

Module 1

Introduction to everything

Section

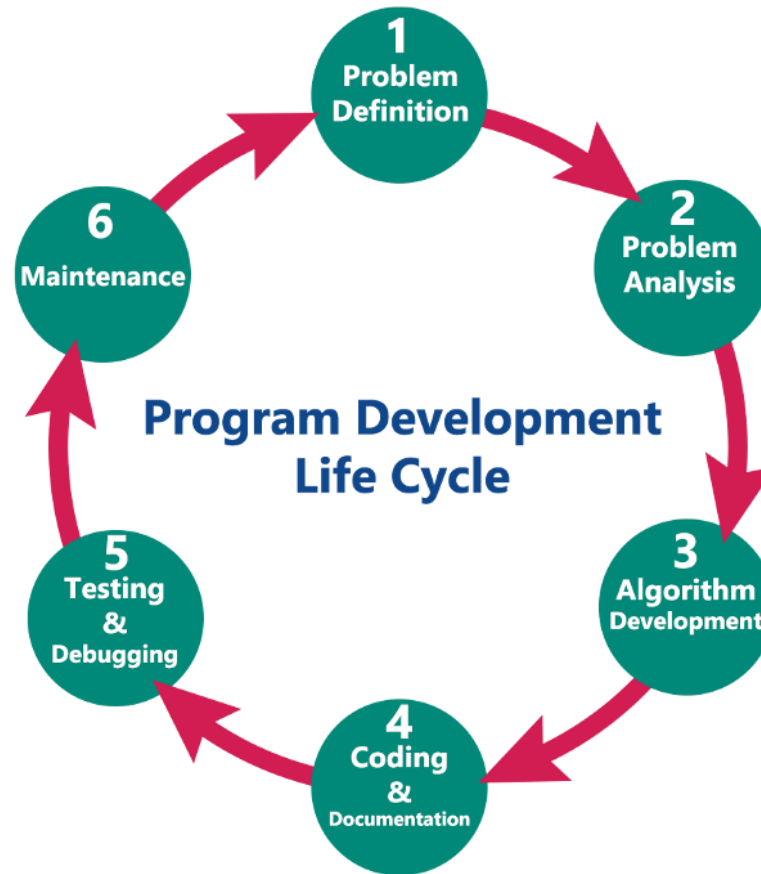
Algoritma, Flowchart, dan
Pengkondisian Lanjutan

Algoritma, Flowchart, dan Pengkondisian Lanjutan



Session I

**Algoritma, Flowchart, dan
Pemrograman (Python Lanjutan)**



Sumber: <https://www.dictio.id/t/apa-saja-langkah-langkah-yang-dilakukan-dalam-pemrograman/12667/2>

