Module 1

Introduction to everything

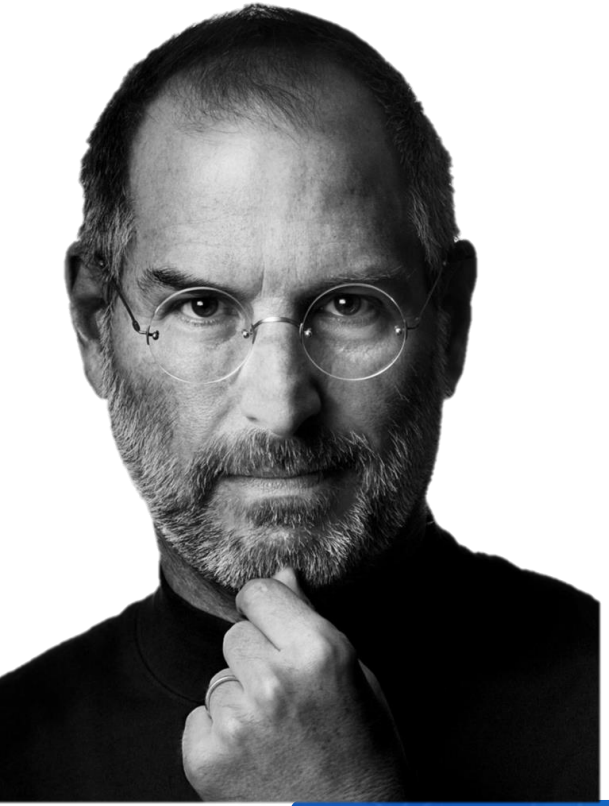
Section

Basic Data Structure & Algorithm

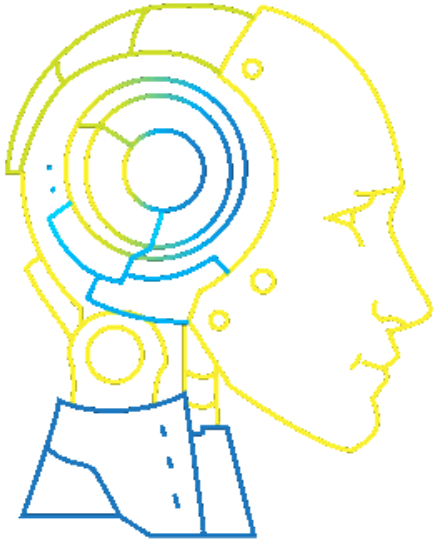
Basic Data Structure & Algorithm

“To develop a complete mind: Study the science of art; Study the art of science. Learn how to see. Realize that everything connects to everything else.”

Leonardo da Vinci



“Everybody should learn to program a computer, because it teaches you how to **think**”
-Steve Jobs



Session I

What is Data Structure?

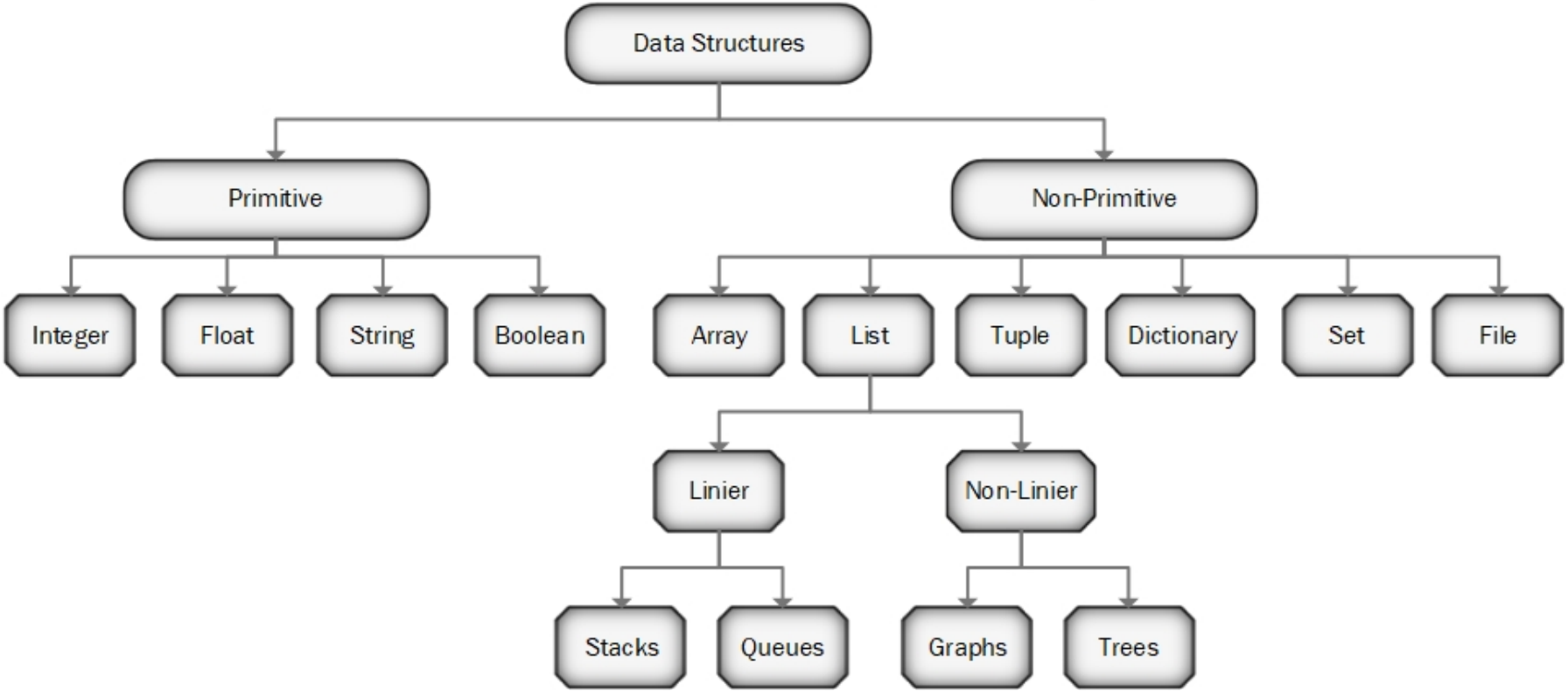
What is Data Structure?

