



# Pengantar Algoritma

Erick Paulus

Mata Kuliah : Algoritma dan Pemrograman  
Program Studi S-1 Teknik Informatika



# Pokok Bahasan

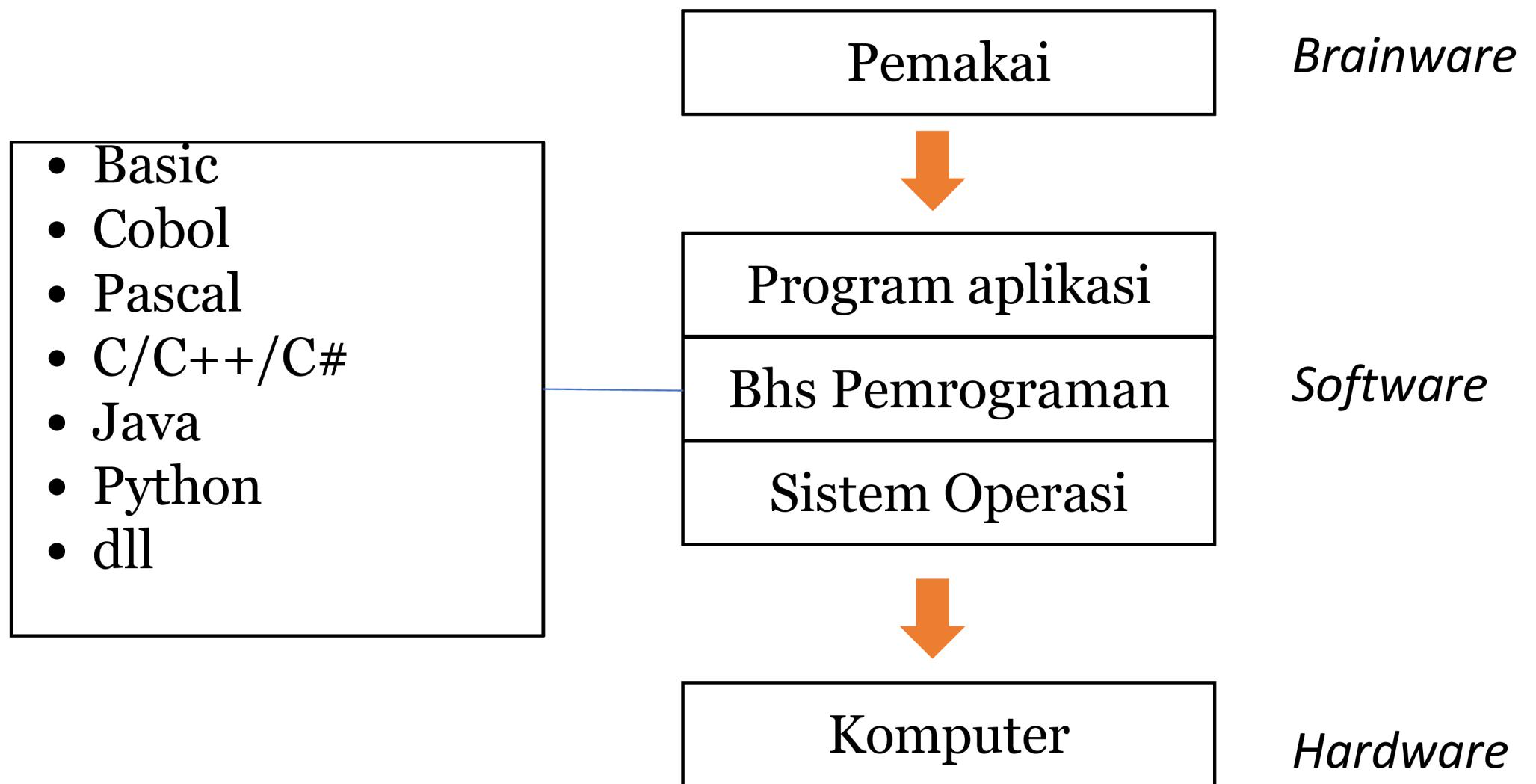
Sesi perkuliahan ini akan membahas topik-topik tentang :

1. Pendahuluan Sistem Komputer
2. Algoritma
3. Pemrograman





# *System Komputer*





# Algoritma

Sejarah





# Asal Usul Nama "Algoritma"

- **Penamaan:** Istilah "algoritma" berasal dari nama matematikawan Persia, Muhammad ibn Musa al-Khwarizmi, yang hidup pada abad ke-9.
- **Karya Al-Khwarizmi:** Al-Khwarizmi menulis buku berjudul "*Al-Kitab al-Mukhtasar fi Hisab al-Jabr wal-Muqabala*" (yang berarti "Buku Singkat tentang Perhitungan dengan Aljabar dan Perbandingan"). Karya ini adalah salah satu teks awal yang menjelaskan metode sistematis untuk menyelesaikan persamaan aljabar.
- **Pengaruh:** Buku ini diterjemahkan ke dalam bahasa Latin pada abad ke-12 dengan judul "*Algoritmi*", yang kemudian berkembang menjadi istilah "algoritma" untuk menyebut metode sistematis dalam pemrograman komputer.





