FIL-150013

PENGANTAR ILMU KOMPUTER

PEMROGRAMAN & ALGORITMA

Dr. Eng., Herman Tolle, ST., MT.

OUTLINE

- Filosofi Pemrograman
- Konsep Pemrograman
- Konsep Algoritma

PROGRAMMING PHILOSOPHY

Definisi

Programming:

- The action or process of writing computer programs
- Kegiatan atau proses menulis program komputer

Programmer:

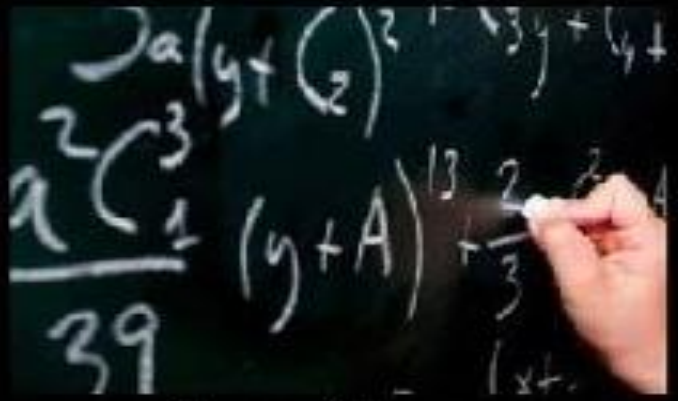
- a person who writes computer program
- Orang yang menulis program komputer

“ Everybody should learn how to program a computer… because it teaches you how to think.”

- Steve Jobs

Programmer

who i think i will like



what i learn

what my father think i do

what my friend think i do

what should i do

what i actually do

A Programmer compared to other profession

Manakah profesi yang posisi pekerjaannya setara dengan programmer?

Programmer is Creator (Pencipta) & *Developer* (Pembangun)

Disain Program (Algoritma)

- Chef / Koki (kreasi / imaginasi)
- Designer Pakaian
- Sutradara Film
- Arsitek / Perancang Bangunan
- Pelukis (imaginasi)

Membuat Code Program (Coding)

- Chef / Koki (teknik membuat makanan)
- Kuli bangunan
- Tukang Kayu
- Tukang jahit
- Pelukis (teknik melukis)

Kuliah Pemrograman Dasar mengajari [1] bagaimana **persoalan sederhana diselesaikan dengan algoritma komputer (membuat algoritma)**, dan [2] bagaimana **menterjemahkan algorima** tersebut menjadi program komputer **(membuat kode program)**

Perbandingan Chef dengan Programmer

Chef

- Memiliki dasar pengetahuan dan teknik pengolahan masakan
- Bekerja mengikuti resep masakan
- Apa beda masterchef dengan normal chef?
  - Menyajikan dengan style
  - Modifikasi resep
  - Bekerja efisien

Programmer

- Memiliki dasar pengetahuan dan teknik pemrograman (Sense of Programming
- Bekerja mengikuti Algoritma / Flowchart
- Good programmer?
  - Menyajikan dengan style
  - Algoritma yang efektif
  - Bekerja efisien

Be a good programmer!

- Algoritma yang efektif dan efisien
- Bekerja dengan efisien
- Menyajikan dengan style

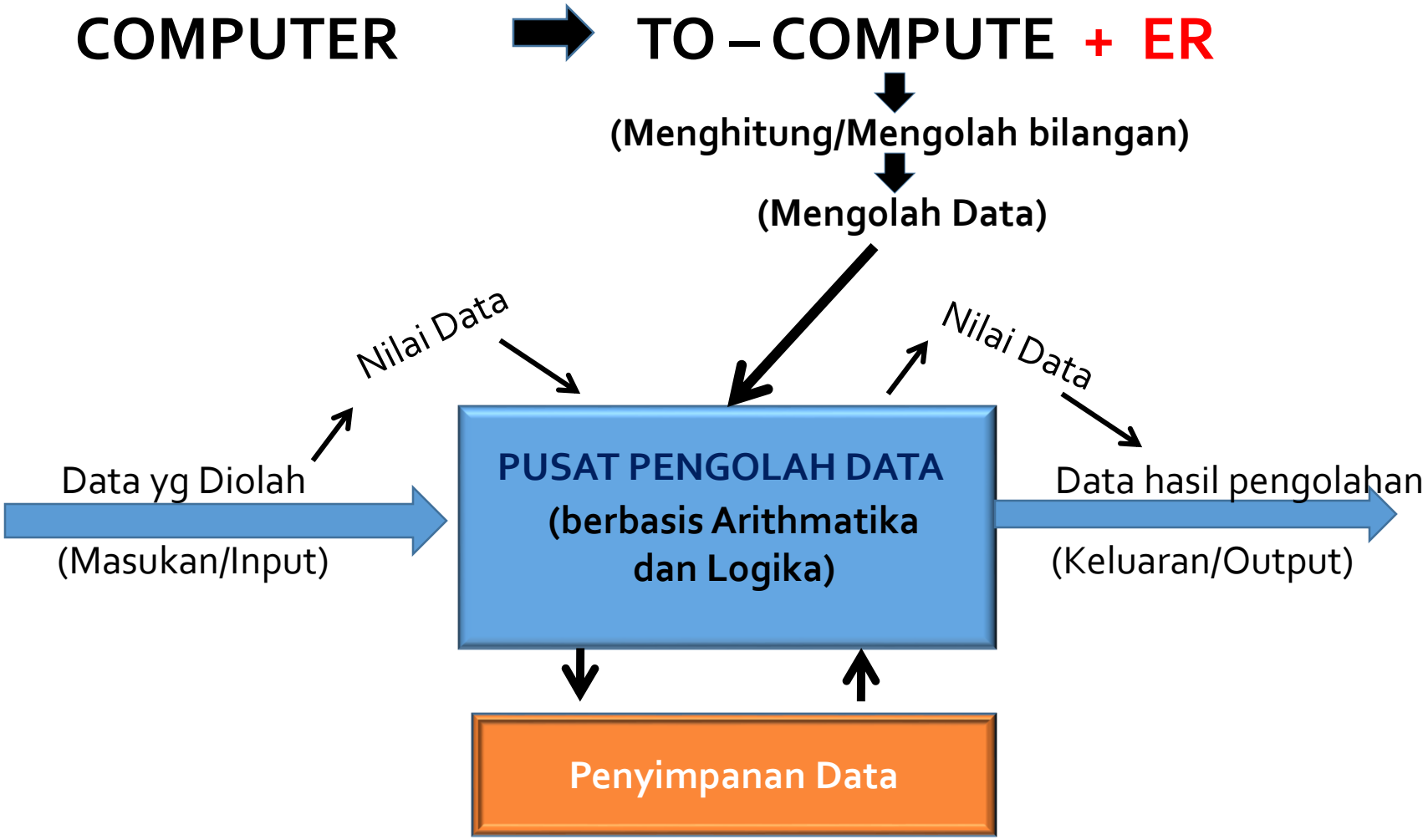
PROGRAMMING is FUN

KONSEP PEMROGRAMAN

Komponen Komputer

- **Perangkat keras**: perangkat komputer yang dapat disentuh secara fisik → ***Hardware***
- **Perangkat lunak**: program yang berisikan perintah-perintah yang menentukan operasi yang dilakukan oleh komputer → ***Software***
- **Pengguna (user)**: orang yang menggunakan komputer → ***Brainware***
  - **User, Operator, Programmer, Developer**

KONSEP DASAR PEMROGRAMAN KOMPUTER



Perangkat Keras

- **Perangkat *Input***: *keyboard*, *mouse*, *scanner*
- **Perangkat Pemroses**: *mainboard*, *prosesor*
- **Perangkat Penyimpan Data**: *memori*, *hard disk*, *flash disk*
- **Perangkat *Output***: *monitor*, *printer*, *speaker* **Perangkat Penunjang**: berbagai *card* (VGA *card*, *sound ca*rd, LAN *card*, TV *card*).