Dasar Penyusunan Algoritma

Finiteness

Definiteness

Masukan

Keluaran

Efektivitas

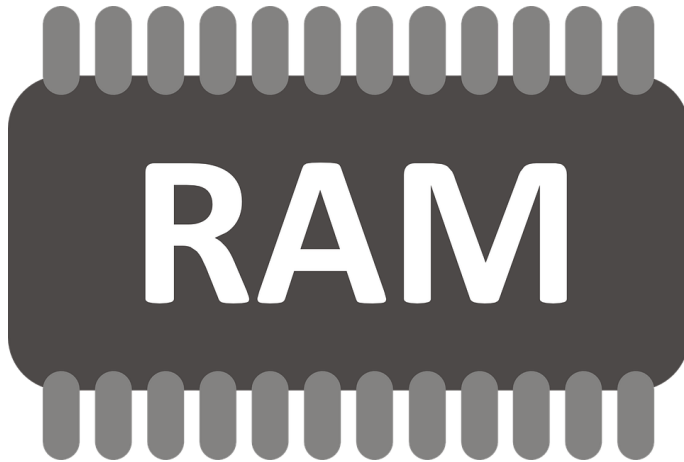
Input

Procedure

Ouput

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

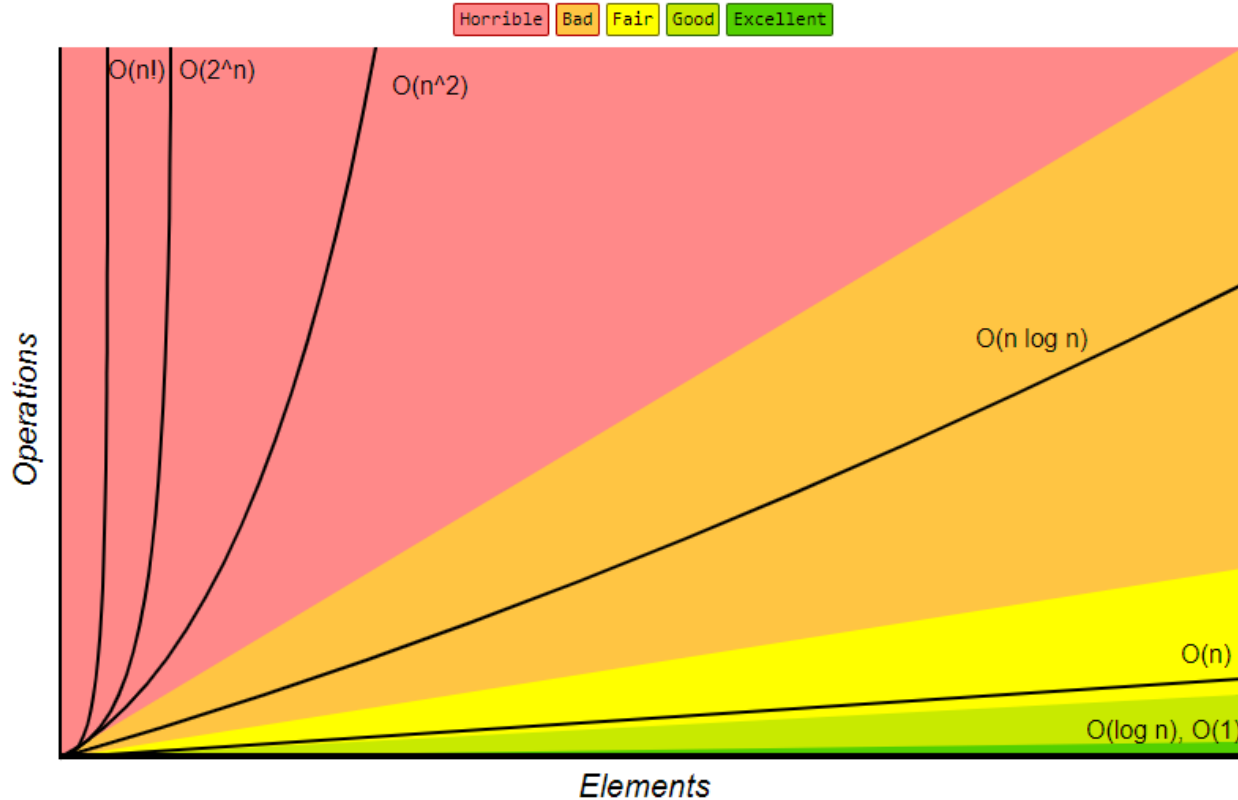
Bagaimana Algoritma yang efisien menurut kamu?



Data type	Description
<code>bool_</code>	Boolean (True or False) stored as a byte
<code>int_</code>	Default integer type (same as C <code>long</code> ; normally either <code>int64</code> or <code>int32</code>)
<code>intc</code>	Identical to C <code>int</code> (normally <code>int32</code> or <code>int64</code>)
<code>intp</code>	Integer used for indexing (same as C <code>ssize_t</code> ; normally either <code>int32</code> or <code>int64</code>)
<code>int8</code>	Byte (−128 to 127)
<code>int16</code>	Integer (−32768 to 32767)
<code>int32</code>	Integer (−2147483648 to 2147483647)
<code>int64</code>	Integer (−9223372036854775808 to 9223372036854775807)
<code>uint8</code>	Unsigned integer (0 to 255)
<code>uint16</code>	Unsigned integer (0 to 65535)
<code>uint32</code>	Unsigned integer (0 to 4294967295)
<code>uint64</code>	Unsigned integer (0 to 18446744073709551615)
<code>float_</code>	Shorthand for <code>float64</code>
<code>float16</code>	Half-precision float: sign bit, 5 bits exponent, 10 bits mantissa
<code>float32</code>	Single-precision float: sign bit, 8 bits exponent, 23 bits mantissa
<code>float64</code>	Double-precision float: sign bit, 11 bits exponent, 52 bits mantissa
<code>complex_</code>	Shorthand for <code>complex128</code>
<code>complex64</code>	Complex number, represented by two 32-bit floats
<code>complex128</code>	Complex number, represented by two 64-bit floats