# Perkembangan Sejarah Algoritma

- **Algoritma Kuno:** Konsep algoritma sudah ada sejak zaman kuno. Misalnya, metode Euclid untuk mencari Faktor Persekutuan Terbesar (GCD) yang dijelaskan dalam karya "*Elements*" pada abad ke-3 SM.
- **Era Renaissance:** Pengembangan algoritma yang lebih formal dimulai pada zaman Renaissance dengan kemajuan dalam matematika dan pemrograman.





# Kontribusi Utama dalam Sejarah Algoritma

- **Charles Babbage (1791-1871)**: Dikenal sebagai "bapak komputer". Merancang mesin analitik yang dianggap sebagai komputer pertama. Babbage mengembangkan konsep algoritma sebagai bagian dari mesin tersebut, walaupun mesin ini tidak pernah selesai dibangun.
- **Ada Lovelace (1815-1852)**: Teman Babbage, sering disebut sebagai programer komputer pertama. Menulis catatan yang mencakup algoritma untuk mesin analitik Babbage, termasuk algoritma untuk menghitung bilangan Bernoulli.
- **Alan Turing (1912-1954)**: Mengembangkan konsep mesin Turing yang membantu mendefinisikan apa itu algoritma dan komputabilitas dalam teori komputer. Turing juga merupakan pelopor dalam analisis algoritma dan kompleksitas.





# Algoritma dalam Komputasi Modern

- **John von Neumann (1903-1957)**: Mengembangkan arsitektur komputer von Neumann yang menjadi dasar bagi desain komputer modern dan mempengaruhi cara algoritma dijalankan di perangkat keras.
- **Donald Knuth (1938-)**: Menulis "*The Art of Computer Programming*", yang merupakan karya monumental tentang algoritma dan struktur data. Knuth memperkenalkan istilah "notasi Big O" untuk menganalisis kompleksitas algoritma.





# Perkembangan Terbaru

- **Algoritma dalam Kecerdasan Buatan dan Pembelajaran Mesin:** Saat ini, algoritma telah berkembang untuk mencakup berbagai teknik dalam kecerdasan buatan, seperti algoritma pembelajaran mesin dan jaringan saraf tiruan.





# Algoritma

Pengertian dan konsep dasar





# Pengertian Algoritma

- **Definisi:** Algoritma adalah urutan langkah-langkah logis yang terdefinisi dengan baik untuk menyelesaikan suatu masalah atau mencapai tujuan tertentu.
- **Karakteristik:**
  - *Finiteness:* Algoritma harus berakhir setelah sejumlah langkah tertentu.
  - *Definiteness:* Setiap langkah harus jelas dan tidak ambigu.
  - *Input:* Algoritma menerima input dari luar.
  - *Output:* Algoritma menghasilkan output yang diharapkan.
  - *Effectiveness:* Langkah-langkah dalam algoritma harus dapat dilaksanakan.





# Pentingnya Algoritma

- **Pemecahan Masalah:** Algoritma menyediakan solusi sistematis untuk masalah yang kompleks.
- **Efisiensi:** Algoritma yang baik dapat menghemat waktu dan sumber daya.
- **Standarisasi:** Algoritma memudahkan komunikasi antara programmer dengan memberikan panduan langkah demi langkah.





# Contoh Algoritma Sederhana

## Algoritma Menghitung Faktorial:

1. Mulai
2. Terima input `n`
3. Inisialisasi variabel `hasil` = 1
4. Untuk `i` dari 1 hingga `n` :
  - `hasil` = `hasil` \* `i`
5. Tampilkan `hasil`
6. Selesai

## Algoritma Pencarian Linear:

1. Mulai
2. Terima input `daftar` dan `target`
3. Untuk setiap elemen `x` dalam `daftar` :
  - Jika `x` sama dengan `target` :
    - Tampilkan "Ditemukan"
    - Hentikan pencarian
4. Jika pencarian selesai tanpa menemukan:
  - Tampilkan "Tidak Ditemukan"
5. Selesai





# Penulisan Algoritma

## 1. Uraian Deskriptif

- **Definisi:** Uraian deskriptif adalah penjelasan langkah demi langkah tentang bagaimana menyelesaikan suatu masalah atau tugas.
- Digunakan pada tahap awal desain.