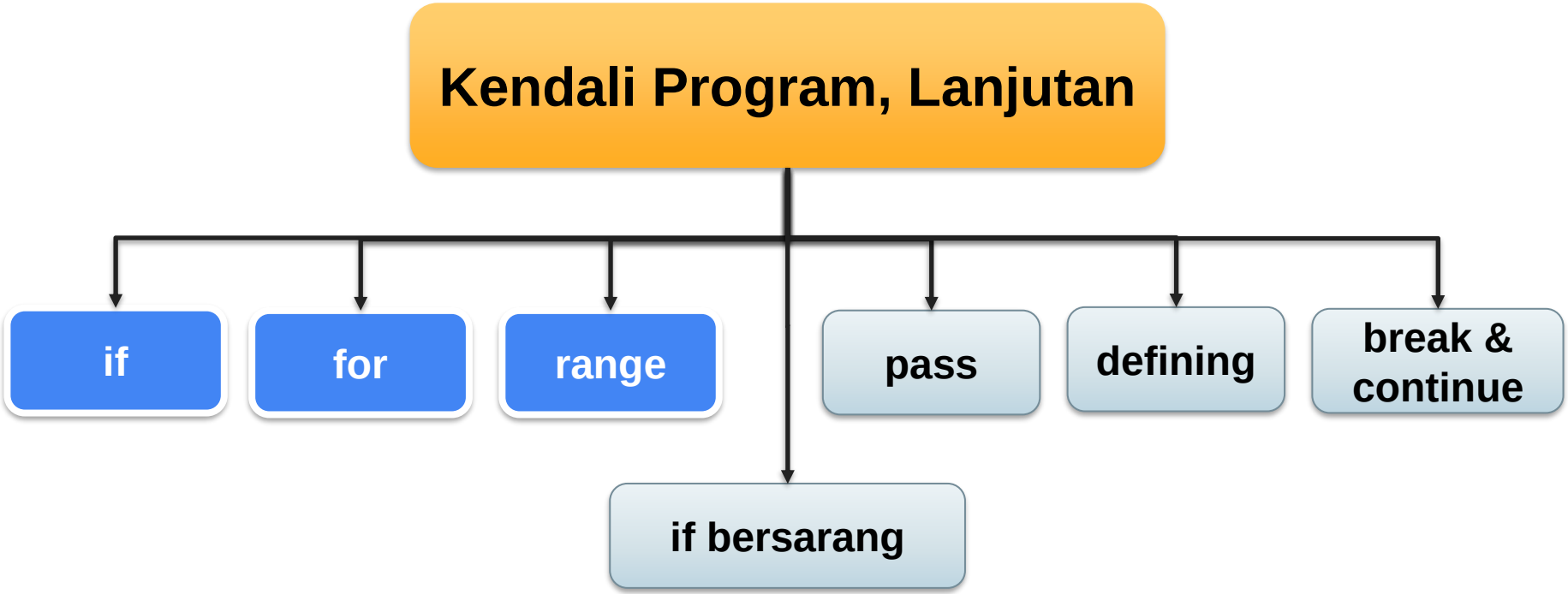
Data Structure adalah sebuah cara mengatur dan menyimpan data, sehingga dapat diakses dan dikerjakan dengan lebih efisien.

Abstract Data Type and Data Structures

Abstract Data Type merupakan struktur data yang dapat membantu kita untuk fokus pada gambaran yang lebih umum, daripada fokus pada detailnya yang dapat membuat kita sulit untuk memahaminya.

Data Structures sebenarnya merupakan implementasi dari Tipe Data Abstrak atau ADT. Implementasi ini membutuhkan tampilan fisik data menggunakan beberapa kumpulan konstruksi pemrograman dan tipe data dasar.


Data Structures
Primitive
Non-Primitive
Integer
Float
String
Boolean
Array
List
Tuple
Dictionary
Set
File
Linier
Non-Linier
Stacks
Queues
Graphs
Trees


Kendali Program, Lanjutan
if
for
range
if bersarang
pass
defining
break & continue

Statement if bersarang

Statemen seleksi **if…else** bersarang untuk beberapa kasus dilakukan dengan menempatkan statemen **if…else** didalam statemen **if…else** lain.

Contoh statement if bersarang

```
1   a = 4
2   b = 3
3   if(a == b):
4       if(a != 4):
5           print("Nilai A sama dengan B namun bukan angka 4")
6       else:
7           print("Nilai A sama dengan B")
8   elif(a > b):
9       print("Nilai A lebih besar dari B")
10  else:
11      print("Nilai A lebih kecil dari B")
```

```
Nilai A lebih besar dari B
```

Statement pass

Statemen **pass** tidak melakukan proses apa-apa. Statement ini dapat digunakan ketika pernyataan diperlukan secara sintaksis tetapi program tidak memerlukan Tindakan apapun.

Contoh pernyataan pass

```
1  x = 5
2  if (x<5):
3      x=3
4  else:
5      pass
6
```

Defining Function

Pernyataan **def** merupakan sebuah definisi fungsi. Pernyataan **def** harus diikuti dengan nama fungsi dan daftar parameter formal dalam kurung. Pernyataan-pernyataan yang membentuk badan fungsi dimulai pada baris berikutnya, dan harus diindentasi (penulisan baris berikutnya harus menjorok kedalam satu tab).

Penulisan Defining Function

Setelah tanda titik 2 dari **def** maka barus selanjutnya harus menjorok ke dalam 1 tab

```
def fib(n):    # write Fibonacci series up to n
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
```

Contoh Defining Function

```
1   def fib(n):    # write Fibonacci series up to n
2     """Print a Fibonacci series up to n."""
3     a, b = 0, 1
4     while a < n:
5       print(a, end=' ')
6       a, b = b, a+b
7
8   # Now call the function we just defined:
9   fib(2000)
10
```

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

Break & Continue

Pernyataan **break**, sama seperti dalam Bahasa C, yaitu digunakan untuk keluar dari looping for atau looping while. Sedangkan **continue** untuk melanjutkan ke iterasi berikutnya dari sebuah loop.

Contoh Break

```
1   for n in range(2, 10):
2     for x in range(2, n):
3       if n % x == 0:
4         print(n, 'equals', x, '*', n//x)
5         break
6       else:
7         # loop fell through without finding a factor
8         print(n, 'is a prime number')
```

```
3 is a prime number
4 equals 2 * 2
5 is a prime number
5 is a prime number
5 is a prime number
6 equals 2 * 3
7 is a prime number
7 is a prime number
7 is a prime number
7 is a prime number
7 is a prime number
8 equals 2 * 4
9 is a prime number
9 equals 3 * 3
```

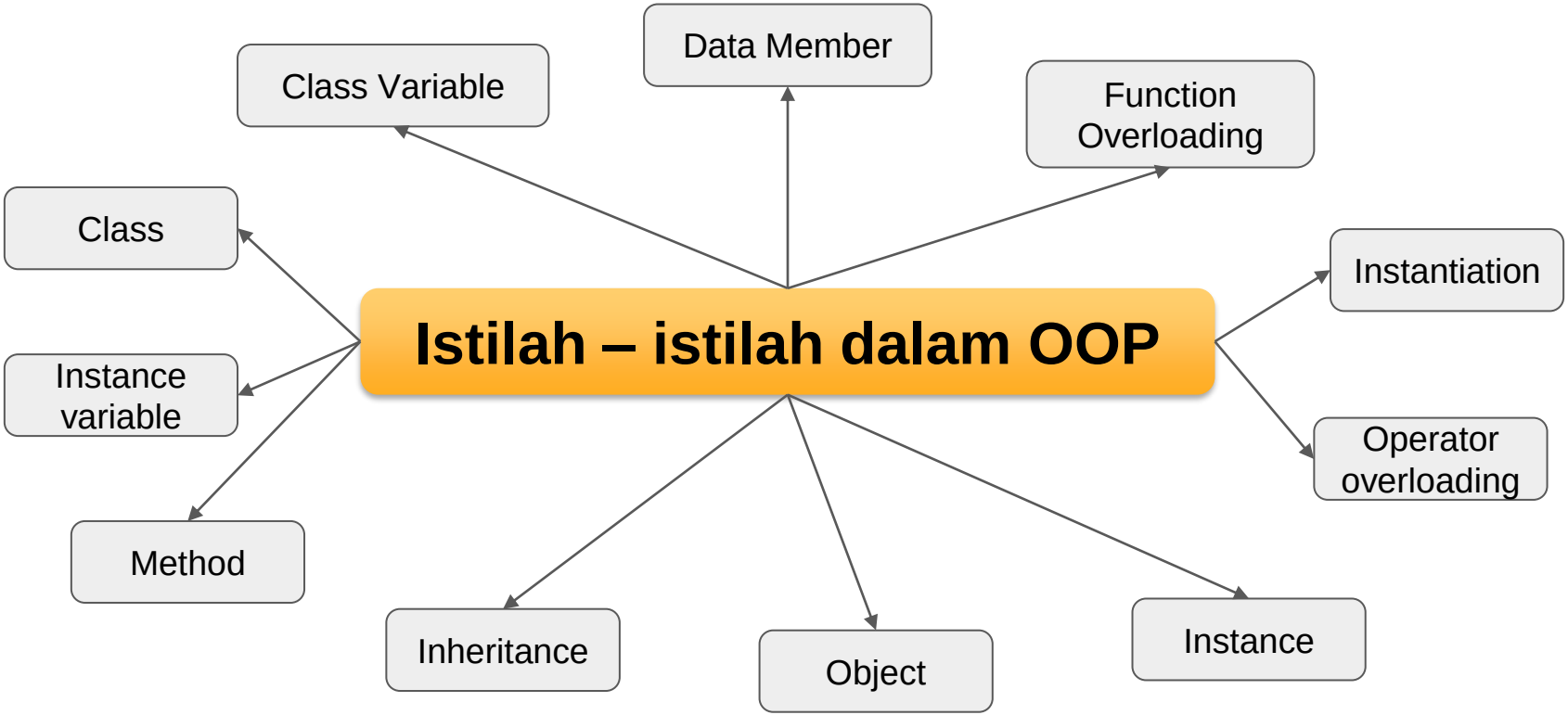
Contoh Continue

```python
for num in range(2, 10):
    if num % 2 == 0:
        print("Found an even number", num)
        continue
    print("Found an odd number", num)
```

