubung satu sama lain. Dalam Ruby, Tree dapat diimplementasikan dengan menggunakan kelas.

```
class Node
  attr_accessor :value, :left_child, :right_child

  def initialize(value, left_child=nil, right_child=nil)
    @value = value
    @left_child = left_child
    @right_child = right_child
  end
end

root = Node.new(1)
root.left_child = Node.new(2)
root.right_child = Node.new(3)
```



## Graph 07

Graph adalah struktur data yang terdiri dari node dan edge yang menghubungkan node-node tersebut. Dalam Ruby, graph dapat direpresentasikan menggunakan array atau hash. Graph juga dapat diimplementasikan dengan menggunakan class.

ruby

 Copy code

```
graph = [  
  [1, 2],      # node 0 terhubung dengan node 1 dan 2  
  [0, 2, 3],   # node 1 terhubung dengan node 0, 2, dan 3  
  [0, 1],      # node 2 terhubung dengan node 0 dan 1  
  [1]          # node 3 terhubung dengan node 1  
]
```



## **Linked List**

08

Dalam linked list, setiap node memiliki pointer yang menunjuk ke node berikutnya, sehingga node-node tersebut membentuk sebuah urutan. Namun, tidak seperti array atau list biasa, node-node tersebut tidak harus disimpan secara berurutan dalam memori. Sebagai gantinya, setiap node disimpan di mana saja dalam memori dan pointer digunakan untuk menghubungkannya dengan node lain.



```
class Node
  attr_accessor :value, :next_node

  def initialize(value, next_node=nil)
    @value = value
    @next_node = next_node
  end
end

head = Node.new(1)
second_node = Node.new(2)
third_node = Node.new(3)

head.next_node = second_node
second_node.next_node = third_node
```

Pada class `LinkedList` di sebelah setiap objek akan memiliki atribut `head` untuk menunjuk ke node pertama dalam linked list. Method `add_node` digunakan untuk menambahkan node baru ke dalam linked list dengan cara membuat objek `Node` baru dan menghubungkannya dengan node terakhir dalam linked list menggunakan pointer `next_node`.





## Matrix 09

Matrix pada Ruby adalah struktur data array dua dimensi yang terorganisasi dengan baik. Dalam matrix, setiap elemen memiliki koordinat yang terdiri dari baris dan kolom, dan nilai elemen dapat diakses dengan menggunakan indeks baris dan kolomnya. Matrix pada Ruby tersedia di dalam library standar dengan menggunakan kelas Matrix yang menyediakan banyak metode untuk melakukan operasi matrix seperti transposisi, perkalian, penyelesaian sistem persamaan linear, dan lainnya.

```
require 'matrix'

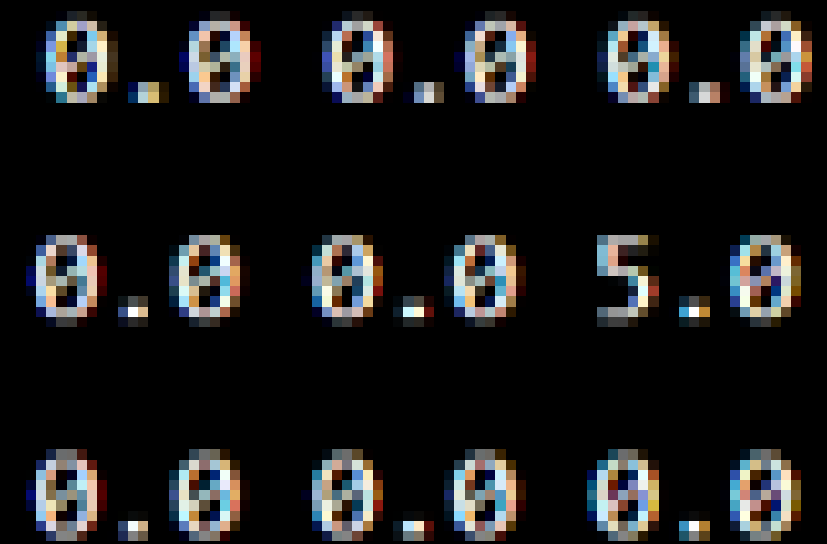
# Membuat matrix 3x3 dengan nilai awal 0
m = Matrix.zero(3)

# Mengakses elemen pada baris 1 kolom 2
puts m[1, 2] #=> 0.0

# Mengubah nilai elemen pada baris 1 kolom 2 menjadi 5
m[1, 2] = 5

# Menampilkan matrix
puts m
```