- **Masalah:** Menghitung rata-rata dari serangkaian angka.
- **Langkah-Langkah:**
  1. Mulai
  2. Terima input jumlah angka `n` dan daftar angka.
  3. Inisialisasi `total` = 0.
  4. Untuk setiap angka dalam daftar:
    - Tambahkan angka ke `total`.
  5. Hitung `rata-rata` = `total` / `n`.
  6. Tampilkan `rata-rata`.
  7. Selesai





# Penulisan Algoritma

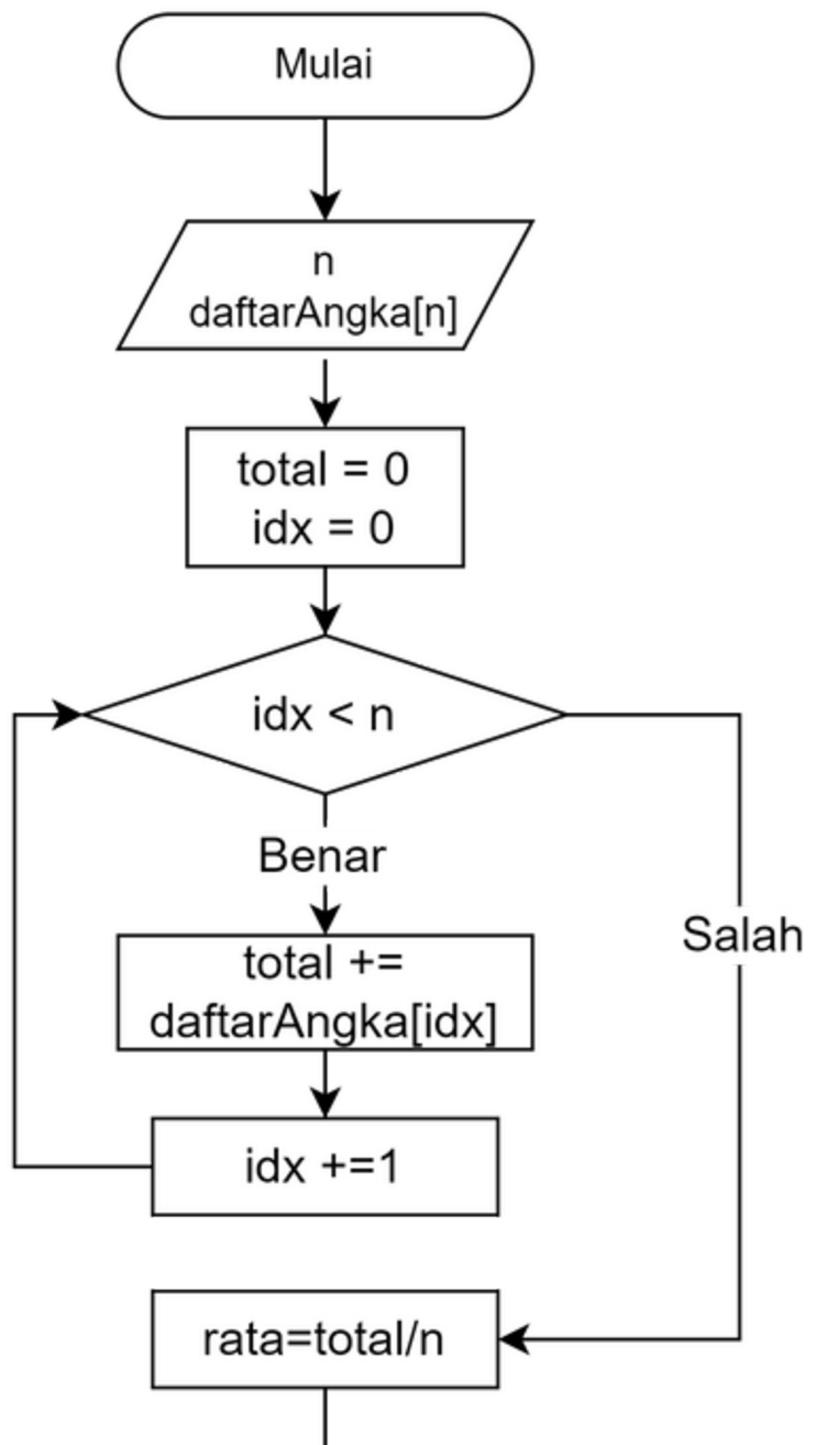
## 2. Diagram-Alir (Flowchart)

- **Definisi:** Diagram-alir adalah representasi grafis dari algoritma yang menggunakan simbol-simbol standar untuk menggambarkan langkah-langkah dan alur kontrol dalam algoritma.
- Berguna untuk memvisualisasikan alur algoritma.





# Contoh Diagram-Alir (Flowchart)





# Penulisan Algoritma

## 3. Pseudocode (Kode Semu)

- **Definisi:** Pseudocode adalah representasi algoritma dalam bentuk teks yang menyerupai bahasa pemrograman, namun lebih mudah dibaca dan dipahami. Pseudocode tidak terikat pada sintaks bahasa pemrograman tertentu.
- Bahasa universal algoritma dan biasa dipakai dalam artikel ilmiah





# Struktur Pseudocode

- Bagian kepala (header) : memuat nama algoritma serta informasi atau keterangan tentang algoritma yang ditulis
- Bagian Deklarasi (definisi variable ) : memuat definisi nama variable, nama tetapan, nama prosedur, nama fungsi, tipe data yang akan digunakan dalam algoritma.
- Bagian Deskripsi (rincian langkah) : memuat langkah-langkah penyelesaian masalah, termasuk beberapa perintah seperti baca data, tampilkan, ulangi, yang mengubah data input menjadi output.