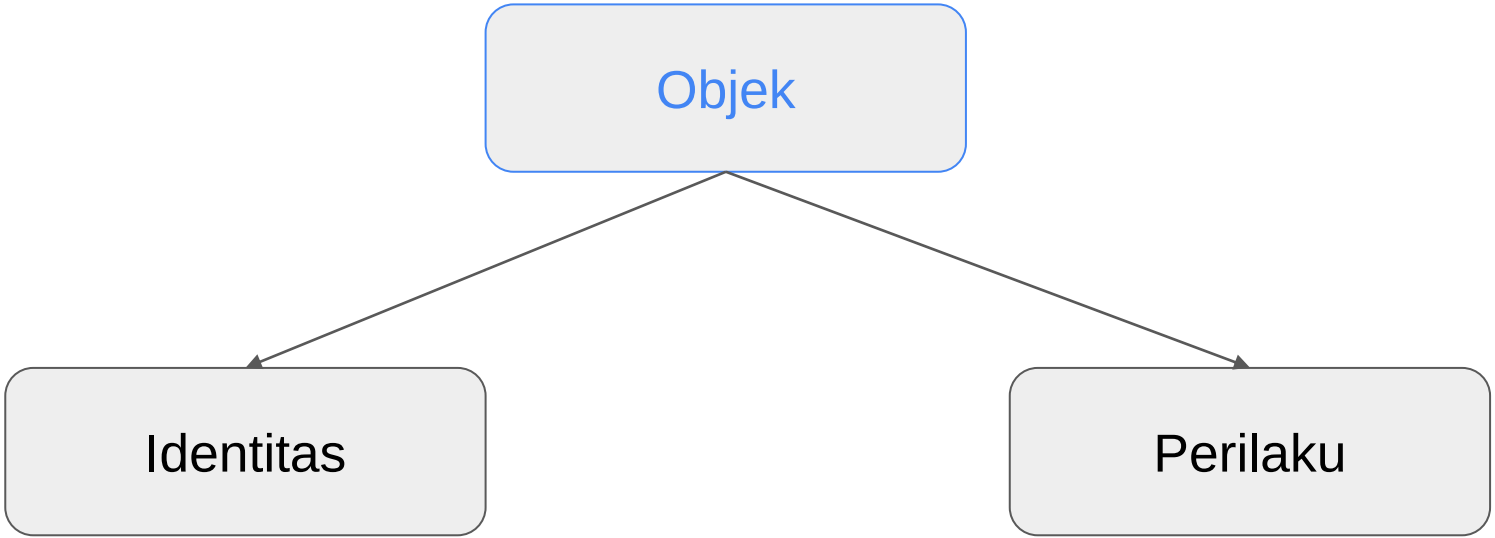
```
Found an even number 2
Found an odd number 3
Found an even number 4
Found an odd number 5
Found an even number 6
Found an odd number 7
Found an even number 8
Found an odd number 9
```

Object-Oriented Programming (OOP) - Intro

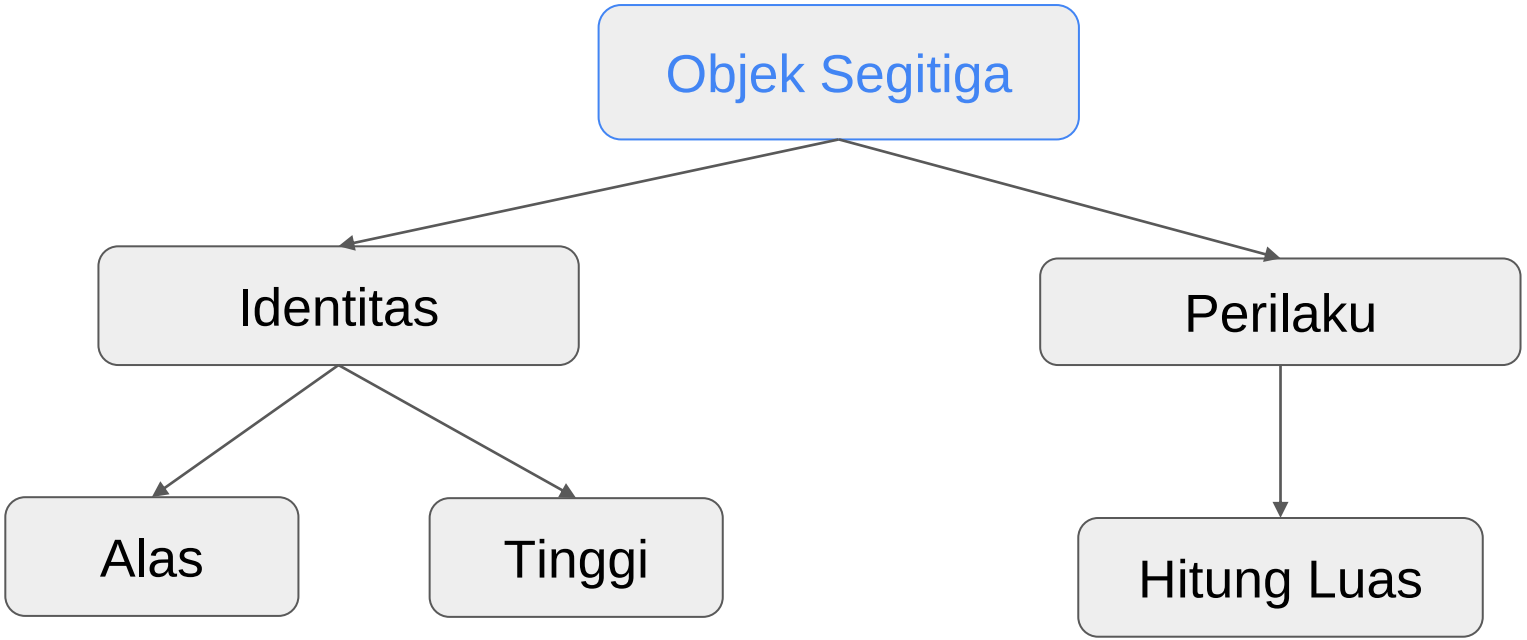
Pemrograman berorientasi objek atau dalam bahasa inggris disebut Object Oriented Programming (OOP) adalah paradigma atau teknik pemrograman di mana semua hal dalam program dimodelkan seperti objek dalam dunia nyata. Objek di dunia nyata memiliki ciri atau attribut dan juga aksi atau kelakuan (behaviour).