Rerangkat Lunak

- Perangkat keras tidak bisa bekerja tanpa perintah yang dikendalikan oleh **perangkat lunak** yang ditulis / dibuat oleh manusia

3 jenis Perangkat lunak:

- Sistem Operasi,
- Program Aplikasi, dan
- Bahasa Pemograman

Sistem Operasi

- **Untuk membantu agar komputer dapat digunakan setiap orang** (meski dia tidak tahu cara kerja perangkat keras) maka dibuatlah sebuah sistem operasi
- Contoh : Windows, Unix, Linux, FreeBSD, Solaris, Macintosh, dll
- **Sistem operasi berguna untuk mengatur seluruh operasi dan sumber daya perangkat keras komputer**
- Sistem operasi ditulis oleh pabrik software sehingga mudah digunakan oleh orang yang tidak mengerti komputer secara rinci, jadi berfungsi sebagai antarmuka (interface) mesin-manusia
- Misalnya: kita bisa menggunakan perintah print pada komputer untuk mencetak tanpa harus tahu bagaimana komputer dan printer bekerja

Program Aplikasi

- Program Aplikasi adalah **program komputer** yang ditulis **untuk dapat menyelesaikan permasalahan atau pekerjaan tertentu**
- **Dapat dibeli (sudah dibuatkan oleh orang lain / software)** atau **dibuat sendiri** menggunakan bahasa pemograman
- Dibuat oleh programmer menggunakan **bahasa pemograman**
- Misalnya Ms Word untuk olah dokumen, Ms Excel untuk olah tabel spreedsheet, Ms Powerpoint untuk presentasi, game, pengolah grafis untuk edit gambar, multimedia player, dll

Jenis Program Aplikasi

- Aplikasi Office: MS Word, MS PowerPoint, MS Excel
- Pengolah Grafis: Adobe Photoshop, Corel Draw
- Pengolah Video: Adobe Premiere
- Utility: Data Recovery,
- Internet Browser
- Game, Multimedia Player, dll

BAHASA PEMOGRAMAN

Bahasa Pemograman

- Bahasa pemograman adalah program komputer yang berguna untuk **memberikan perintah kepada komputer** untuk **menyelesaikan permasalahan** tertentu atau **menghasilkan program aplikasi** tertentu (program aplikasi, sistem operasi, dll)
- Bahasa Pemrograman mempermudah manusia (programmer) untuk berinteraksi dan menggunakan sumber daya yang ada pada sebuah mesin komputer

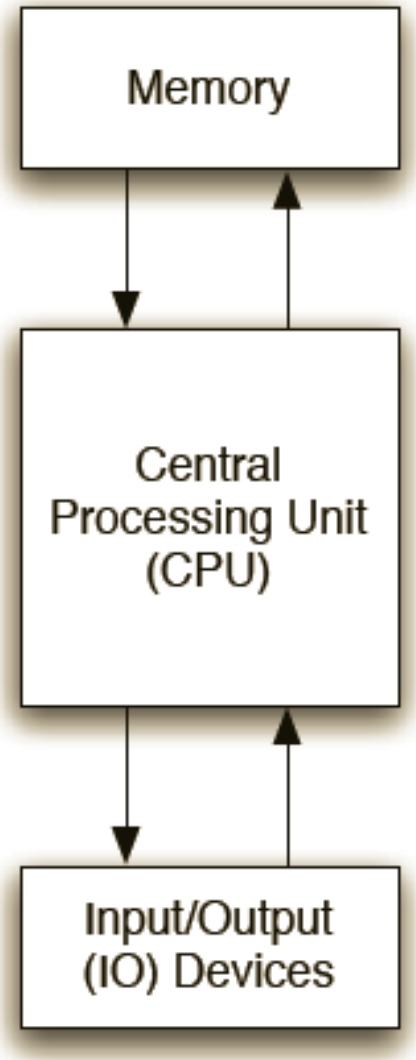
Bahasa Pemrograman

- Bahasa komunikasi manusia dengan komputer untuk memberikan instruksi kepada komputer
- *Low Level Programming Language* → dekat ke Bahasa mesin (01)
  - Bahasa Mesin
  - Bahasa Assembly
- *High Level Programming Language* → dekat ke Bahasa manusia

Tingkatan Bahasa Pemrograman

- Bahasa Mesin
- *Low Level Language (Mnemonic, Assembler*)
- Middle Level Language
- High Level (3rd Generation) Language
- Generasi ke 4 → GUI based Programming (drag and drop)
- Generasi ke-5 → Agile Programming

The Computer



CPU Instructions

$z = x + y$

Read location x

Read location y

Add

Write to location z

Bahasa Mesin

- Pada prinsipnya komputer (CPU) bekerja atas perintah dalam bentuk sinyal bit positif (1) dan sinyal bit negatif (0)
- Perintah dalam bentuk kombinasi biner (bit 1 dan bit 0) sedemikian sehingga komputer dapat bereaksi sesuai perintah tersebut, misal: 0110010110011O
- Kombinasi bit 0 dan 1 sangat sulit untuk diingat manusia

Low Level Programming Language

Bahasa Mesin

- 8B542408 83FA0077 06B80000 0000C383 FA027706 B8010000 00C353BB 01000000 B9010000 008D0419 83FA0376 078BD98B C84AEBF1 5BC3

Assembly

```
fib:
    mov edx, [esp+8]
    cmp edx, 0
    ja @f mov eax, 0
    ret @@:
    cmp edx, 2
    ja @f mov eax, 1
    ret @@:
    push ebx
    mov ebx, 1
    mov ecx, 1
```

High Level Programming Language

- Mendekati bahasa manusia (natural language)
- Menggunakan istilah-istilah dalam bahasa Inggris dan notasi matematis umum
- *Third-generation programming language* (Fortran, ALGOL, and COBOL)
- Pemrograman **Terstruktur.**
- *General-purpose languages* C, C++, C#, Java, BASIC and Pascal

Bahasa Tingkat Tinggi

- Satu pernyataan menyelesaikan tugas-tugas substantial
- Program **compiler** → mengkonversi ke bahasa mesin (seluruh kode program)
- Program **interpreter** → mengeksekusi perintah satu persatu dalam bahasa tingkat tinggi secara langsung

Paradigma Pemrograman

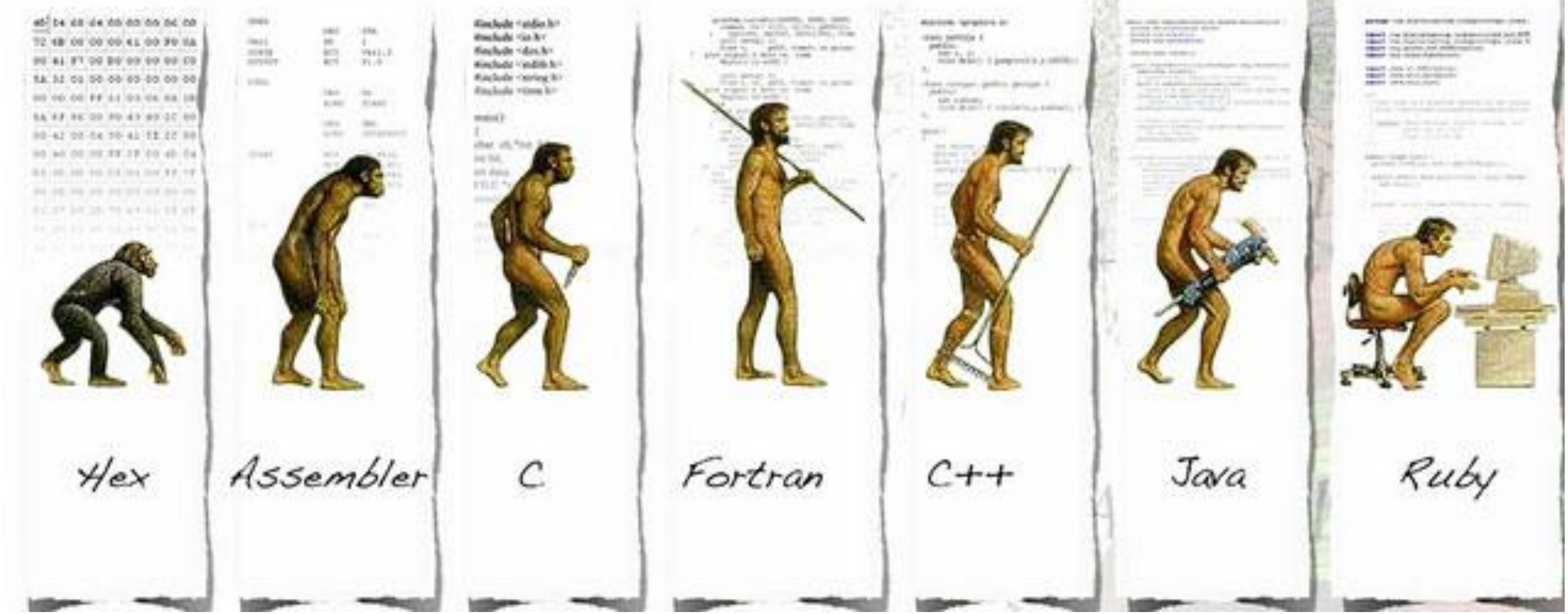
- Pemrograman **Terstruktur / Sequiensial** → diajarkan di MK Pemrograman Dasar
- Pemrograman **Berorientasi Objek (OOP)** → Pemrograman Lanjut
- Pemrograman **Berbasis Kejadian** (Event Based Programming)
- Pemrograman Visual / Window / GUI
  - Pemrograman Platform Khusus (Semester 5)