Contoh alokasi penggunaan
memori untuk data type
pada library Numpy

Big-O Complexity Chart



$O(N!)$	Factorial
$O(2^N)$	Exponential
$O(N^3)$	Cubic
$O(N^2)$	Quadratic
$O(N \log N)$	$N \times \log N$
$O(N)$	Linear
$O(\log N)$	Logarithmic
$O(1)$	Constant

Sumber: <https://www.bigocheatsheet.com/> dan <https://towardsdatascience.com>

Common Data Structure Operations

Data Structure	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
<u>Array</u>	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<u>Stack</u>	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
<u>Queue</u>	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
<u>Singly-Linked List</u>	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
<u>Doubly-Linked List</u>	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$	$O(n)$	$O(1)$	$O(1)$	$O(n)$
<u>Skip List</u>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n \log(n))$
<u>Hash Table</u>	N/A	$O(1)$	$O(1)$	$O(1)$	N/A	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<u>Binary Search Tree</u>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<u>Cartesian Tree</u>	N/A	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	N/A	$O(n)$	$O(n)$	$O(n)$	$O(n)$
<u>B-Tree</u>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
<u>Red-Black Tree</u>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
<u>Splay Tree</u>	N/A	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	N/A	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
<u>AVL Tree</u>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$
<u>KD Tree</u>	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(\log(n))$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$

Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
<u>Quicksort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(\log(n))$
<u>Mergesort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
<u>Timsort</u>	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
<u>Heapsort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(1)$
<u>Bubble Sort</u>	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Insertion Sort</u>	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Selection Sort</u>	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
<u>Tree Sort</u>	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(n)$
<u>Shell Sort</u>	$\Omega(n \log(n))$	$\Theta(n(\log(n))^2)$	$O(n(\log(n))^2)$	$O(1)$
<u>Bucket Sort</u>	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$	$O(n)$
<u>Radix Sort</u>	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$	$O(n+k)$
<u>Counting Sort</u>	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$	$O(k)$
<u>Cubesort</u>	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$

Sumber: <https://www.bigocheatsheet.com/>



Session II

**Menyusun Algoritma, Flowchart
dan Pembuatan Program
(Python Lanjutan)**

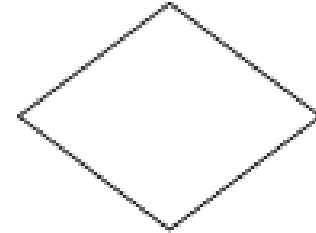
Simbol standar Flowchart



Terminator



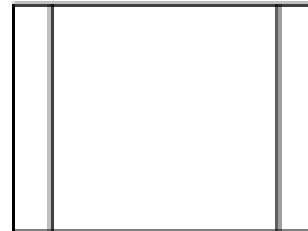
Proses



Keputusan



Input/Output



Proses terdefinisi



Konektor



Simbol Arus

SIMBOL TERMINAL



Menunjukkan permulaan (start) atau akhir (stop) dari suatu proses

SIMBOL INPUT/OUTPUT



Menunjukkan proses input/output yang terjadi

SIMBOL PROSES



Digunakan untuk menunjukkan kegiatan yang dilakukan oleh computer

SIMBOL KEPUTUSAN



Digunakan untuk memilih keputusan berdasarkan kondisi yang ada

SIMBOL ARUS



Digunakan untuk menghubungkan satu symbol dengan symbol lainnya. Simbol ini berfungsi untuk menunjukan garis alir dari proses.

Belajar menyusun algoritma dari beberapa kasus berikut

Konversi bilangan Desimal (basis 10) ke Hexadecimal (basis 16), Octal (basis 8), dan Biner (basis 2)

Menghitung luas dan keliling bangun datar

Menghitung Volume dan Luas Permukaan Bangun Ruang

Algoritma Menghitung Luas & Keliling Persegi

Sisi



Apa yang kamu ketahui tentang persegi ini?