Istilah – istilah dalam OOP
Class Variable
Data Member
Function Overloading
Class
Instance variable
Method
Instantiation
Operator overloading
Inheritance
Object
Instance

Konsep Dasar OOP



Konsep Dasar OOP



Contoh program OOP

```python
class Segitiga:
    def __init__(self, alas, tinggi):
        self.alas = alas
        self.tinggi = tinggi

    def get_luas(self):
        return 0.5 * self.alas * self.tinggi

segitiga1 = Segitiga(5, 10)
segitiga2 = Segitiga(10, 10)

print('luas segitiga1:', segitiga1.get_luas())
print('luas segitiga2:', segitiga2.get_luas())
```

```
luas segitiga1: 25.0
luas segitiga2: 50.0
```

Session II

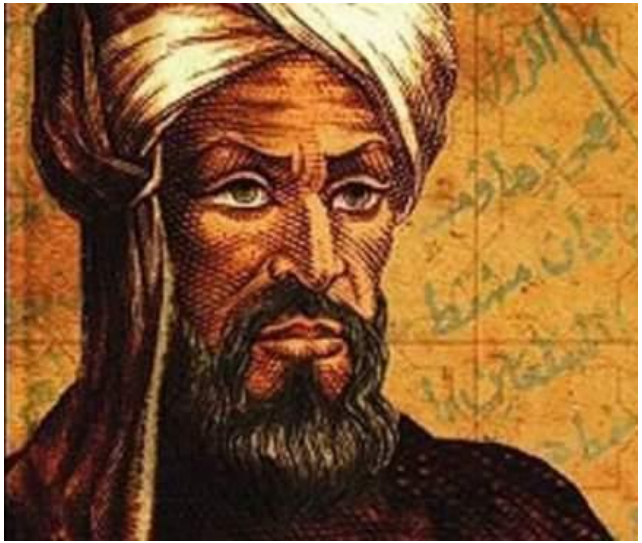
What is Algorithm?

What is Algorithm?

Algoritma adalah urutan langkah logis yang digunakan untuk menyelesaikan suatu masalah. Singkatnya, sebuah masalah harus diselesaikan dengan beberapa langkah yang logis.

Sejarah Algoritma

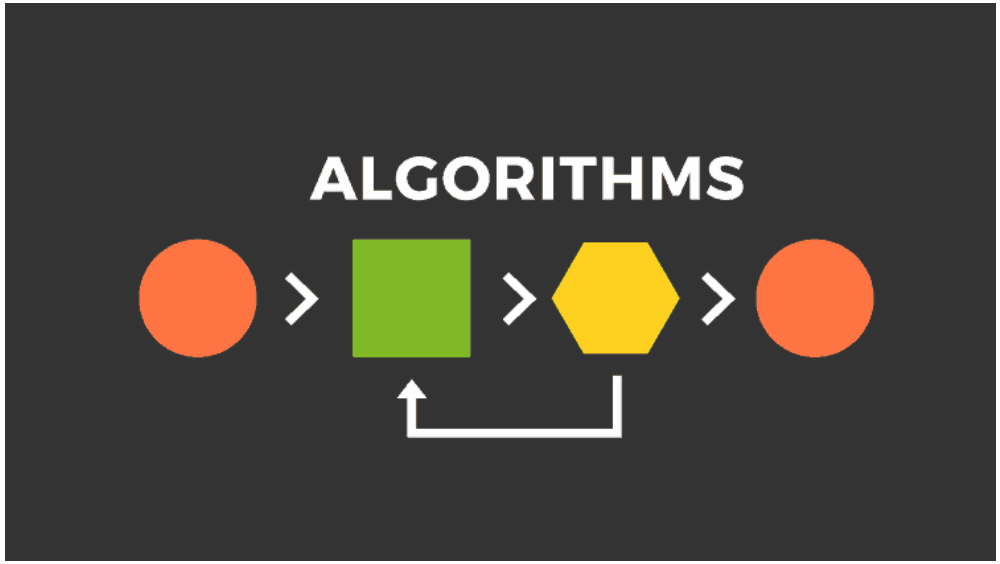
Istilah algoritma berasal dari nama seorang pengarang berkebangsaan Arab bernama Abu Ja'far Mohammad ibn Musa al Khowarizmi (tahun 790 – 840), yang sangat terkenal sebagai 'Bapak Aljabar'. Beliau juga adalah seorang astronom, ahli geografi, dan sarjana di House of Wisdom di Baghdad.



Ada 3 pendekatan dalam menyelesaikan algortima

Dasar Penyusunan Algoritma

- Finiteness
- Defitininess
- Masukan
- Keluaran
- Efektivitas

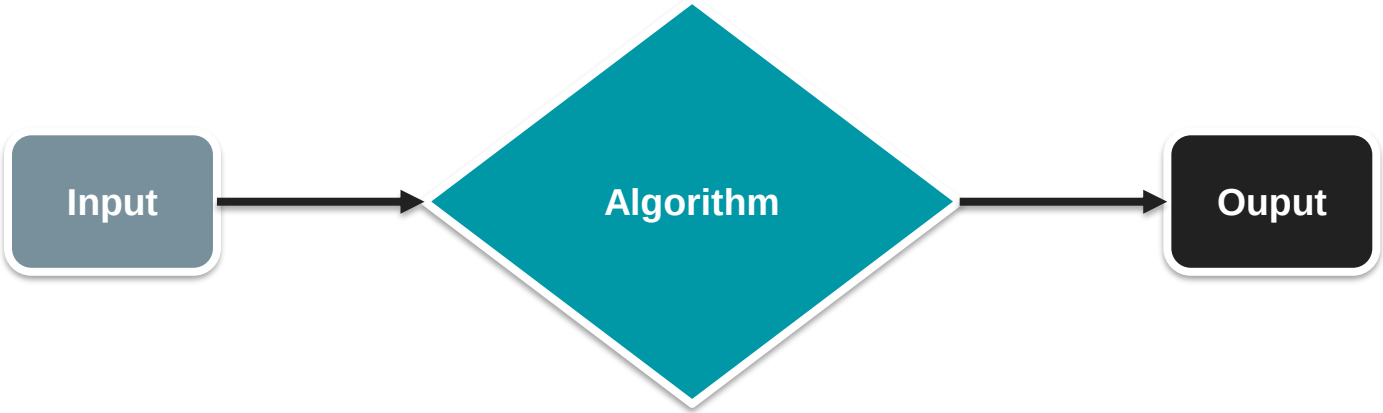


sumber gambar: google

Menurut Knuth (1973) dan juga Horowitz dkk. (1999)

Sebuah algoritma dikatakan benar, untuk berbagai ragam masukan, jika algoritma berakhir dengan keluaran yang benar. Pada keadaan seperti ini, algoritma menyelesaikan masalah komputasi yang diberikan.