Object Oriented Programming

- Pemrograman Berorientasi Objek
- C++, Java and C#

- Memodelkan kode program komputer dari perspektif bagaimana manusia memandang sesuatu sebagai sebuah objek
- *Object-Class-Attribute-Behaviour*

The Evolution Of Computer Programming Languages

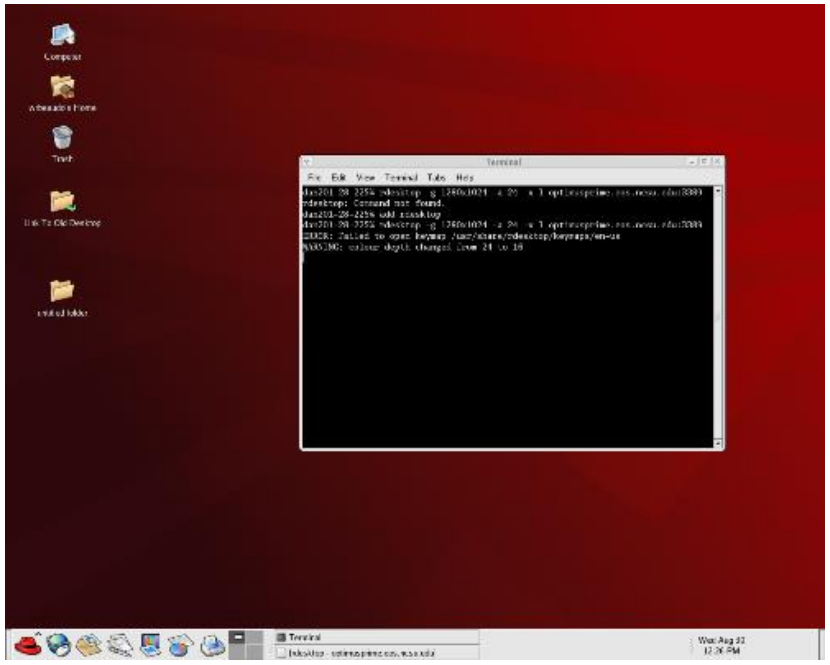


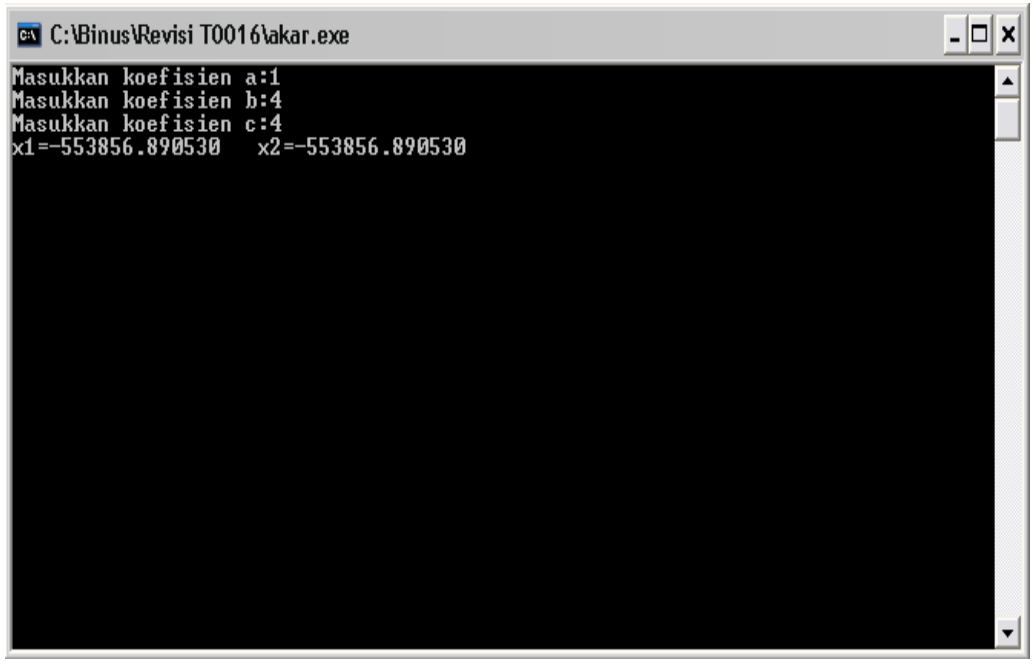
Jenis Program berdasarkan Output

- Console
- Console with GUI
- Desktop / Windows / GUI
- Web
- Mobile
- Wearable

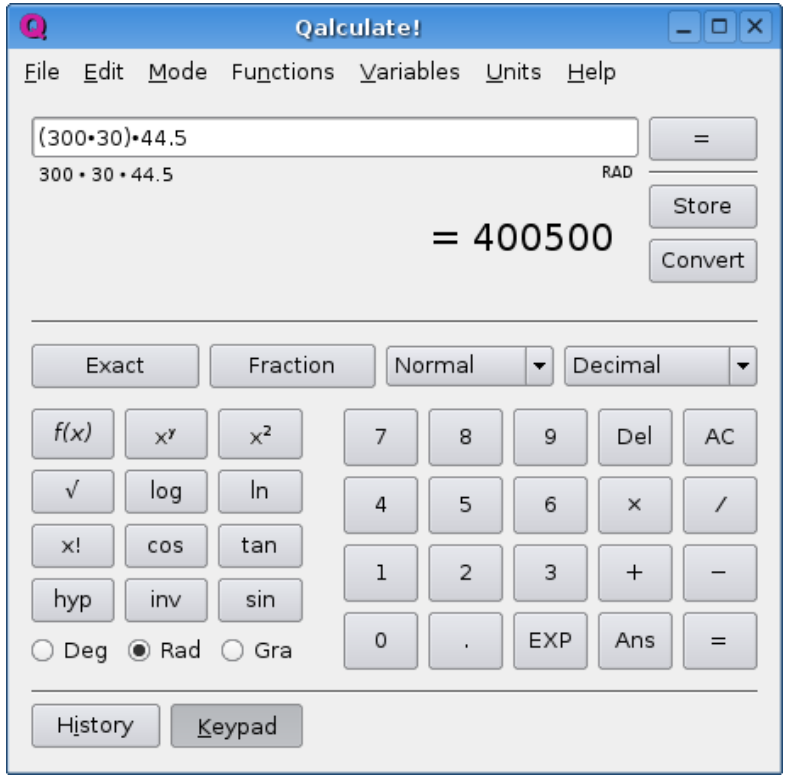
Console Programming

- Input & Output program pada layar *console*
- Command line Instruction (CLI)





Desktop / Windows / GUI



Paradigma Pemrograman:

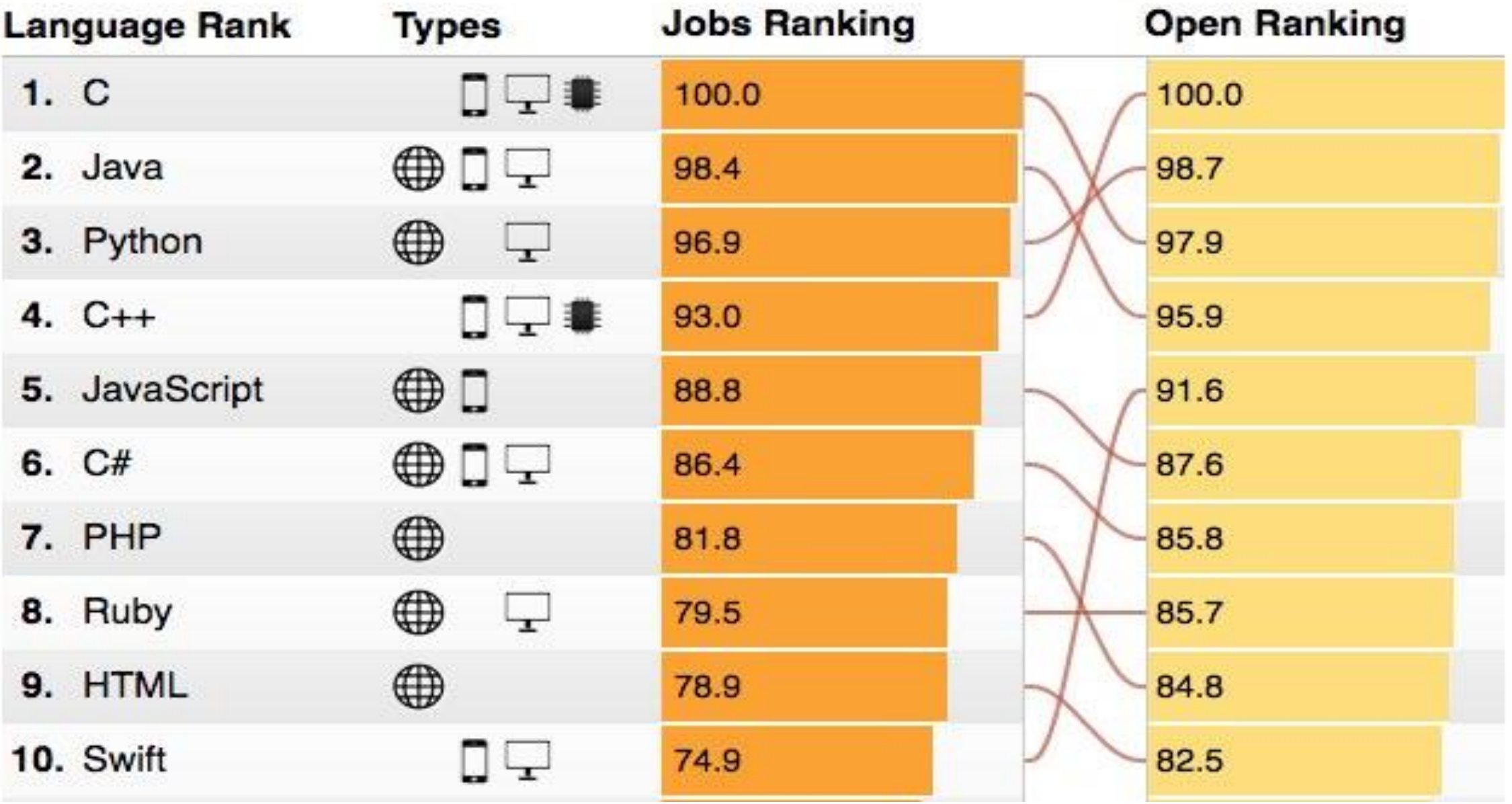
- Event-Based Programming

Programming Lainnya

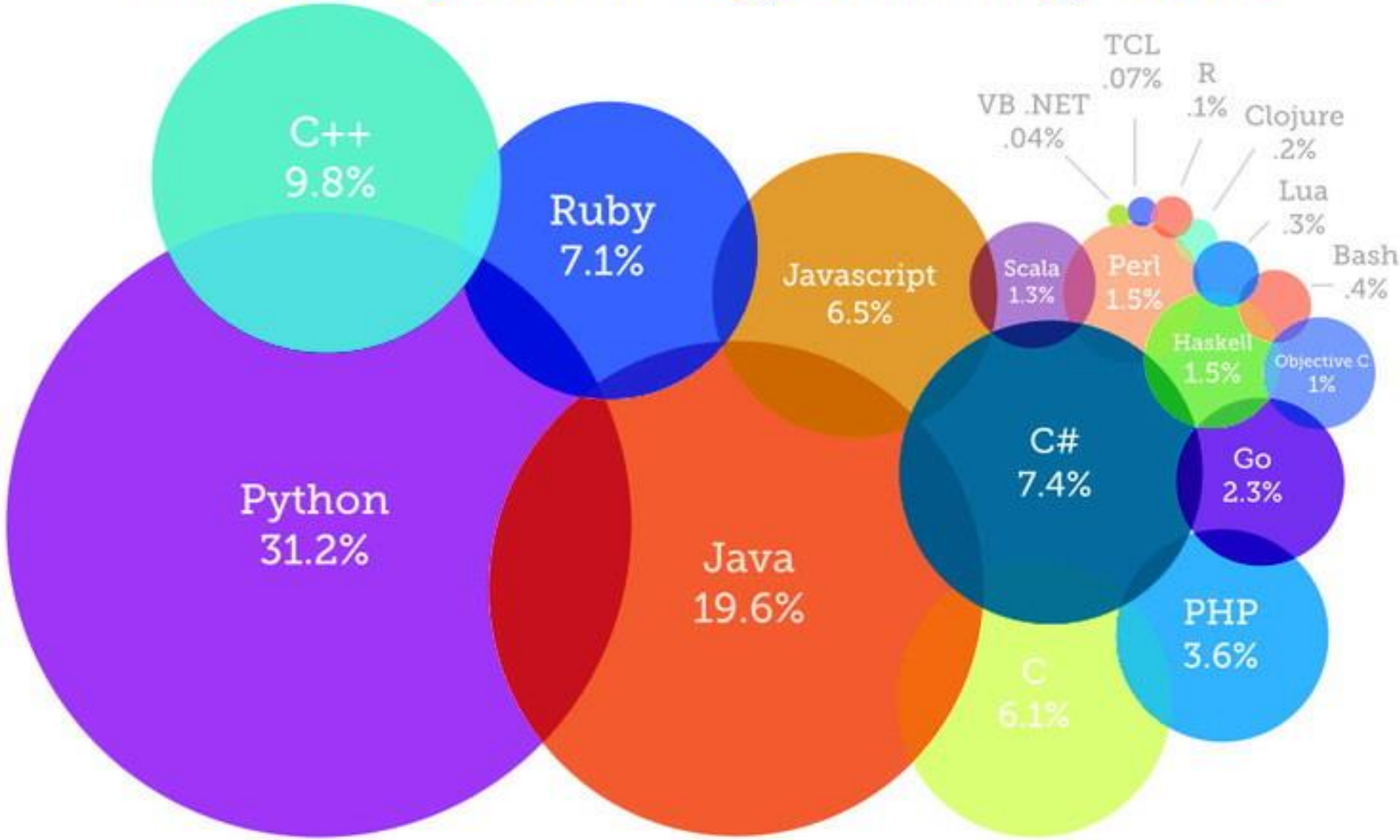
- Pemrograman Windows: Visual Basic, Delphi, Visual Studio
- Pemrograman Web:
  - Markup Language (HTML style)
  - Script style (Javascript, AJAX)
  - Server Programming (CGI, PHP, ASP, Ruby, Phyton)
- Pemrograman Embedded
- Pemrograman Game
- Pemrograman Enterprise
- Pemrograman Mobile: Android, Symbian, Objective-C