Rumus Menghitung Luas & Keliling Persegi

Rumus luas dan keliling Persegi
adalah

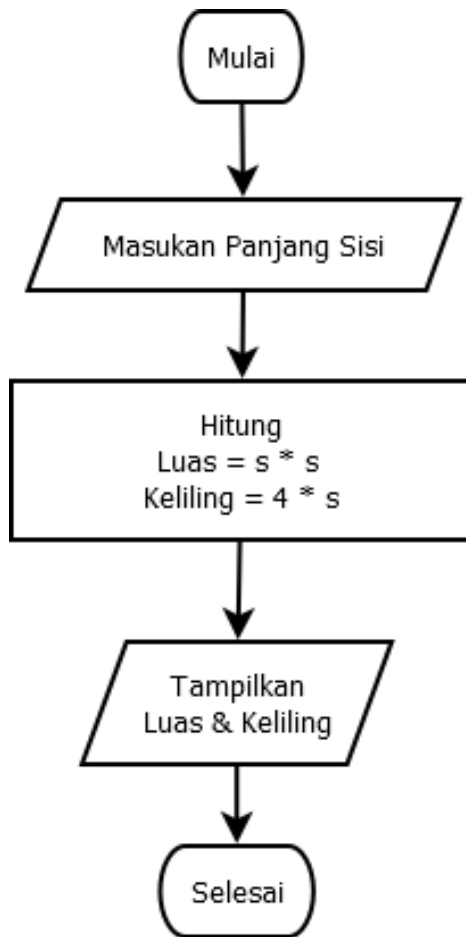
$$\text{Luas} = s^2$$

$$\text{Keliling} = 4 \times s$$

Algoritma Menghitung Luas & Keliling Persegi

- Mulai
- Masukkan sisi (s)
- Hitung luas = s^2
- Hitung keliling = $4 * s$
- Tampilkan Luas dan Keliling
- Selesai

Flowchart Menghitung Luas & Keliling Persegi



Code Menghitung Luas & Keliling Persegi

```
1  print("MENGHITUNG LUAS & KELILING PERSEGI")
2
3  s = float(input("\nMasukan Panjang Sisi dalam cm: "))
4
5  luas = s**2
6  keliling = 4 * s
7
8  print("\nLuas Persegi \t\t:",luas, "cm^2")
9  print("Keliling Persegi\t:",keliling, "cm")
```

Rumus Menghitung Volume dan Luas Permukaan Bola

Rumus Volume dan Luas Permukaan Bola

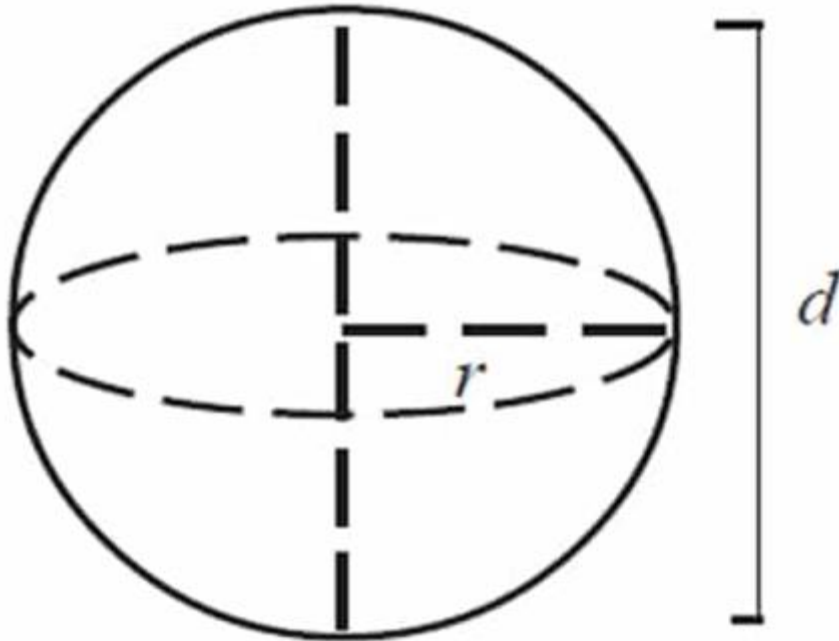
$$Volume = \frac{4}{3} \times \pi \times r^3$$