Cormen, dkk. (1994)

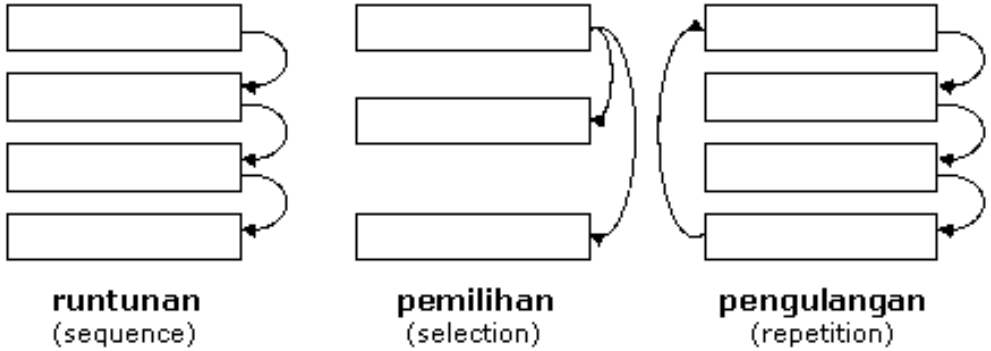

Input
Algorithm
Ouput

Algoritma dalam kehidupan sehari-hari

- Taruhlah makanan dalam wadah yang aman untuk *microwave*.
- Tutuplah pintu *microwave* dengan rapat.
- Tancapkan steker ke stop kontak.
- Putarlah knop ke posisi 5 menit.
- Tunggu sampai lampu mati dan ada bunyi 'ting'.
- Lepaskan steker dari stop kontak.
- Bukalah Pintu pemasak *microwave* dan keluarkan wadah yang berisi makanan tersebut.

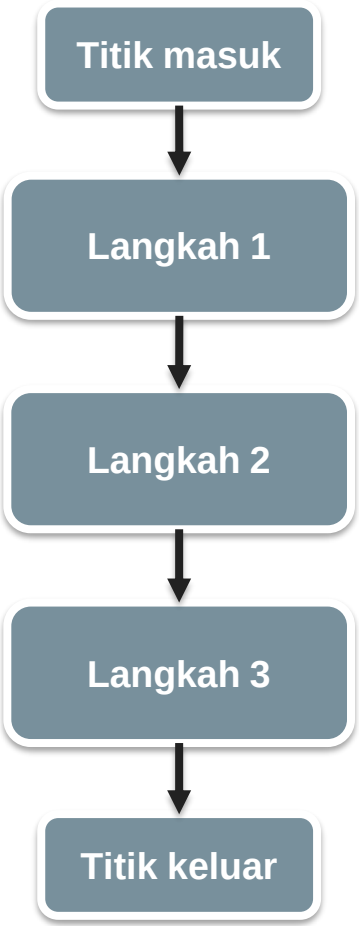
Jenis Struktur Dasar Algoritma

- Sekuensial
- Seleksi
- Pengulangan

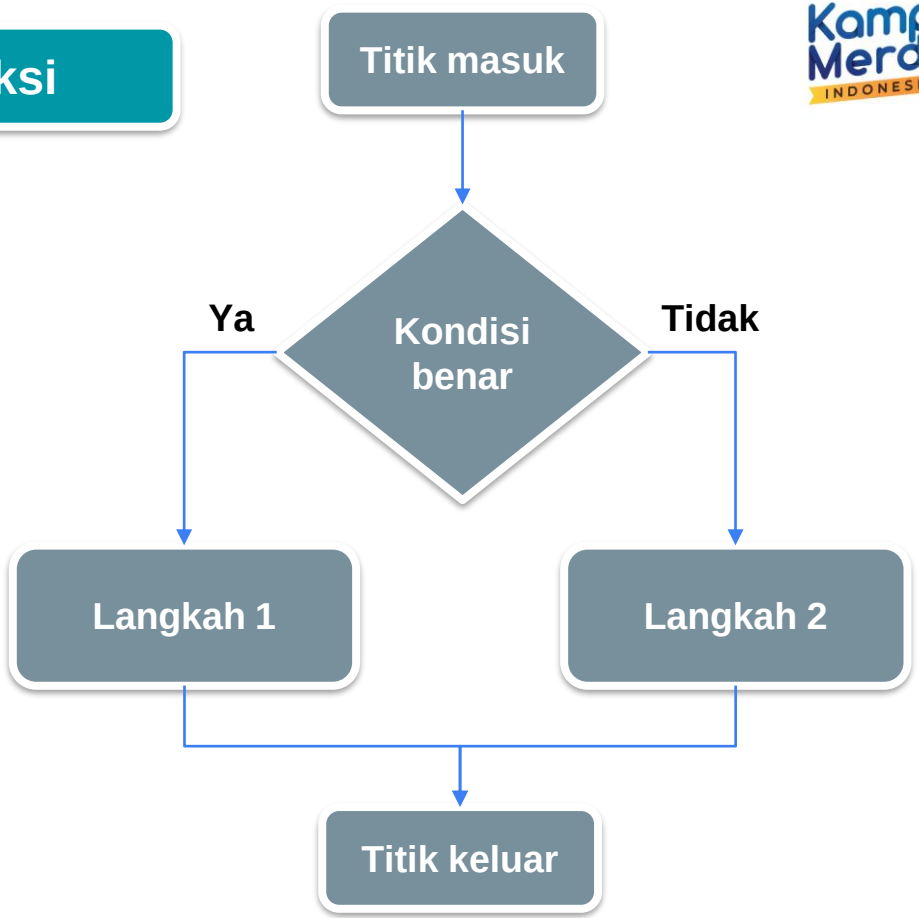


Sumber Gambar: Google

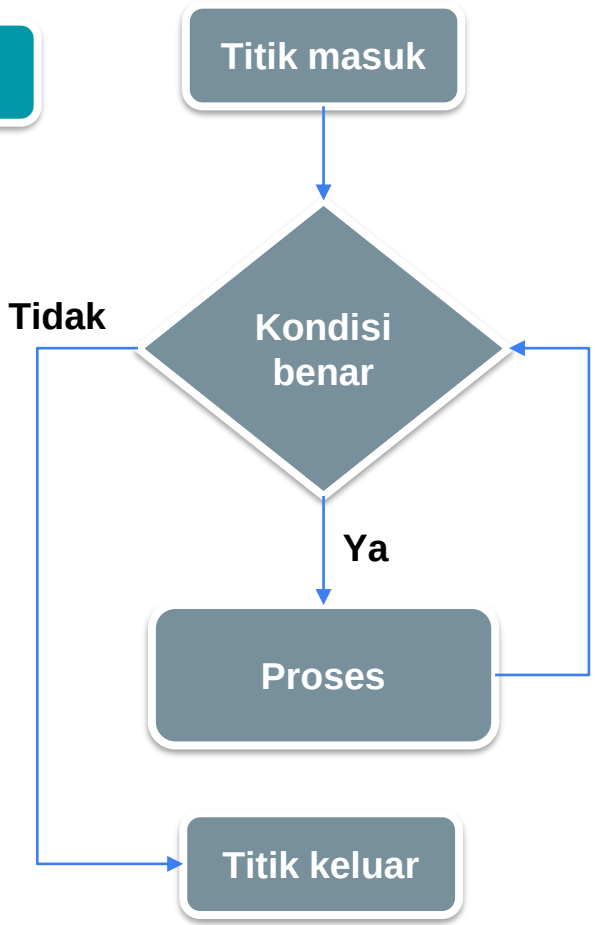
Struktur Sekuensial



Struktur Seleksi



Struktur Perulangan



Activity

Algoritma Terapan

- Tree Traversal Algorithm
- Searching Algorithm
- Sorting Algorithm
- Insertion Algorithm
- Deletion Algorithm
- Merging Algorithm