TOP LANGUAGE 2016

Web | Mobile | Enterprise | Embedded

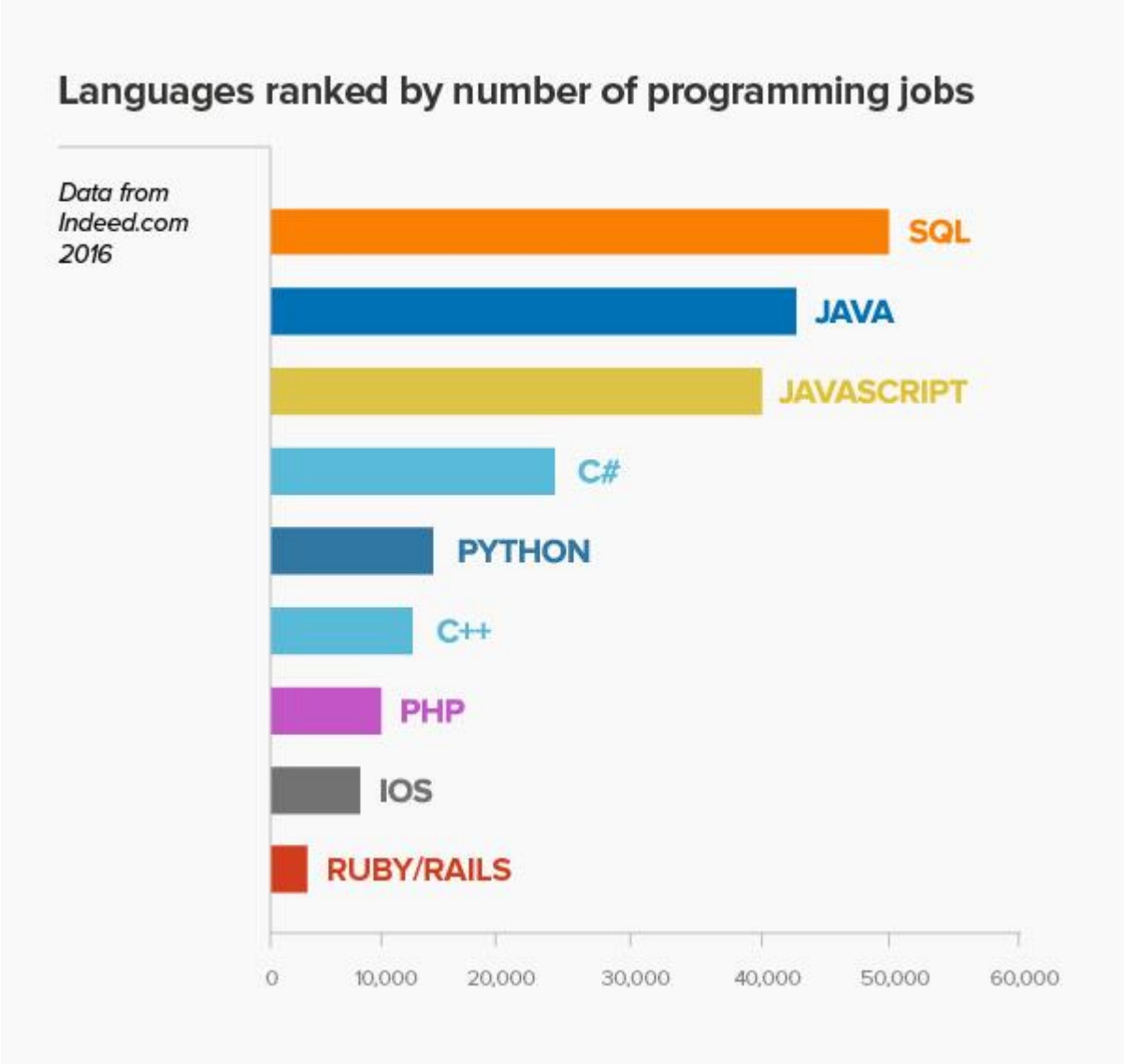


Most Popular Programming 2016



Profesi Programmer

- Desktop Application Programmer
- Software Programmer / Developer
- Web Programmer / Developer
- Mobile Apps Programmer / Developer
- Embedded System Programmer
- Scientist / Researcher
- Network Administrator
- Database Apps Developer
- Etc..