# Contoh

## **Algoritma** Luas\_persegi\_panjang

{Menghitung luas persegi panjang apabila panjang dan lebar persegi panjang tersebut diberikan}

### **Deklarasi**

{Definisi nama peubah/variabel}

float panjang, lebar, luas

### **Deskripsi**

read(panjang,lebar) // input / baca

luas  $\square$  panjang \* lebar

write(Luas) // output / tulis





# Tipe-Tipe Algoritma

- **Algoritma Berurutan:** Menyelesaikan langkah-langkah secara berurutan (contoh: algoritma menghitung faktorial).
- **Algoritma Pemisahan dan Penaklukan:** Memecah masalah menjadi sub-masalah yang lebih kecil (contoh: Merge Sort).
- **Algoritma Dinamis:** Menggunakan hasil perhitungan sebelumnya untuk menghindari perhitungan ulang (contoh: Algoritma Fibonacci).
- **Algoritma Greedy:** Membuat keputusan terbaik secara lokal untuk mencapai solusi global (contoh: Algoritma Kruskal).





# Notasi Algoritma

- **Notasi Big O:** Mengukur kompleksitas waktu dan ruang dari algoritma.
  - $O(1)$ : Waktu tetap
  - $O(n)$ : Waktu linier
  - $O(n^2)$ : Waktu kuadrat
  - $O(\log n)$ : Waktu logaritmik





# Analisis Algoritma

- **Kompleksitas Waktu:** Waktu yang dibutuhkan algoritma untuk menyelesaikan masalah sebagai fungsi dari ukuran input.
- **Kompleksitas Ruang:** Jumlah memori yang digunakan algoritma selama proses eksekusi.





# Implementasi dan Uji Coba

- **Pseudocode:** Representasi algoritma yang mudah dipahami tanpa terikat pada sintaks bahasa pemrograman tertentu.
- **Implementasi Kode:** Mengubah pseudocode menjadi kode program menggunakan bahasa pemrograman pilihan.
- **Uji Coba:** Menguji algoritma dengan berbagai kasus untuk memastikan keakuratannya.





# Contoh Implementasi

- CUI (Character User Interface)
  - C++ □ MinGW ( Visual Studio code, Borland C++)
- GUI (Graphical User Interface)
  - Java □ Netbeans





MinGW Developer Studio - [Hello1.cpp]

File Build Debug Tools Window Help

Hello1.cpp

```
1 #include <iostream>
2 #include <conio.h>
3
4 using namespace std;
5
6 int main() {
7     float panjang,lebar,luas;
8     cout<<"Panjang = "; cin>>panjang;
9     cout<<"Lebar   = "; cin>>lebar;
10    luas=panjang*lebar;
11    cout<<"Luas   = " <<luas<<endl;
12
13    getch();
14 }
15
```

E:\akmal\kuliah\struktur

Panjang = 5  
Lebar = 3  
Luas = 15

Activate Window Go to PC settings to ac

Chrome File Excel Word Skype Google Chrome Cube 3D Model Paint

The screenshot displays the MinGW Developer Studio interface. On the left is the code editor with a file named 'Hello1.cpp' containing a simple C++ program. The program includes headers for iostream and conio.h, uses the std namespace, and defines a main function that reads length and width from the console, calculates their product, and prints the result. On the right, a terminal window titled 'E:\akmal\kuliah\struktur' shows the execution of the program, displaying the input values (5 and 3) and the calculated area (15). The taskbar at the bottom shows icons for various applications like Google Chrome, Microsoft Office, and system tools.



PersegiProject - NetBeans IDE 7.3

File Refactor Run Debug Profile Team Tools Window Help

<default config> T T D G S

StartPage Stack.java Fkalkulator.java FormUtama.java PersegFrame.java CUIPerseg.java Main.java

Source Design History

```
264
265     private void prosesHitungLuas() {
266         float panjang, lebar, luas;
267
268         try{
269             panjang=Float.parseFloat(panjangField.getText());
270             lebar=Float.parseFloat(lebarField.getText());
271             luas = panjang*lebar;
272             //luas=hitungLuas(panjang,lebar);
273             luasField.setText(String.valueOf(luas));
274         }
275         catch(Throwable t) {
276             JOptionPane.showMessageDialog(null,"Nilai bukan numerik, "
277             + "silahkan diperbaiki untuk bisa diproses");
278
279             bersihButton.requestFocus();
280         }
281     }
282
283     private void bersihField(){
284         <
```

PersegFrame > prosesHitungLuas > try >

Output - PersegiProject (run) >

run:

Program Persegi Panjang

Persegi Panjang

Masukkan Panjang Persegi Panjang :

Masukkan Lebar Persegi Panjang :

Luas Persegi Panjang :

Proses Bersihkan



The screenshot shows the Microsoft Visual Studio Code (VS Code) interface with the C/C++ extension installed. The main editor window displays a C++ file named `test.cpp` containing the following code:

```
/* Nama program      : hello.cpp
Nama             : Erick
NPM              : 1234
Tanggal buat : 19 Agustus
Deskripsi       : Pencetakan Hello Unpad 2 kali

#include <iostream>
using namespace std;
main() {
    double pi = 3.141592654;
    // Tampilan default: left justified, presisi 6.
    cout << pi << endl;
    // Ubah dg presisi 4, lebar field 12, isi dg #
    cout.precision(4);
    cout.width(12);
    cout.fill('#');
    cout << pi << endl;
    // Ubah presisi ke 10
    cout.precision(10);
    cout << pi << endl;
}
```

The sidebar on the left contains several tabs and icons for debugging, variables, watch, call stack, and breakpoints. The bottom status bar shows the current file path as `C:\Users\ErickP\Documents\Unpad\Bahan Ajar\Algo\code>`.