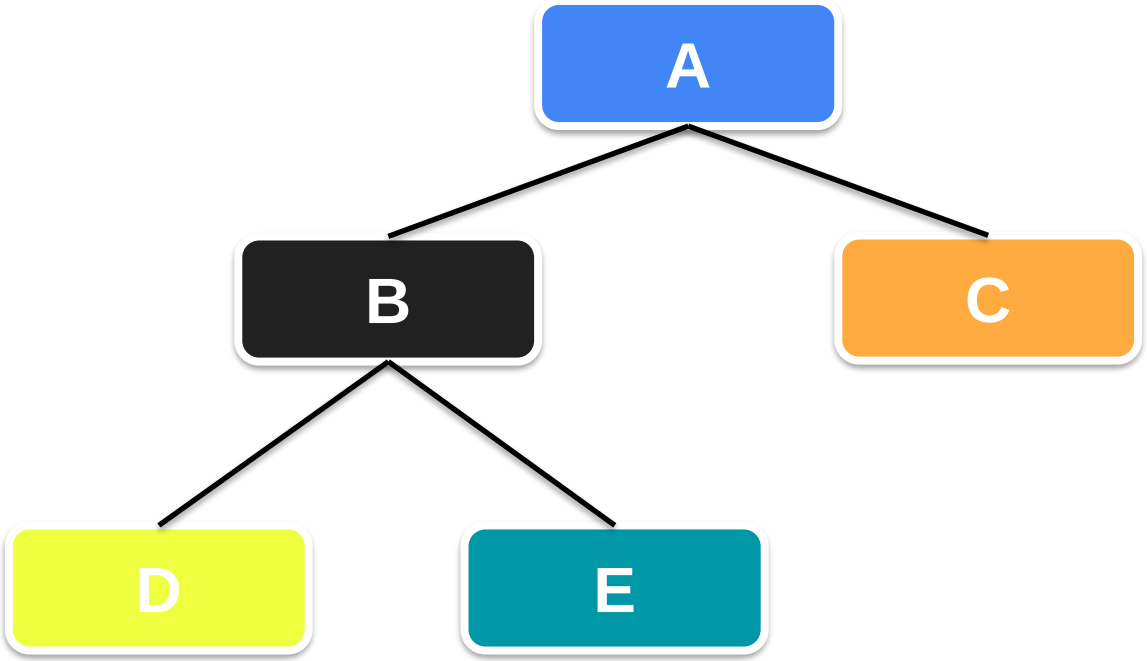
Tree Traversal Algorithm

Tree Traversal Algorithm

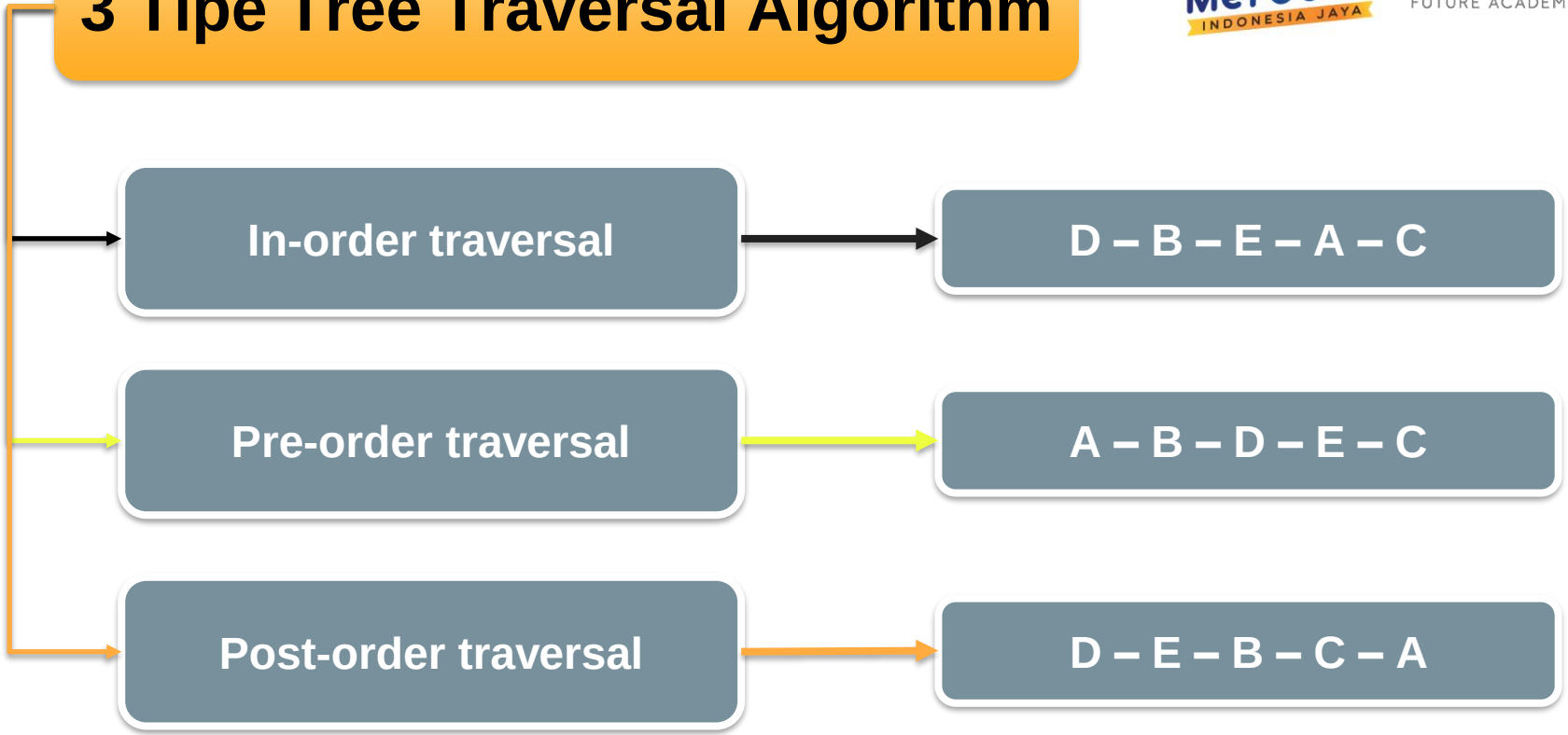
Trees di python adalah struktur data non-linear. Memiliki ciri-ciri akar dan simpul. *Tree* adalah kumpulan *element* yang saling terhubung secara hirarki (*one to many*). *Element* pada *tree* disebut *node*.