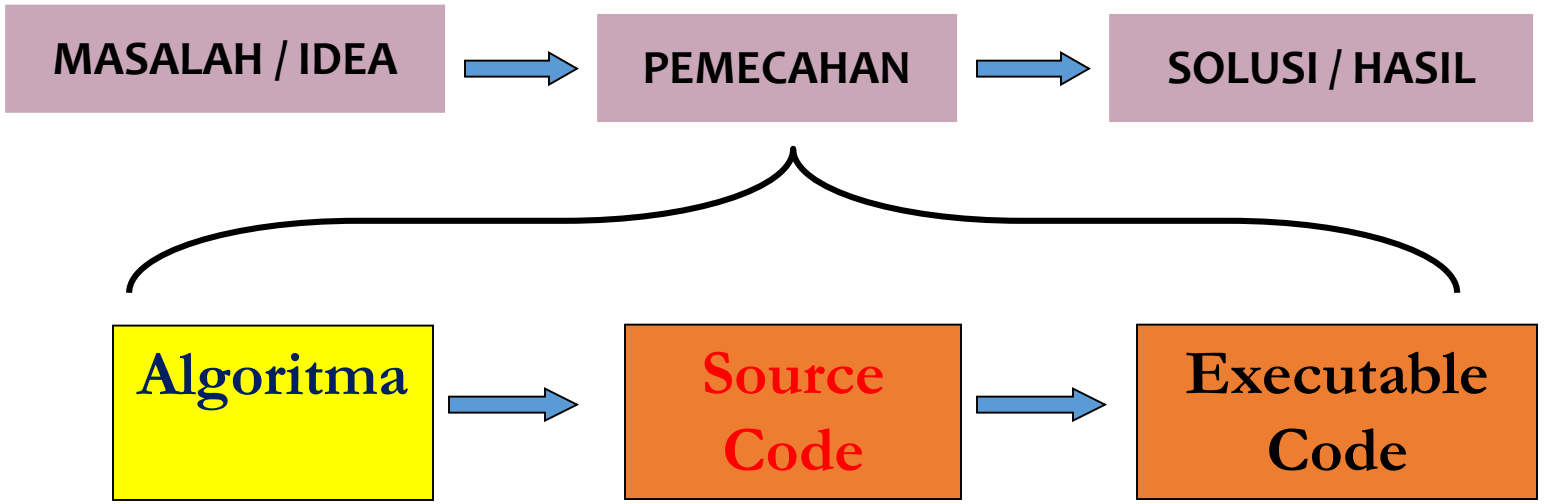


ALGORITMA PEMROGRAMAN

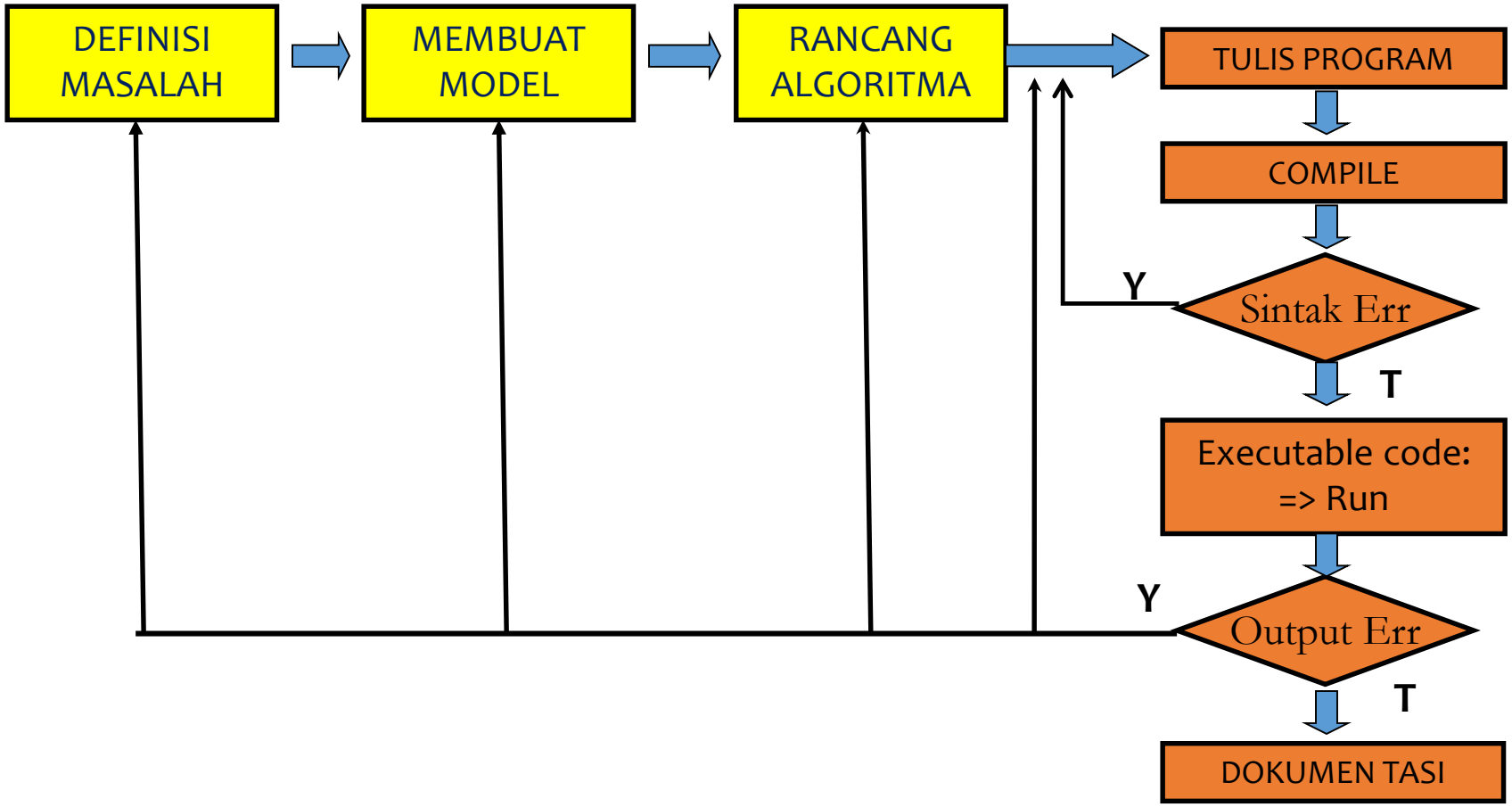
DEFINISI ALGORITMA

- Algoritma adalah **sekumpulan langkah-langkah terbatas untuk mencari solusi suatu masalah**.
- Berasal dari kata algoris dan ritmis. Awalnya diungkapkan oleh Al Khowarizmi.
- Di pemrograman, algoritma didefinisikan sebagai metode yang terdiri dari langkah-langkah terstuktur untuk mencari solusi suatu masalah dengan bantuan komputer.
- Belajar Algoritma adalah belajar bagaimana suatu persoalan diselesaikan dengan langkah-langkah berurut

TAHAP PENGEMBANGAN ALGORITMA



TAHAP PENGEMBANGAN ALGORITMA



Kasus: Akar Persamaan Kuadrat

PENYELESAIAN SECARA MATEMATIS

- Persamaan Kuadrat: **$aX^2 + bX + c = 0$**
- Contoh:
  $X^2 + 5X + 6 = 0$ → $(X+x1)(X+x2) = 0$; x1, x2 = ?
- Solusi:
  $X^2 + 5X + 6 = 0$ → a = 1, b = 5, c = 6

  x1 = $(-b + \sqrt{(b^2 - 4.a.c)}) / 2.a = (-5 + \sqrt{5^2 - 4.1.6}) / 2$
  $= (-5 + \sqrt{(25-26)}) / 2 = (-5 + 1) / 2 = -4 / 2 =$ **-2**

  x2 = $(-b - \sqrt{(b^2 - 4.a.c)}) / 2.a = (-5 - \sqrt{5^2 - 4.1.6}) / 2$
  $= = (-5 - 1) / 2 = -6 / 2 =$ **- 3**
- Akar dari **$X^2 + 5X + 6 = 0$ adalah**
  $(X+2)(X+3) = 0$; x1 = -2, x2 = -3

Kasus: Akar Persamaan Kuadrat

PENYELESAIAN DENGAN KOMPUTER

- Membuat sebuah program komputer yang dapat **menghitung secara otomatis** akar dari suatu persamaan kuadrat, dimana **pengguna hanya menginputkan** nilai (a, b, c) dari bentuk persamaan kuadratnya.
- Program komputer adalah bersifat **generic** dan **umum**, dibuat untuk menyelesaikan semua kasus persamaan kuadrat dengan nilai a, b, c apapun yang memenuhi **$aX^2 + bX + c = 0$**
- Programmer **memberikan perintah** dalam bentuk **urutan-urutan langkah** yang harus dikerjakan oleh komputer, perintah-perintah tersebut **dalam bentuk Kode Program**
- Sekali komputer telah diprogram, maka proses tsb dapat dikerjakan sendiri oleh komputer

Kasus: Akar Persamaan Kuadrat

PENYELESAIAN DENGAN KOMPUTER

- **Langkah-langkah yang dilakukan Programmer:**

1. Membuat **Pemodelan** Matematis
2. Membuat **Algoritma**, urutan langkah penyelesaian
3. Membuat **Kode** Program
4. **Menjalankan** / Menguji Program
5. **Mendokumentasikan** Program jika sudah berhasil

Kasus: Akar Persamaan Kuadrat

PENYELESAIAN DENGAN KOMPUTER

Langkah 1: Pemodelan Matematis

- Persamaan Kuadrat: **$aX^2 + bX + c = 0$**
- Langkah Penyelesaian:

1. Tentukan **Output**: *Output: x1, x2*
2. Buat **Proses** untuk menghitung output dengan menggunakan nilai-nilai yang sudah diketahui (*input*)
   Proses: **x1** = (-b + √ (b^2–4.a.c)) / 2.a
   x2 = (-b - √ (b^2–4.a.c)) / 2.a
   Pastikan bahwa semua nilai yang ada pada bagian sebelah kanan dari tanda = (sama dengan) telah diketahui nilainya.
3. Tentukan **Input**: *Input: a, b, c?*

Kasus: Akar Persamaan Kuadrat

PENYELESAIAN DENGAN KOMPUTER

Langkah 2 Membuat Algoritma

- Persamaan Kuadrat: **$aX^2 + bX + c = 0$**
- Langkah Penyelesaian:

1. ***Input:*** *a, b, c?*
2. ***Proses***:
   x1 = (-b + √ (b^2–4.a.c)) / 2.a
   x2 = (-b - √ (b^2–4.a.c)) / 2.a
3. **Output**: *x1, x2*

Algoritma secara umum harus dalam urutan **IPO**

INPUT-PROSES-OUTPUT

Ingat Algoritma Membuat Kopi!