In the bottom right corner of the terminal tab, there is a dropdown menu with the following options:

- powershell
- C/C++: g++.exe
- cppdbg: test.exe
- cppdbg: test.exe



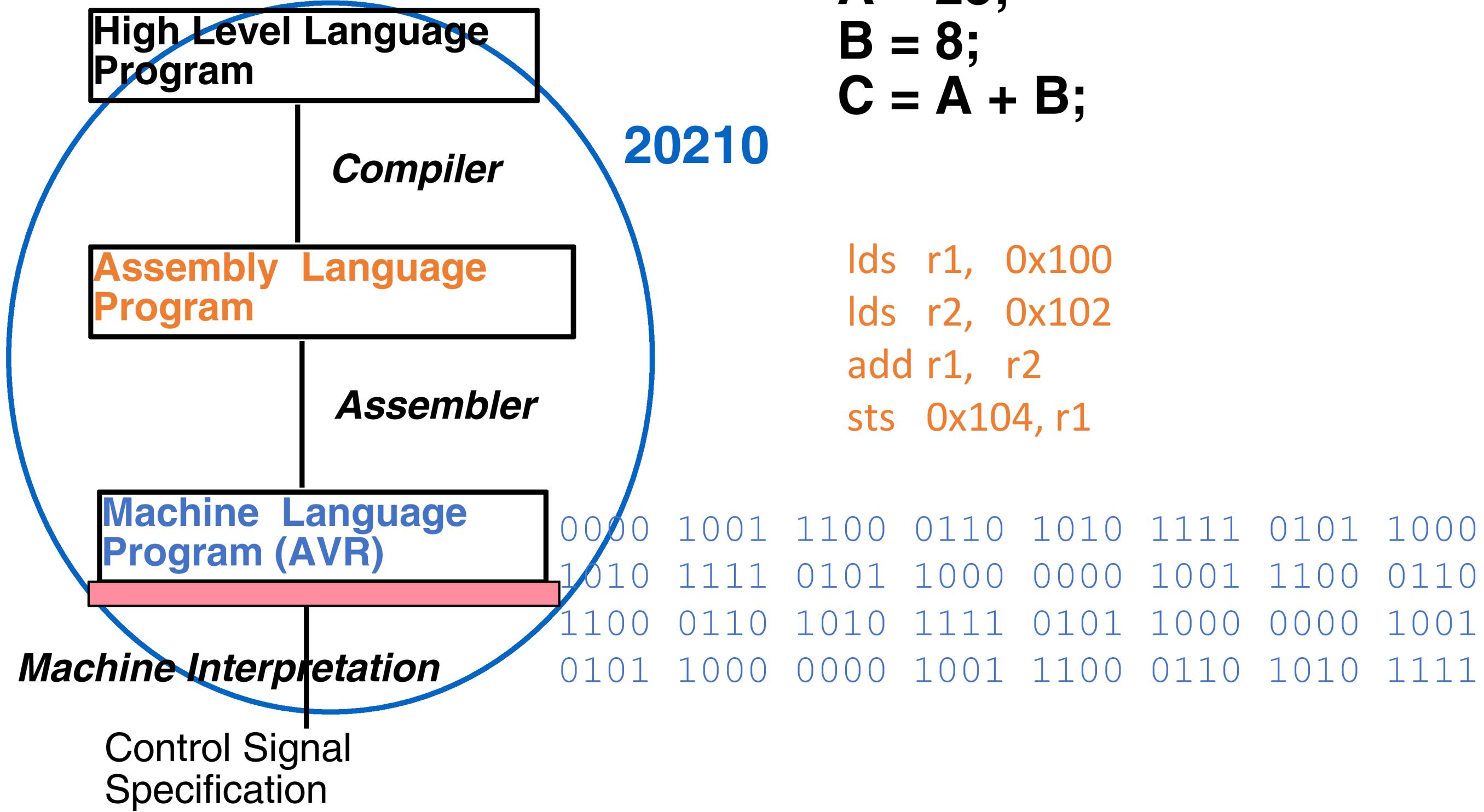
# Pemrograman

- Program secara umum didefinisikan sebagai kumpulan instruksi atau perintah yang disusun sedemikian rupa sehingga mempunyai urutan nalar yang logis untuk menyelesaikan suatu persoalan yang dimengerti oleh komputer.
- Pemrograman adalah aktifitas yang berhubungan dengan pembuatan program dengan mengikuti kaidah bahasa pemrograman tertentu.
- Dalam konteks pemrograman terdapat sejumlah bahasa pemrograman seperti Pascal, Delphi, C, C++, C#, Java, Python dll.