$$Luas Permukaan = 4 \times \pi \times r^2$$

Algoritma Menghitung Volume & Luas Permukaan Bola

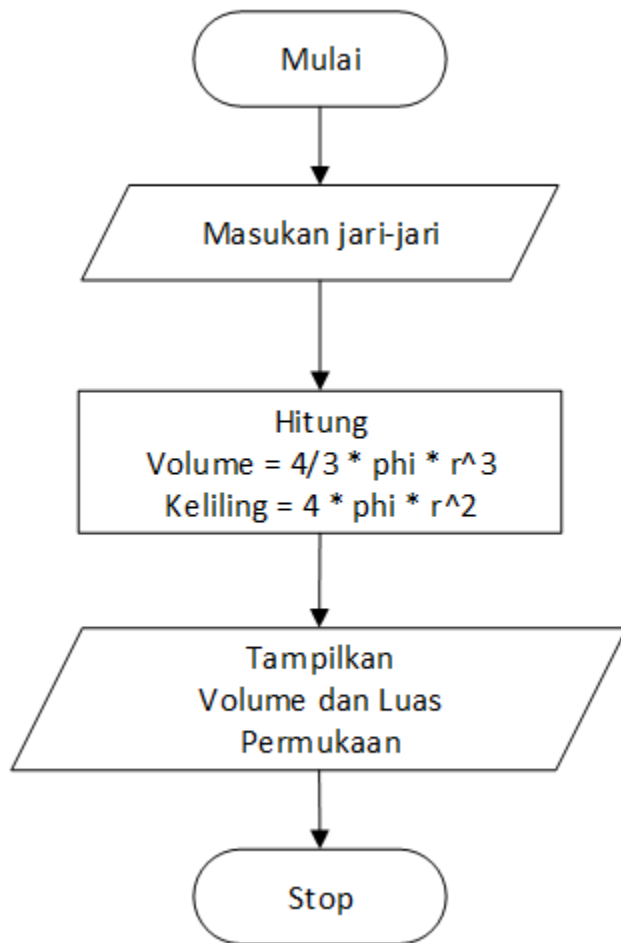
- Mulai
- Masukan jari-jari
- Hitung $\text{volume} = \frac{4}{3} * \pi * r^3$
- Hitung $\text{luas permukaan} = 4 * \pi * r^2$
- Tampilkan volume dan luas permukaan bola
- Selesai

Algoritma Menghitung Volume & Luas Permukaan Bola



Apa yang kamu ketahui tentang bangun ruang ini?

Flowchart Menghitung Volume & Luas Permukaan Bola



Code Menghitung Volume & Luas Permukaan Bola

```
1  print("MENGHITUNG VOLUME & LUAS PERMUKAAN BOLA")
2
3  jari_jari = float(input("\nMasukan jari-jari dalam cm: "))
4
5  phi = 3.14
6  volume = 4/3 * phi * jari_jari**3
7  luas_permukaan = 4 * phi * jari_jari**2
8
9  print("\nVolume Bola adalah \t\t:", volume, "cm^3")
10 print("Luas Permukaan Bola adalah\t:", luas_permukaan, "cm^2")
```



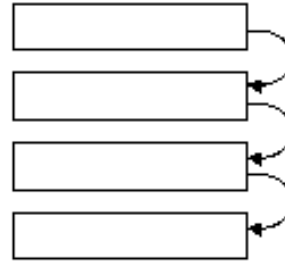

Sumber: Google

Jenis Struktur Dasar Algoritma

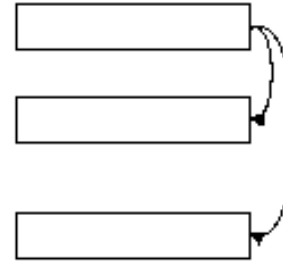
Sekuensial

Seleksi

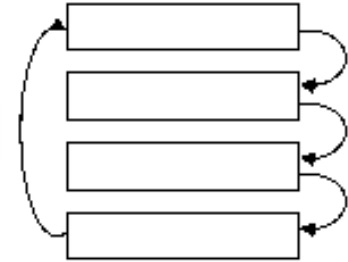
Pengulangan



runtunan
(sequence)