Algoritma Persamaan Kuadrat

```
Mulai

 Input a?
 Input b?
 Input c?

 x1 = (-b + √ (b²–4.a.c)) / 2.a
 x2 = (-b - √ (b²–4.a.c)) / 2.a

 Cetak hasil x1
 Cetak hasil x2

Selesai
```

Kasus: Akar Persamaan Kuadrat

PENYELESAIAN DENGAN KOMPUTER

Langkah 3 Membuat Program Komputer

```
class PersamaanKuadrat {
public static void main (String arg[])
{
  System.out.print("Nilai A?");
  int A = input.nextInt();
  System.out.print("Nilai B?");
  int B = input.nextInt();
  System.out.print("Nilai C?");
  int C = input.nextInt();

  double X1, X2;
  X1 = (-B + Math.sqrt(B*B-4*A*C))/(2*A);
  X2 = (-B - Math.sqrt(B*B-4*A*C))/(2*A);

  System.out.println("X1 = " + X1);
  System.out.println("X2 = " + X2);
}}
```

Algoritma Persamaan Kuadrat

```
Mulai

 Input a?
 Input b?
 Input c?

 x1 = (-b + √ (b²–4.a.c))/ 2.a
 x2 = (-b - √ (b²–4.a.c))/ 2.a

 Cetak hasil x1
 Cetak hasil x2

Selesai
```

Kasus: Akar Persamaan Kuadrat

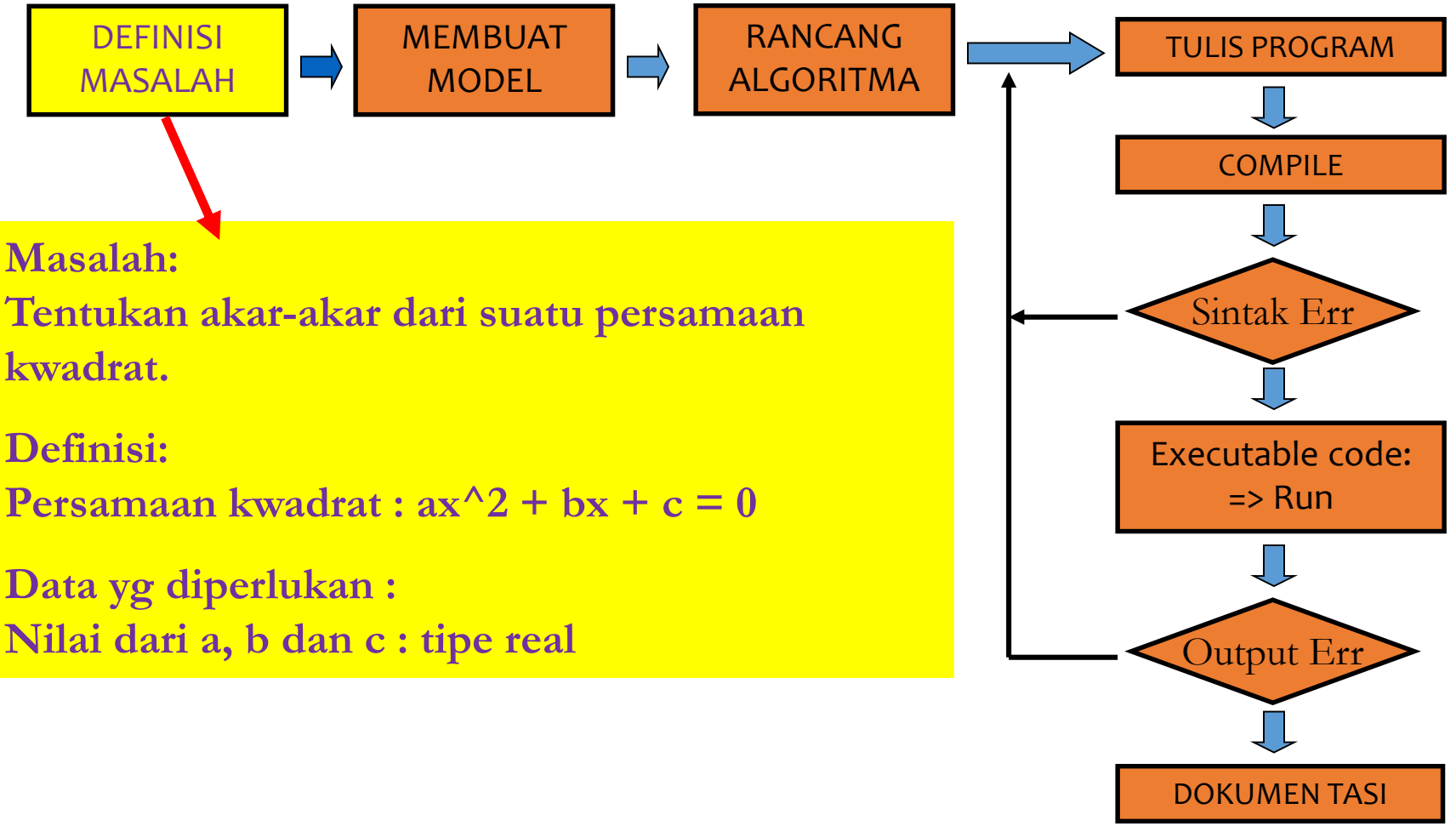
PENYELESAIAN DENGAN KOMPUTER

Langkah 4 Menjalankan Program Komputer

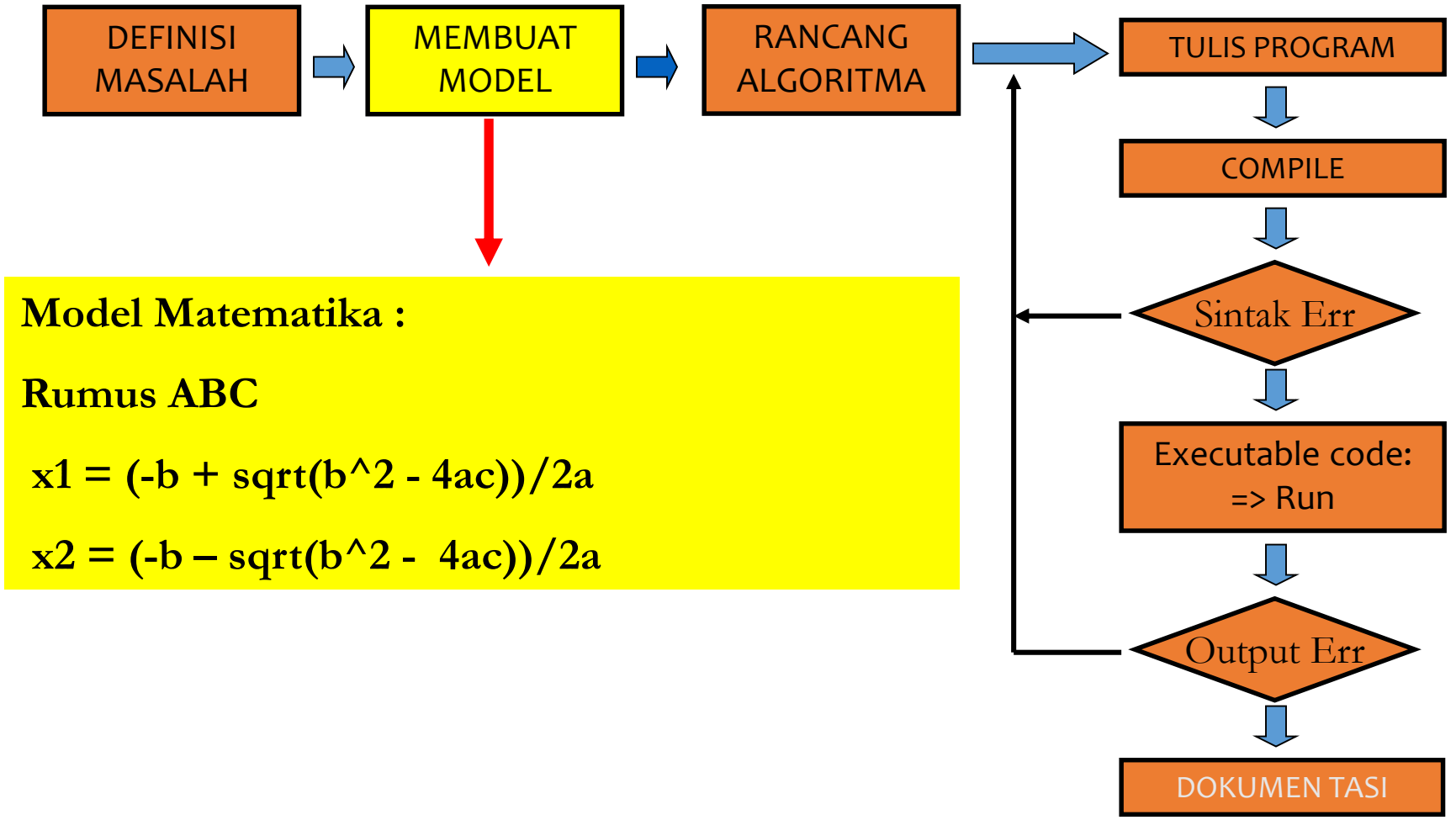
```
class PersamaanKuadrat {
public static void main (String arg[])
{
  System.out.print("Nilai A?");
  int A = input.nextInt();
  System.out.print("Nilai B?");
  int B = input.nextInt();
  System.out.print("Nilai C?");
  int C = input.nextInt();

  double X1, X2;
  X1 = (-B + Math.sqrt(B*B-4*A*C))/(2*A);
  X2 = (-B - Math.sqrt(B*B-4*A*C))/(2*A);

  System.out.println("X1 = " + X1);
  System.out.println("X2 = " + X2);
}}
```

```
Welcome to DrJava.  Working
> run PersamaanKuadrat
Nilai A?
1
Nilai B?
5
Nilai C?
6
X1 = -2.0
X2 = -3.0
```