## Output



0.0	0.0	0.0
0.0	0.0	5.0
0.0	0.0	0.0

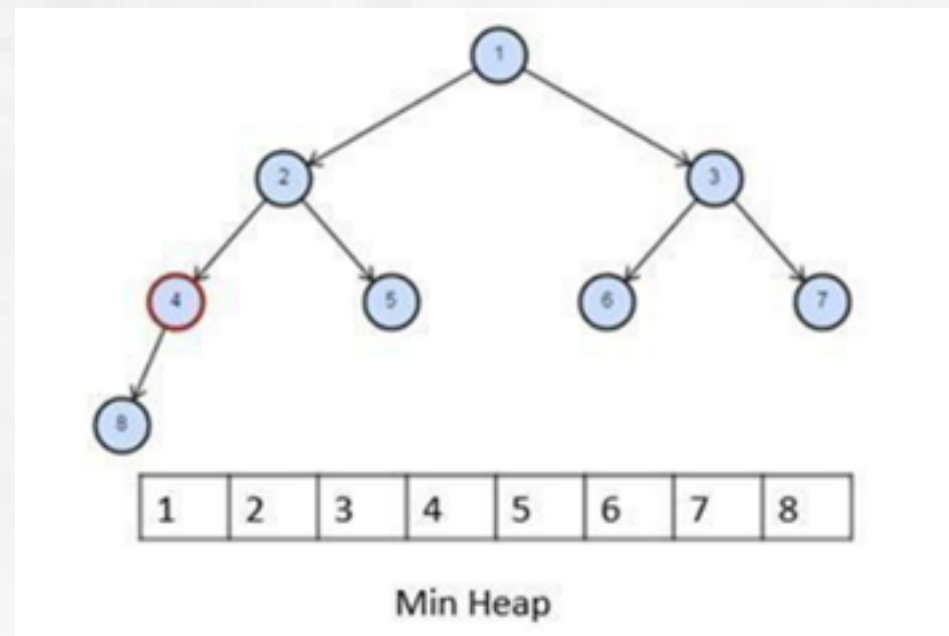




## Heap

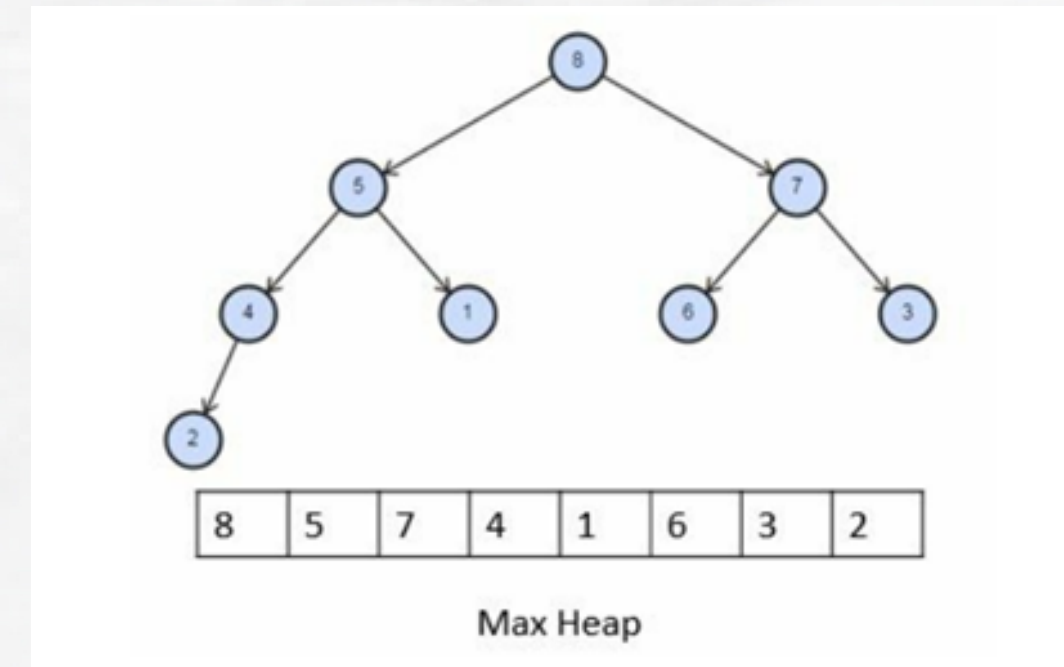
10

Heap adalah salah satu struktur data pada Ruby yang terdiri dari sebuah array/list yang diatur secara teratur dalam bentuk pohon biner. Heap dibagi menjadi dua jenis, yaitu heap maksimum (max heap) dan heap minimum (min heap).