Contohnya adalah binary tree

Tree Traversal Algorithm



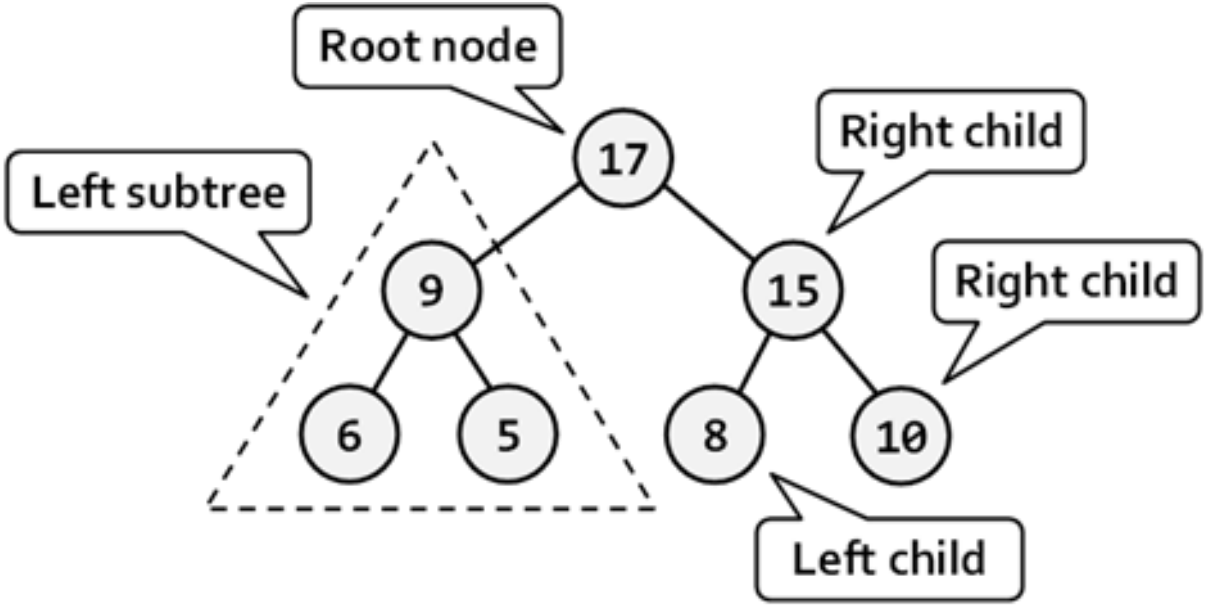
3 Tipe Tree Traversal Algorithm



Implementasi Algoritma Tree Traversal

- Penyimpanan Data Statik
- Penyimpanan Data Dinamik
- Pencarian Data

Binary Tree – Implementation



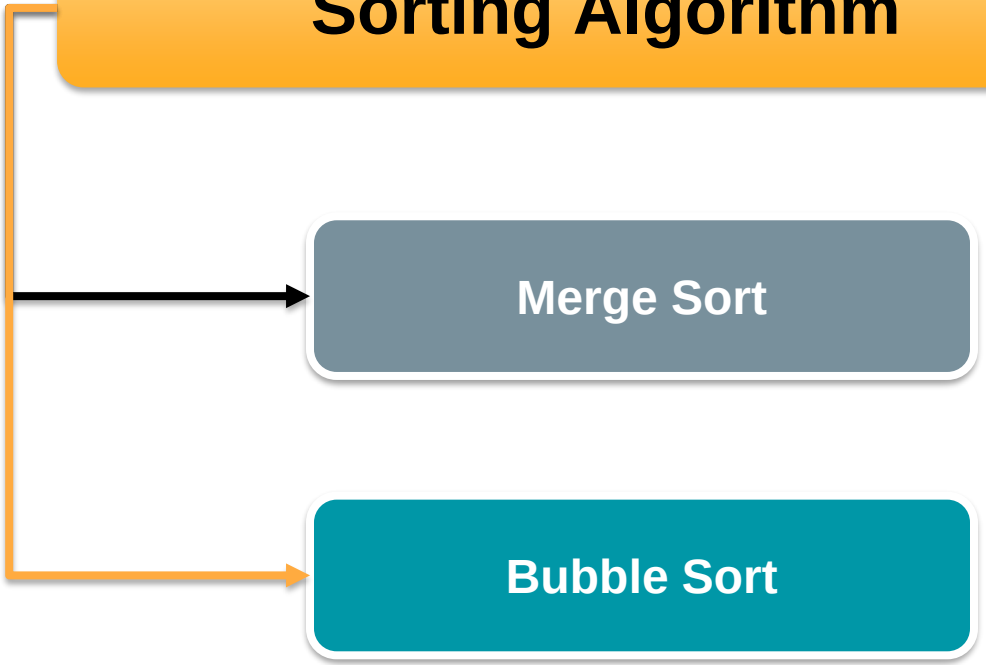
Sorting Algorithm

Sorting Algorithm

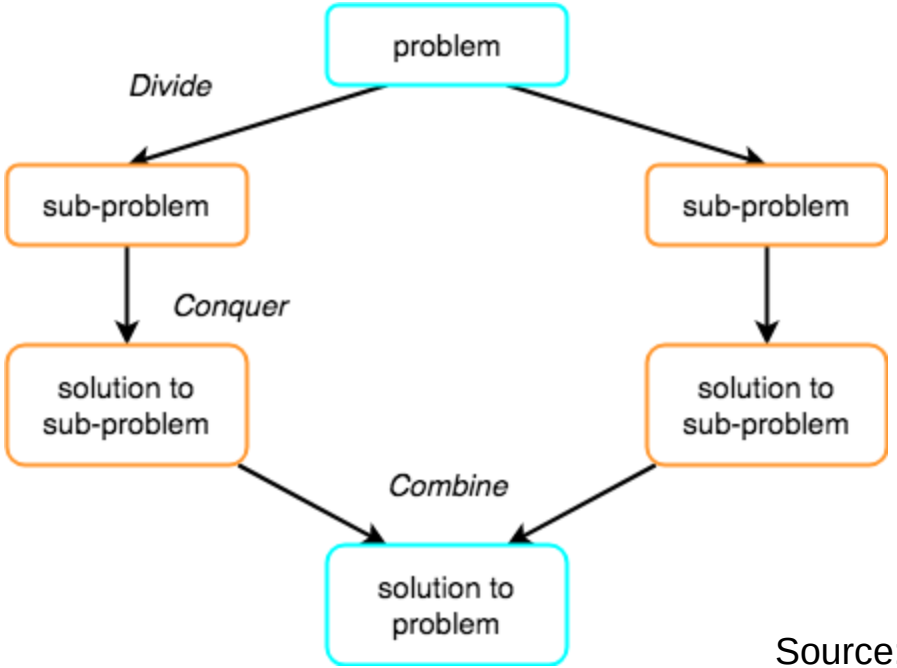
Algoritma pengurutan digunakan untuk mengurutkan data dengan urutan tertentu.