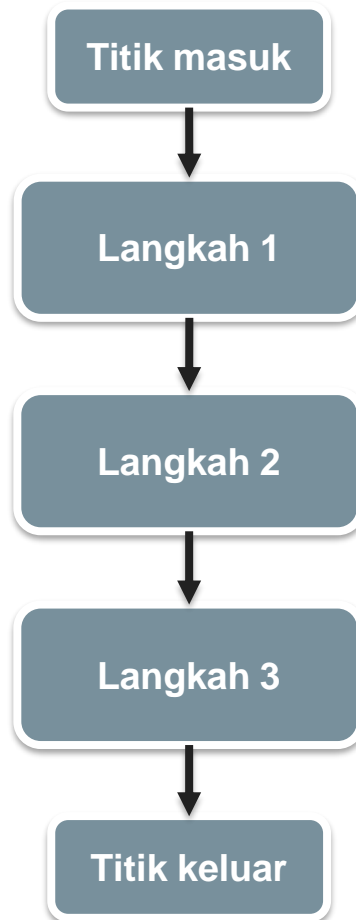
pemilihan
(selection)



pengulangan
(repetition)

Sumber Gambar: Google

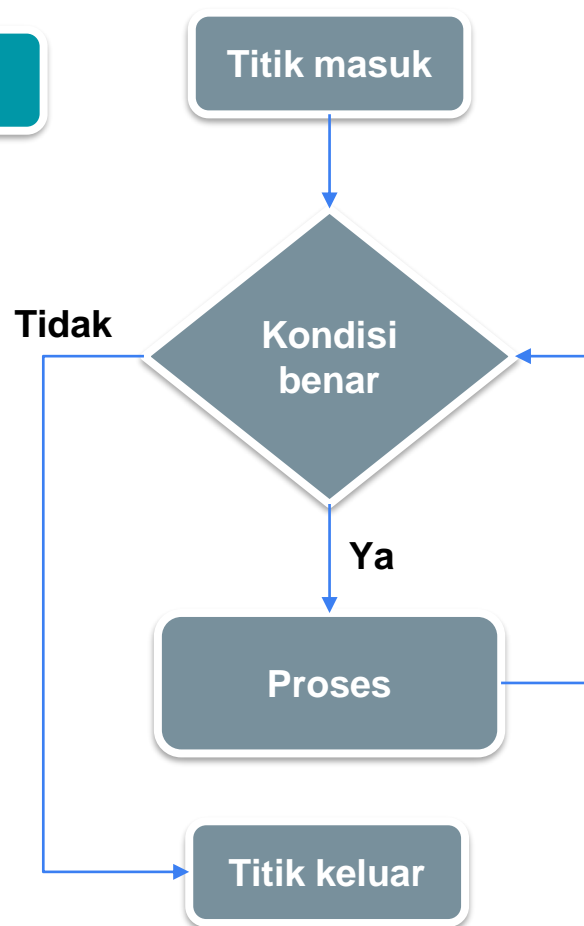
Struktur Sekuensial



Struktur Seleksi



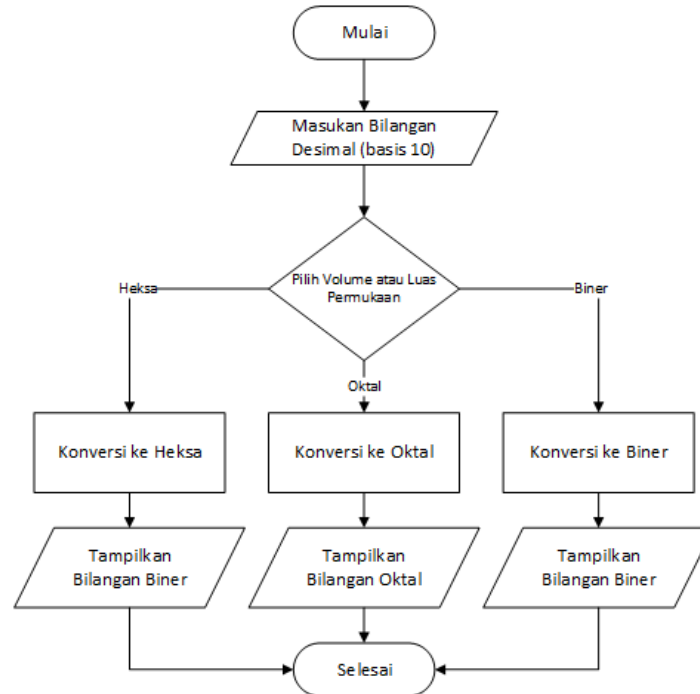
Struktur Perulangan



Algoritma Percabangan Mengkonversi Bilangan

- Mulai
- Masukkan bilangan desimal yang ingin di konversi
- Pilih: a. Konversi ke Hexa, b. Konversi ke Oktal, dan c. konversi ke Biner
- Tampilkan hasil Konversi
- Selesai

Algoritma Percabangan Mengkonversi Bilangan



Algoritma Percabangan Menghitung Volume & Luas Permukaan Bola