heap minimum merupakan jenis heap yang memiliki elemen dengan nilai terkecil diletakkan pada root dari pohon biner.

**Heap  
Minimum**



Heap maksimum merupakan jenis heap yang memiliki elemen dengan nilai terbesar diletakkan pada root dari pohon biner.

**Heap  
Maksimum**

```
require 'heap'

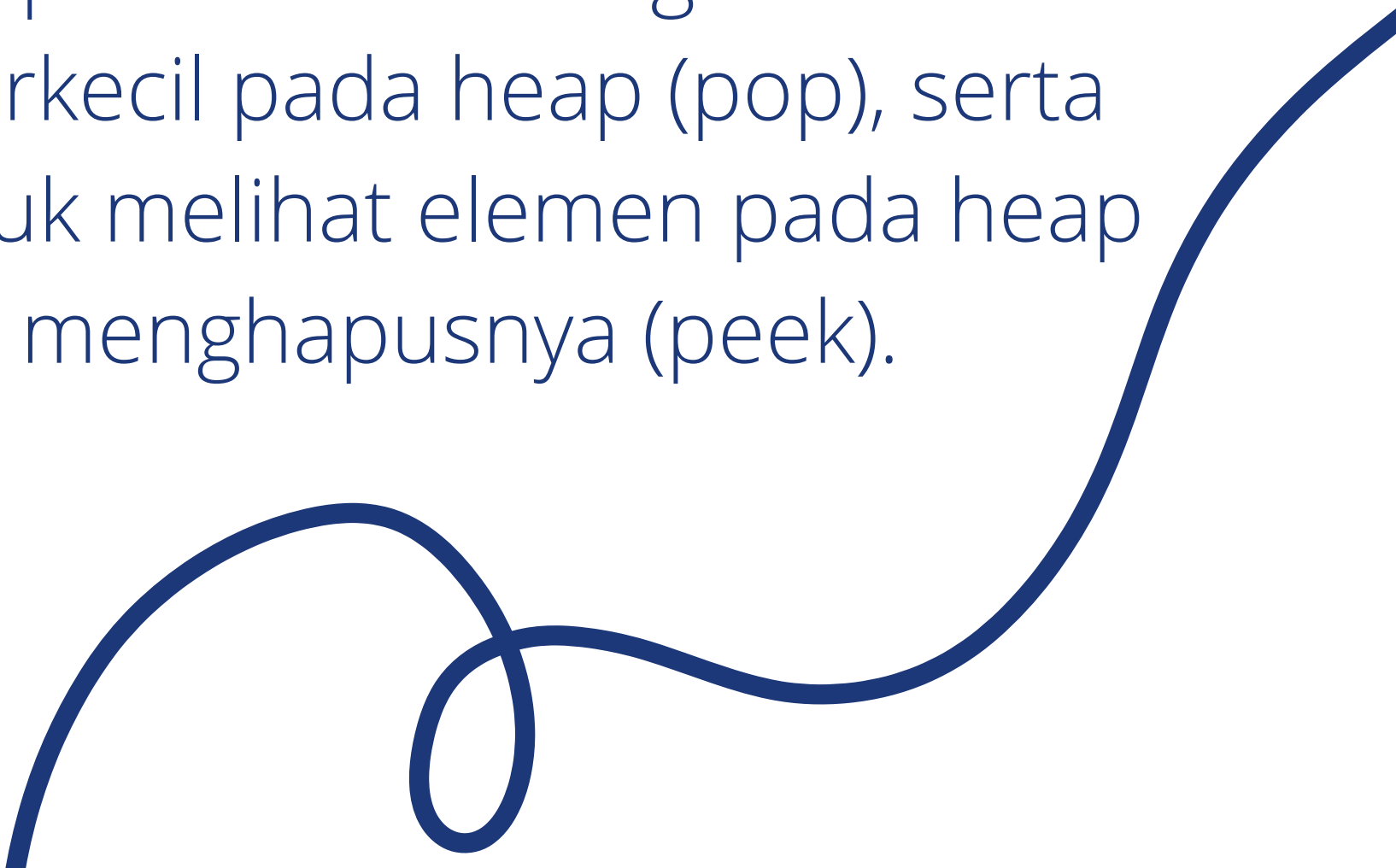
# Membuat heap minimum
min_heap = Heap.new(:<)

# Menambah elemen pada heap
min_heap.push(3)
min_heap.push(1)
min_heap.push(2)

# Menghapus elemen pada heap
min_heap.pop

# Melihat elemen pada heap
min_heap.peak
```