Contoh: Merge Sort dan Bubble Sort

Sorting Algorithm

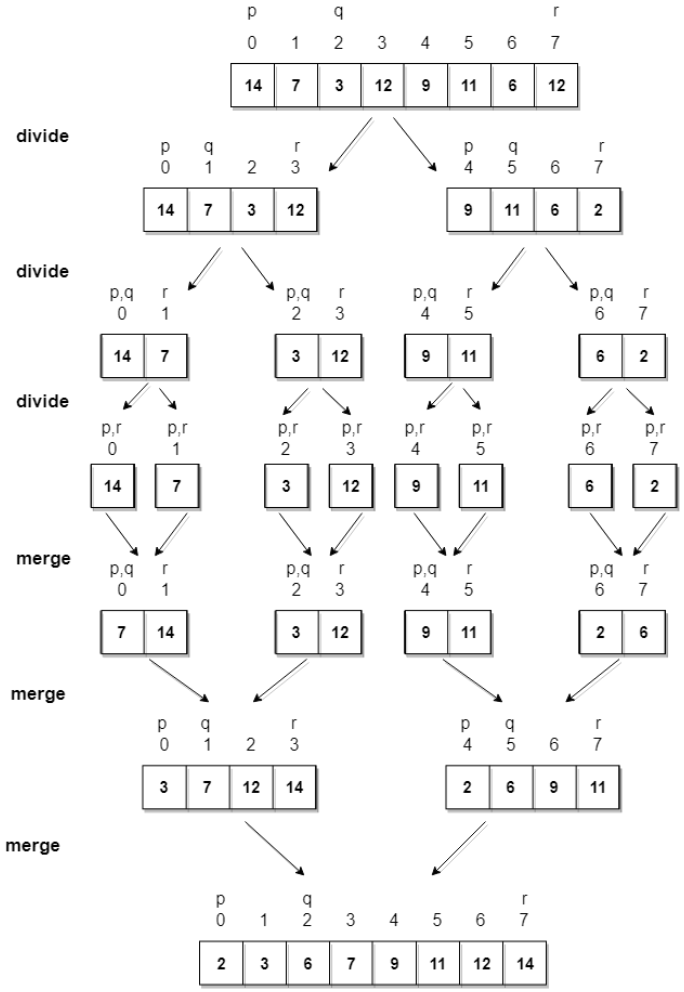


Divide and Conquer in Merge Sort



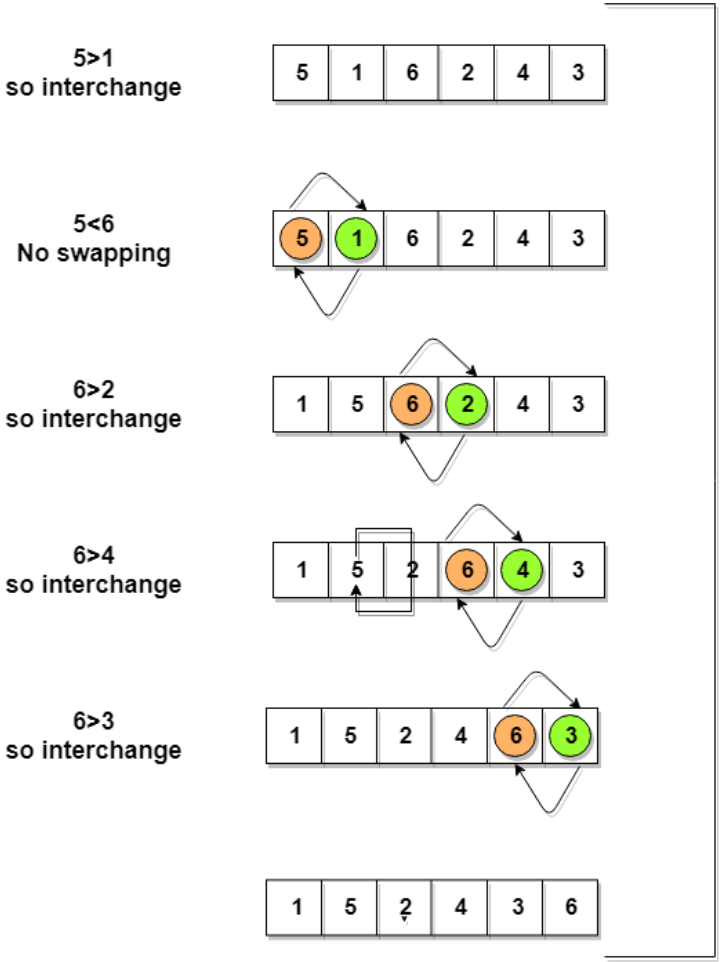
Source: studytonight.com

How Merge Sort Works?



Source: studytonight.com

How Bubble Sort Works?



This is first insertion

similarly, after all the iterations, the array gets sorted

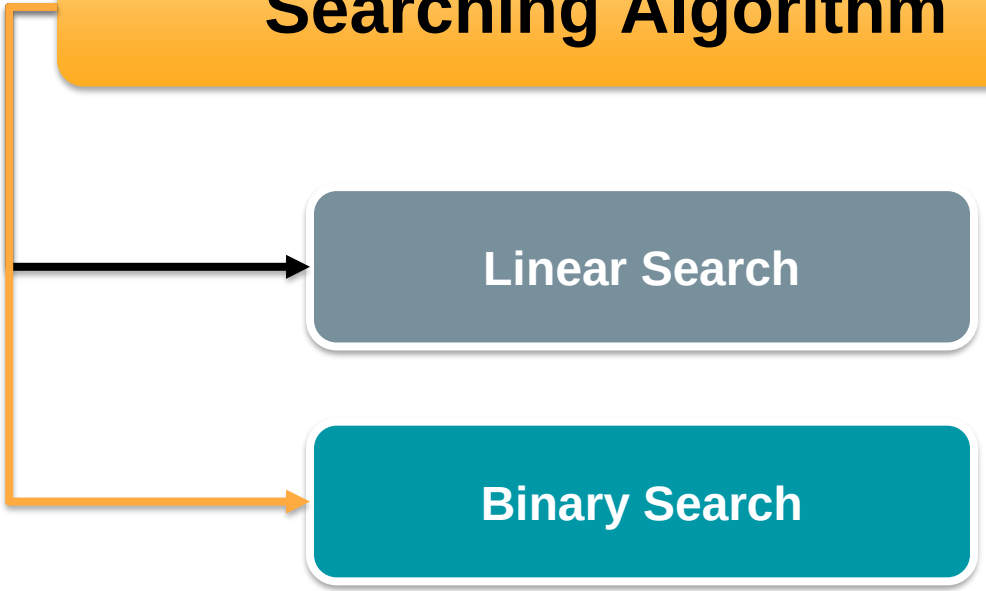
Source: studytonight.com

Searching Algorithm

Searching Algorithm

Algoritma pencarian digunakan untuk mencari beberapa elemen yang ada dalam kumpulan data. Ada banyak jenis algoritma pencarian seperti “Linear Search”, “Binary Search”, “Exponential Search”, “Interpolation Search”, dan lain sebagainya. Tapi pada pertemuan ini kita hanya membahas “Linier Search” dan “Binary Search” saja.

Searching Algorithm



Contoh Linier Search

1

```
1  def lin_search(ourlist, key):
2
3      for index in range(0, len(ourlist)):
4          if (ourlist[index] == key):
5              print("Nilai ditemukan")
6              return  index
7      else:
8          return "Nilai tidak ditemukan"
9
10 ourlist = [15, 1, 9, 3]
11
12 lin_search(ourlist, 2)
13
14
15
```

List Nilai

Nilai dicari

```
'Nilai tidak ditemukan'
```

2

```
1  def lin_search(ourlist, key):
2
3      for index in range(0, len(ourlist)):
4          if (ourlist[index] == key):
5              print("Nilai ditemukan")
6              return  index
7      else:
8          return "Nilai tidak ditemukan"
9
10 ourlist = [15, 1, 9, 3]
11
12 lin_search(ourlist, 9)
13
14
15
```

List Nilai

Nilai dicari

```
Nilai ditemukan
2
```

Contoh Binary Search

```python
def bin_search(ourlist, key):
    left = 0 # I assign left position to zero
    right = len(ourlist)-1 # I assign right position by defining the length of ourlist minus one
    matched = False
    while(left<=right and not matched): # the loop will continue untill the left element is less
                                        # or equal to the right element and the matched is True
        mid = (left+right)//2 # I find the position of the middle element
        if ourlist[mid] == key: # if the middle element correponds to the key element
            matched = True
        else: #otherwise
            if key < ourlist[mid]: # if key element is less than the middle element
                right = mid - 1 #I assign the position of the right element as mid - 1
            else: #otherwise
                left = mid + 1 #left position will become the middle position plus 1
    return matched

print(bin_search([1, 3, 9, 15], 17))
print(bin_search([1, 3, 9, 15], 3))
```

```
False
True
```

[1, 3, 9, dan 15] adalah list data yang terdaftar

[17 dan 3] adalah nilai yang ingin kita cari

Merangkum Materi

Quiz

1. Apa itu data struktur?
2. Apa itu OOP?
3. Apa itu algoritma?
4. Apa perbedaan algoritma dan program?
5. Sebutkan 3 jenis struktur dasar algoritma!
6. Bagaimana cara kerja *Merge Sort* pada algoritma *Sorting?*

Refleksi & Diskusi

Refrensi buku:

1. Algoritma & Pemrograman menggunakan C & C++ karangan Abdul Kadir

THANK YOU