- Mulai
- Masukan jari-jari
- Pilih: 1. Hitung Volume, 2. Hitung Luas Permukaan

Jika Volume

Hitung volume = $\frac{4}{3} * \pi * r^3$

Tampilkan Volume Bola

Selesai

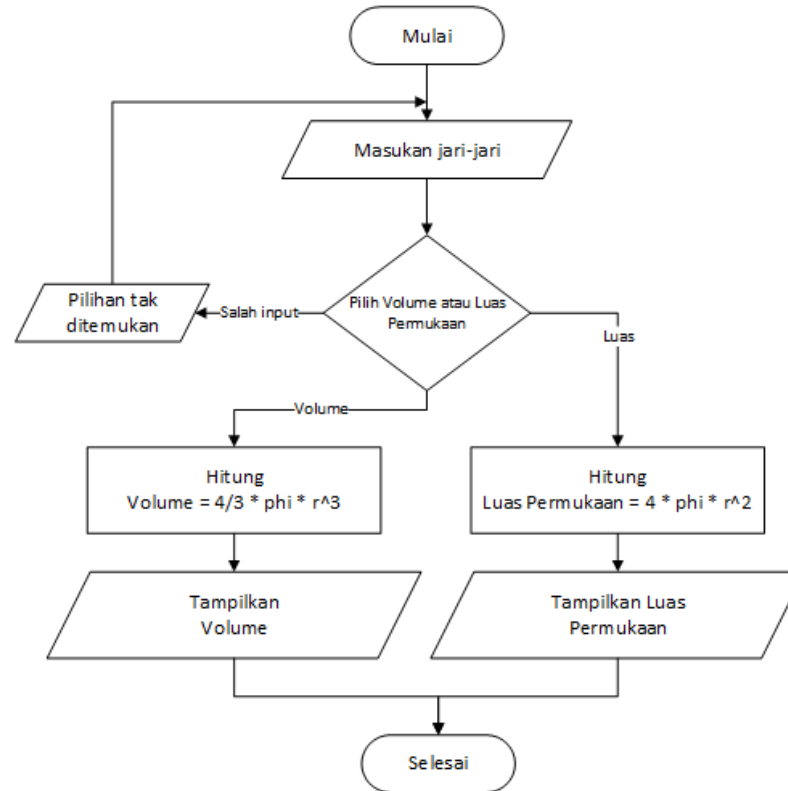
Jika Luas Permukaan

Hitung luas permukaan = $4 * \pi * r^2$

Tampilkan luas permukaan bola

Selesai

Flowchart Percabangan Menghitung Volume & Luas Permukaan Bola



Algoritma Perulangan dan Percabangan Menghitung Volume & Luas Permukaan Bola

- Mulai
- Masukan pilihan untuk melanjutkan atau keluar dari looping
- Pilih: 1. Lanjut Menghitung, 2. Exit Looping