heap dapat diimplementasikan menggunakan kelas Heap yang tersedia pada modul standard library bernama 'heap'. Kelas Heap ini memiliki metode untuk menambah elemen pada heap (push), menghapus elemen dengan nilai terbesar/terkecil pada heap (pop), serta metode untuk melihat elemen pada heap tanpa menghapusnya (peek).





# Set



```
makefile Copy code

# Membuat set kosong
my_set = Set.new

# Membuat set dengan beberapa nilai
my_set = Set.new([1, 2, 3, 3, 4, 5])

# Menambah nilai ke set
my_set.add(6)

# Menghapus nilai dari set
my_set.delete(1)

# Memeriksa apakah sebuah nilai ada di dalam set
puts my_set.include?(4) # Output: true
puts my_set.include?(10) # Output: false
```

Set adalah struktur data yang berisi kumpulan nilai unik tanpa adanya indeks. Set mirip dengan array, namun hanya dapat berisi nilai unik dan tidak memiliki urutan tertentu. Contoh penggunaan set:





Pertanyaan

# 1. Bagaimana cara membuat sebuah Array di Ruby?


- a. dengan menggunakan tanda kurung kurawal {}
- b. dengan menggunakan tanda kurung siku []
- c. dengan menggunakan tanda kurung biasa ()
- d. dengan menggunakan tanda petik dua ""



## 2. Apa yang dilakukan oleh method `Array#pop` di Ruby?

- a. Menambahkan sebuah elemen pada akhir array
- b. Menambahkan sebuah elemen pada awal array
- c. Menghapus sebuah elemen pada akhir array
- d. Menghapus sebuah elemen pada awal array

### **3. Apa jenis-jenis Heap yang ada di Ruby?**

- a. Binary Heap dan N-ary Heap
  - b. Max Heap dan Min Heap
  - c. Balanced Heap dan Unbalanced Heap
  - d. Binary Heap dan Balanced Heap
- 

## 4. Apa yang dilakukan oleh method Hash#[]= di Ruby?

- a. Menambahkan sebuah key-value pair pada hash
- b. Menghapus sebuah key-value pair pada hash
- c. Mengakses nilai dari sebuah key pada hash
- d. Mengakses key dari sebuah nilai pada hash

## **5. Apa yang dimaksud dengan Leaf dalam sebuah Tree?**

- a. Node yang tidak memiliki parent
- b. Node yang terletak paling bawah dari Tree
- c. Node yang memiliki lebih dari satu child
- d. Node yang terletak paling atas dari Tree



## **6. Apa yang dapat dilakukan oleh kelas Heap pada modul standard library di Ruby?**

- A. Menambah elemen pada array
- B. Menghapus elemen pada array
- C. Mengurutkan elemen pada array
- D. Menambah dan menghapus elemen pada heap.



# Hello!

Allow me to introduce myself. I am Greta Mae Evans, the Head of Design in Shodwe Furniture. Today, I will present my brand product and why ours are worth buying. First of all, thank you for coming to this event and listening to my presentation.

[www.reallygreatsite.com](http://www.reallygreatsite.com)



## 7. Apa perbedaan antara Set dan Array?

- a. Set dapat berisi nilai yang duplikat, sedangkan Array tidak
- b. Array memiliki indeks, sedangkan Set tidak
- c. Set memiliki urutan tertentu, sedangkan Array tidak
- d. Set dan Array tidak memiliki perbedaan



**Thank you**  
I really appreciate your time.

[www.reallygreatsite.com](http://www.reallygreatsite.com)