## Tingkat Bahasa Pemrograman (Tinggi dan Rendah)





# Paradigma Pemrograman

- Digunakan beberapa pendekatan dalam bidang pemrograman (paradigma) yaitu : sudut pandang tertentu yang diprioritaskan terhadap kelompok problema, realitas, keadaan dll.

**Antara Lain :**

- Pemrograman **Prosedural / Terstruktur**
- Pemrograman **Fungsional**
- Pemrograman **Berorientasi Obyek**
- Pemrograman **Berorientasi Fungsi**
- Pemrograman **Deklaratif**





# Paradigma Pemrograman

- Pemrograman **Prosedural / Terstruktur**

- Berdasarkan urutan-urutan, sekuensial
- Program adalah suatu rangkaian prosedur untuk memanipulasi data. Prosedur merupakan kumpulan instruksi yang dikerjakan secara berurutan.
- Harus mengingat prosedur mana yang sudah dipanggil dan apa yang sudah diubah.
- Program dapat dibagai-bagi menjadi prosedur dan fungsi.
- Contoh: PASCAL dan C





# Paradigma Pemrograman

- Pemrograman **Fungsional**
  - Berdasarkan teori fungsi matematika
  - Fungsi merupakan dasar utama program.
  - Contoh : Lisp
- Pemrograman **Berorientasi Obyek**
  - Pemrograman berdasarkan prinsip obyek, dimana obyek memiliki data/variabel/property dan method/event/prosedur yang dapat dimanipulasi
  - Contoh: C++, C#, Object Pascal, Python , Java dll
- Pemrograman **Berorientasi Fungsi**
  - Pemrograman ini berfokus pada suatu fungsi tertentu saja. Sangat tergantung pada tujuan pembuatan bahasa pemrograman ini.
  - Contoh: SQL (Structured Query Language), HTML, XML dan lain-lain.
- Pemrograman **Deklaratif**
  - Pemrograman ini mendeskripsikan suatu masalah dengan pernyataan daripada memecahkan masalah dengan implementasi algoritma.
  - Contoh: PROLOG





# Langkah-langkah Penyelesaian Masalah dengan Pemrograman

- Menganalisis masalah
  - Yaitu tindakan untuk mengidentifikasi informasi yang menjadi keluaran pemecahan masalah dan data-data yang menjadi masukan.
  - Dengan kerangka pemecahan masalah :
    - **Masukan Proses Keluaran**
- Membuat algoritma
  - Yaitu menuangkan ide dari pengidentifikasian masalah ke dalam bentuk algoritma baik dengan menggunakan Flowchart atau PseudoCode
- Menuangkan algoritma ke dalam bentuk program.
  - Yaitu proses membuat kode dengan menggunakan sebuah bahasa pemrograman untuk mendapatkan hasil sesuai dengan permasalahan
- Mengeksekusi , menguji program (implementasi) dan dokumentasi
  - Yaitu proses pengujian terhadap suatu program yang digunakan, apakah nantinya berhasil sesuai yang diharapkan atau masih terjadi kesalahan-kesalahan.





# Tabel Konversi Penugasan, Masukan dan Keluaran

Kelompok	PseudoCode	C / C++	Python
Assignment / Penugasan	<input type="checkbox"/>	=	=
Operator relasi	=	==	==
Masukan	input( ) baca / read	scanf / cin	input()
Keluaran	output() tampilkan / cetak /write	printf / cout	print()





## Belajar memprogram atau belajar bahasa pemrograman ?

Belajar memprogram : belajar tentang strategi pemecahan masalah, metodologi dan sistematika pemecahan masalah yang kemudian dituangkan dalam suatu notasi yang disepakati bersama (bahasa program)

==> lebih bersifat pemahaman persoalan, analisis, sintesis

==> designer

Belajar bahasa pemrograman : belajar memakai suatu bahasa, aturan sintaks (tata bahasa) setiap instruksi yang ada dan tata cara pengoperasian kompilator bahasa yang bersangkutan pada mesin tertentu

==> lebih bersifat keterampilan

==> juru kode (coder)

Ilmu pemrograman berkembang menggantikan “Seni” memprogram atau memprogram secara coba-coba (“trial and error”).