Jika Menghitung

Masukkan jari-jari

Pilih: 1. Hitung Volume, 2 Hitung Luas Permukaan

Jika Volume

Hitung $\text{volume} = \frac{4}{3} * \pi * r^3$

Tampilkan Volume Bola

Kembali ke Pilihan Melanjutkan Hitungan atau Exit

Jika Luas Permukaan

Hitung $\text{luas permukaan} = 4 * \pi * r^2$

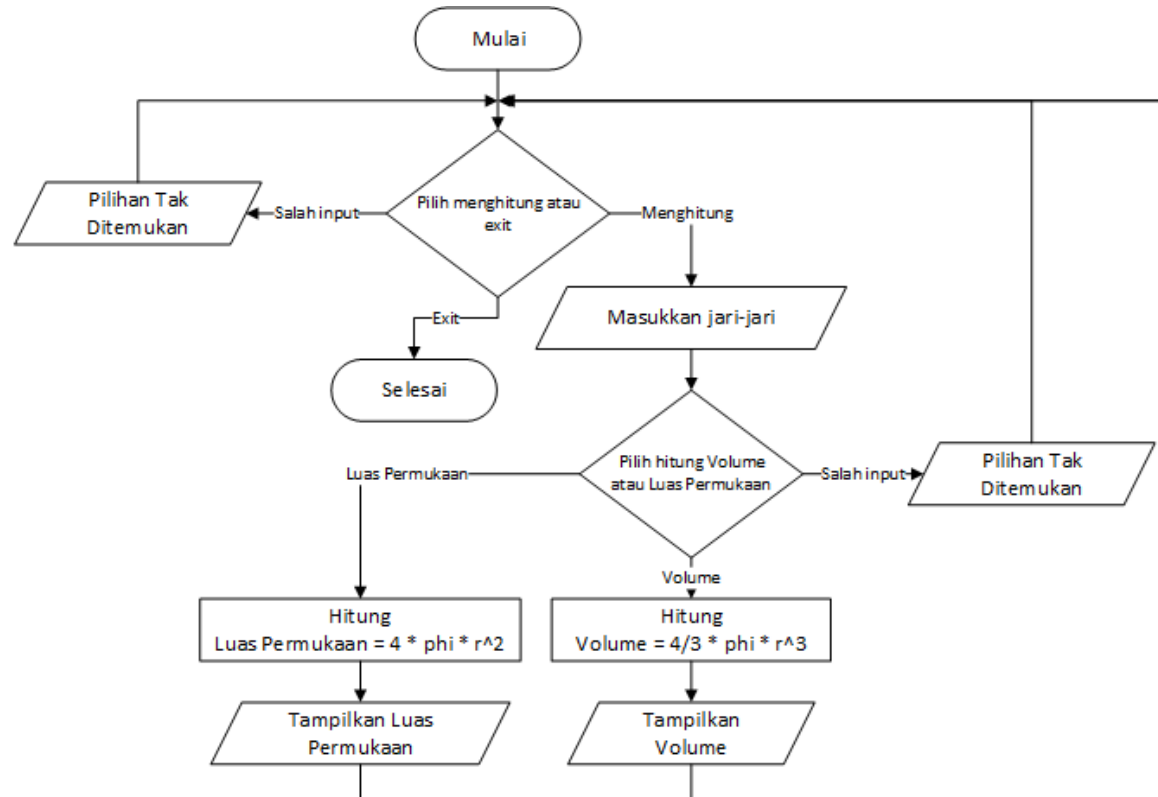
Tampilkan luas permukaan bola

Kembali ke Pilihan Melanjutkan Hitungan atau Exit

Jika pilih Exit

Selesai

Flowchart Percabangan dan Perulangan Menghitung Volume & Luas Permukaan Bola



Activity

IF Bersarang (Nested IF)

Jika suatu kondisi if terpenuhi, maka kita bisa menambahkan syarat lagi di dalamnya untuk seleksi yang lebih spesifik/mendalam.

Sintak dalam IF Bersarang Tanpa Percabangan

```
if (kondisi) :  
    if (kondisi):  
        -perintah yang akan dieksekusi jika kondisi terpenuhi-
```

Sintak dalam IF Bersarang Dengan Percabangan

```
if (kondisi) :  
    if (kondisi) :  
        -perintah yang akan dieksekusi jika kondisi terpenuhi-  
    else :  
        -perintah yang akan dieksekusi jika kondisi terpenuhi-  
else :  
    -perintah yang akan dieksekusi jika kondisi terpenuhi-
```

Merangkum Materi

Quiz

1. Apa syarat algoritma yang efisien?
2. Apa itu Big O (n)?
3. Kesulitan apa yang kalian peroleh dari menyusun algoritma, membuat flowchart, hingga program?



THANK YOU