# Latihan

## Menyusun Algoritma

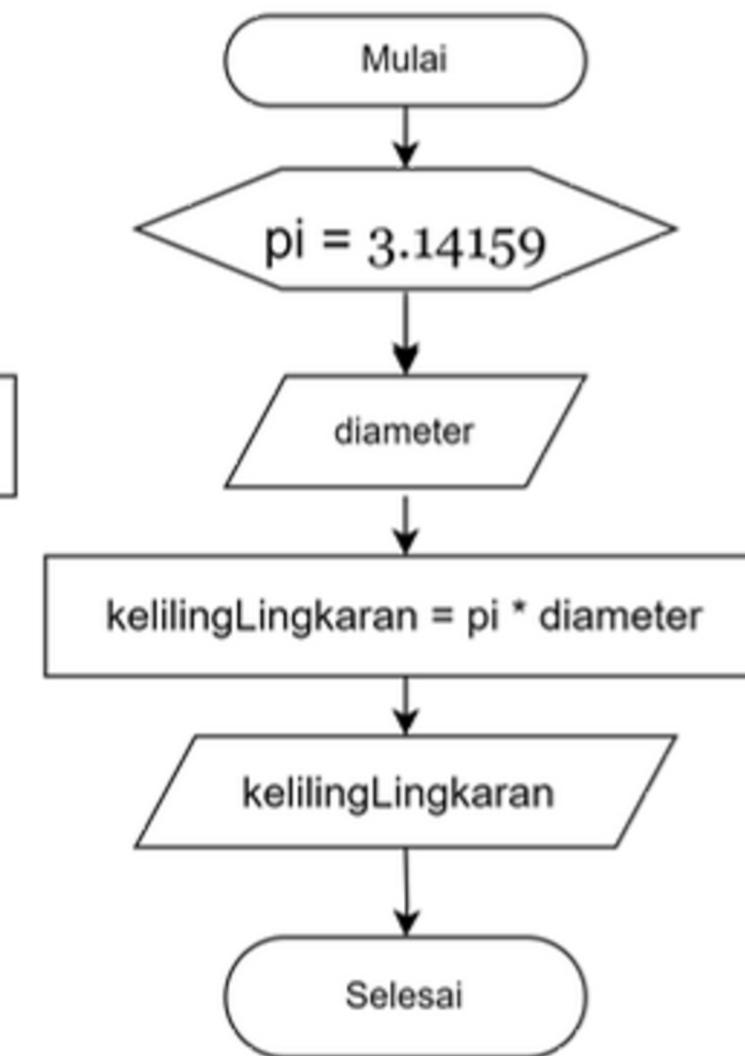
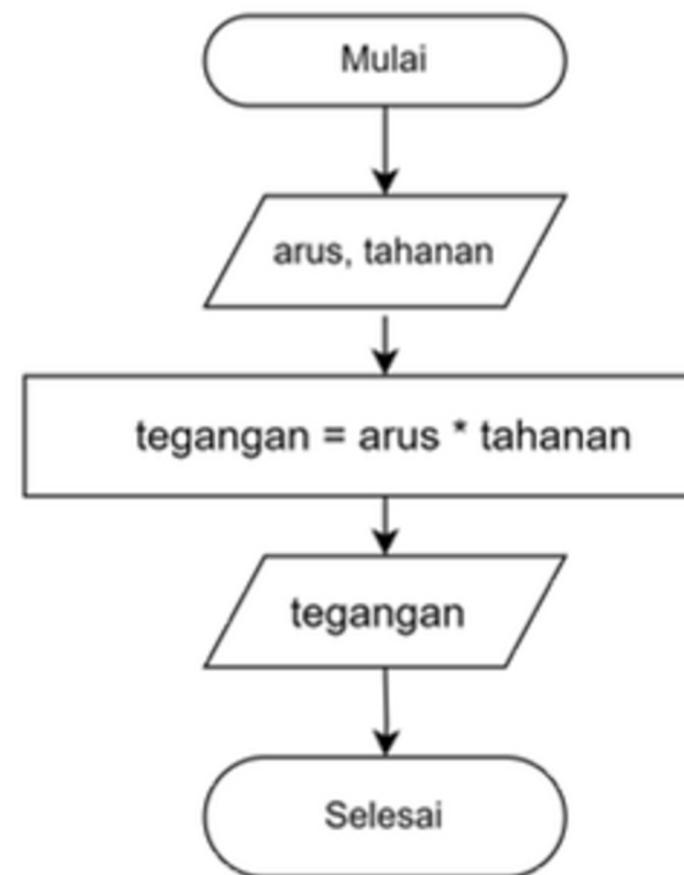
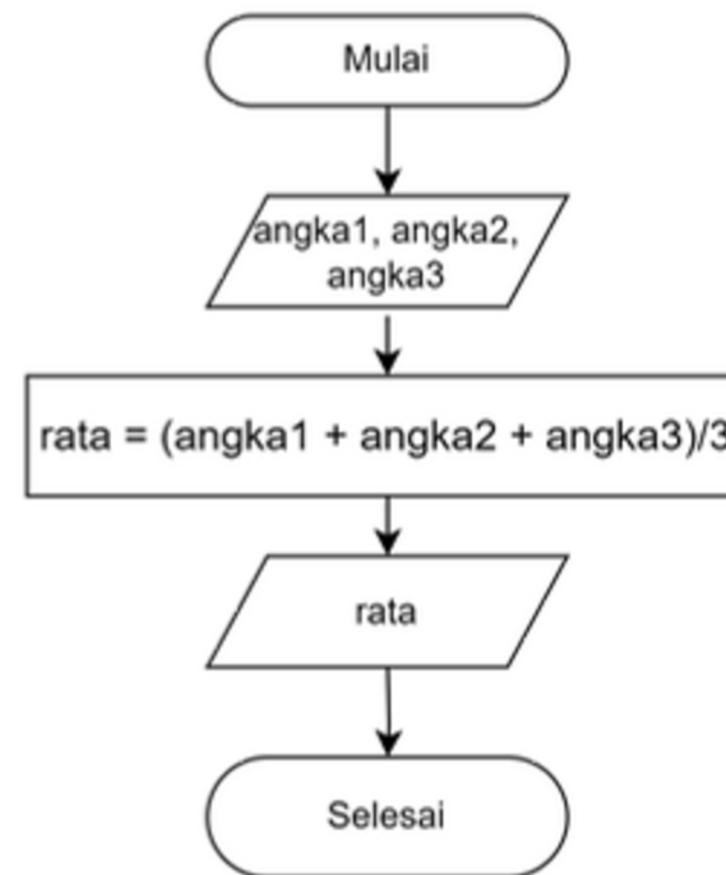
Dari masalah-masalah berikut, buatlah algoritma untuk menyelesaikannya. Anda bisa menyusun algoritma tersebut dalam bentuk pseudocode atau diagram-alir (flowchart)

- **Soal 1:** Buatlah algoritma untuk menghitung rata-rata dari tiga bilangan bulat yang diberikan sebagai input.
  - Input: 5, 10, 15
  - Output: 10 (karena rata-rata dari 5, 10, dan 15 adalah  $(5 + 10 + 15) / 3 = 10$ )
- **Soal 2:** Diberikan dua nilai sebagai input: arus (dalam Ampere) dan tahanan (dalam Ohm). Buatlah algoritma untuk menghitung tegangan yang dihasilkan berdasarkan hukum Ohm ( $V = I * R$ ).
  - Input: Arus = 3 Ampere, Tahanan = 4 Ohm
  - Output: Tegangan = 12 Volt (karena  $V = 3 * 4 = 12$ )
- **Soal 3:** Diberikan nilai sebagai input: diameter lingkaran. Buatlah algoritma untuk menghitung keliling lingkaran. Catatan  $\pi$  (pi) kira-kira 3.14159.
  - Input: diameter = 10
  - Output: keliling lingkaran = 31.4159 (karena  $K = \pi \times d = 3.14159 * 10 = 31.4159$ )





# Jawaban





# Latihan

- **Soal 3:** Diberikan sebuah nilai dalam rupiah (bilangan bulat positif), buatlah algoritma untuk menghitung nilai ekuivalen dalam dolar AS (\$). Tampilkan hasilnya dan pertimbangkan perubahan kurs yang sering terjadi. Jelaskan bagaimana Anda akan menyesuaikan perhitungan jika juga perlu menghitung ekuivalen dalam Yen, Euro, atau mata uang lainnya.
  - Input: Rp.100000, Kurs Rupiah ke Dolar AS = 15000
  - Output: 6.67 Dollar AS (karena  $100000 / 15000 = 6.67$ )
- **Soal 4:** Diberikan sebuah nilai suhu dalam derajat Celsius, buatlah algoritma untuk menghitung nilai ekuivalennya dalam derajat Fahrenheit, Reamur, dan Kelvin.
  - Input: Suhu =  $25^{\circ}\text{C}$
  - Output:
    - Fahrenheit =  $77^{\circ}\text{F}$  (karena  $(25 * 9/5) + 32 = 77$ )
    - Reamur =  $20^{\circ}\text{Re}$  (karena  $25 * 4/5 = 20$ )
    - Kelvin = 298.15 K (karena  $25 + 273.15 = 298.15$ )





# Latihan



Soal 5: Buatlah algoritma untuk menghitung faktorial dari sebuah bilangan bulat positif n. Faktorial dari n (ditulis sebagai  $n!$ ) adalah hasil perkalian semua bilangan bulat positif dari 1 hingga n.

Input: 5

Output: 120 (karena  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ )



Soal 6: Buatlah algoritma untuk mencari sebuah nilai dalam sebuah daftar yang tidak terurut menggunakan pencarian linear. Input: Daftar = [3, 7, 1, 8, 5], Target = 8 Jika nilai ditemukan, tampilkan indeksnya. Jika tidak ditemukan, tampilkan pesan "Tidak Ditemukan". Output: 3 (karena nilai 8 ada di indeks 3)



Soal 7: Buatlah algoritma untuk mengurutkan sebuah daftar angka menggunakan metode Bubble Sort. Dalam Bubble Sort, setiap pasangan elemen yang berdekatan dibandingkan dan ditukar jika urutannya salah, sampai daftar menjadi terurut.

Input: [64, 34, 25, 12, 22, 11, 90]

Output: [11, 12, 22, 25, 34, 64, 90]



# Terima kasih

---

- *Algorithm design is not about discovering the best algorithm. It is about discovering a good enough algorithm that works efficiently for the problem at hand*

Tenaga Pengajar :

- Erick Paulus, S.Si.,
- Dr. Akmal, S.Si, MT