TAHAP PENGEMBANGAN ALGORITMA



TAHAP PENGEMBANGAN ALGORITMA



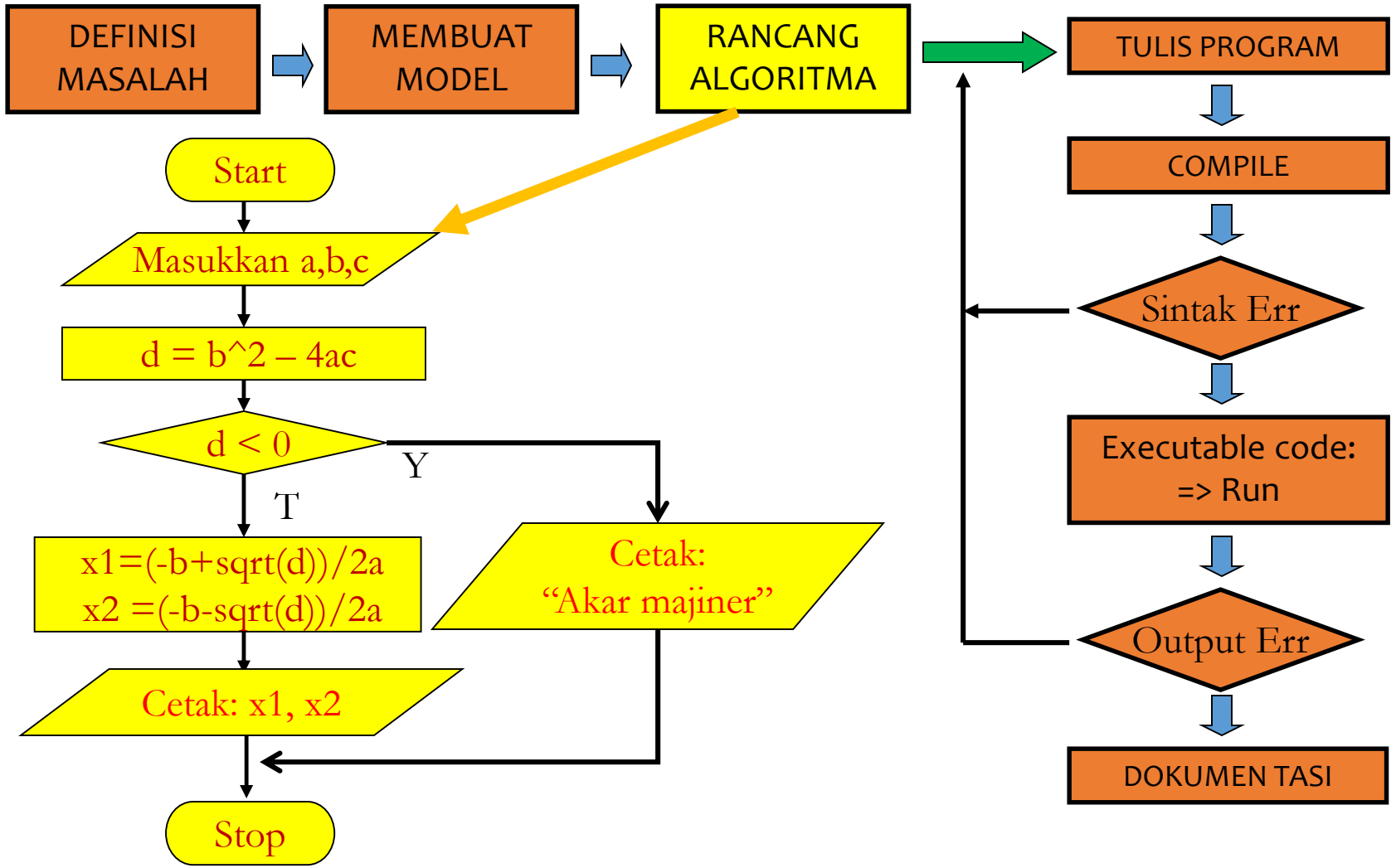
Model Matematika :

Rumus ABC

x1 = (-b + sqrt(b^2 - 4ac))/2a

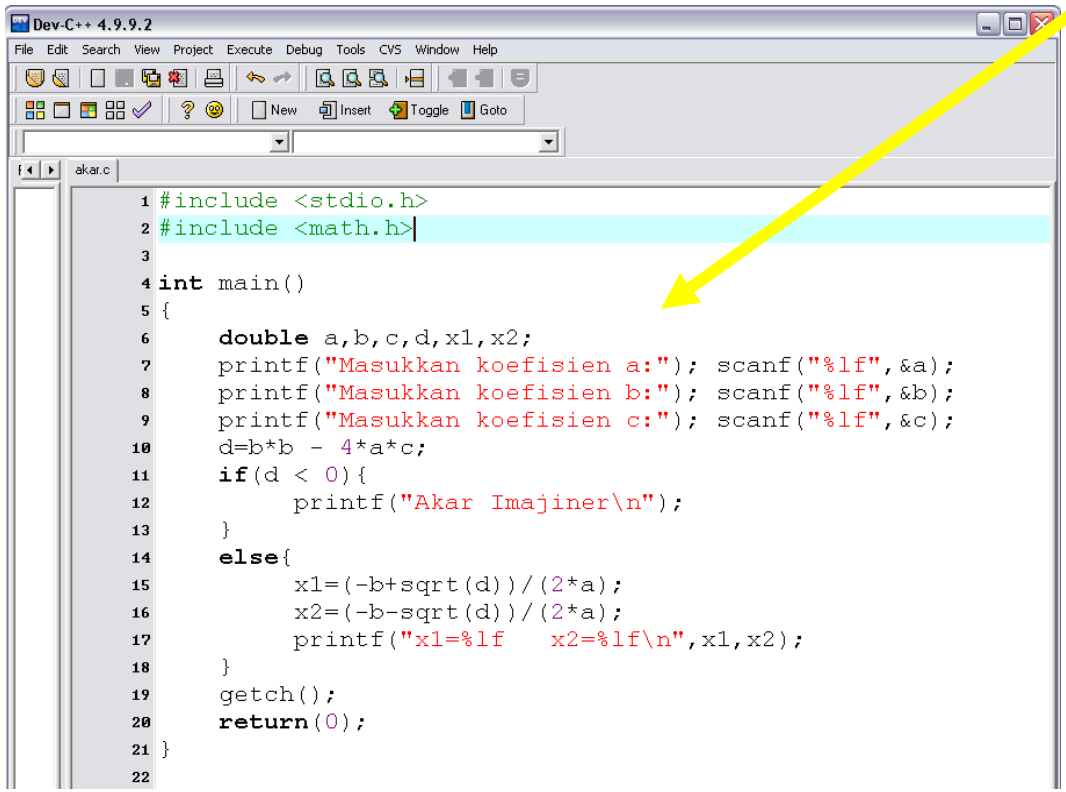
x2 = (-b – sqrt(b^2 - 4ac))/2a

TAHAP PENGEMBANGAN ALGORITMA



TAHAP PENGEMBANGAN ALGORITMA

DEFINISI MASALAH → MEMBUAT MODEL → RANCANG ALGORITMA → TULIS PROGRAM



COMPILE

Sintak Err

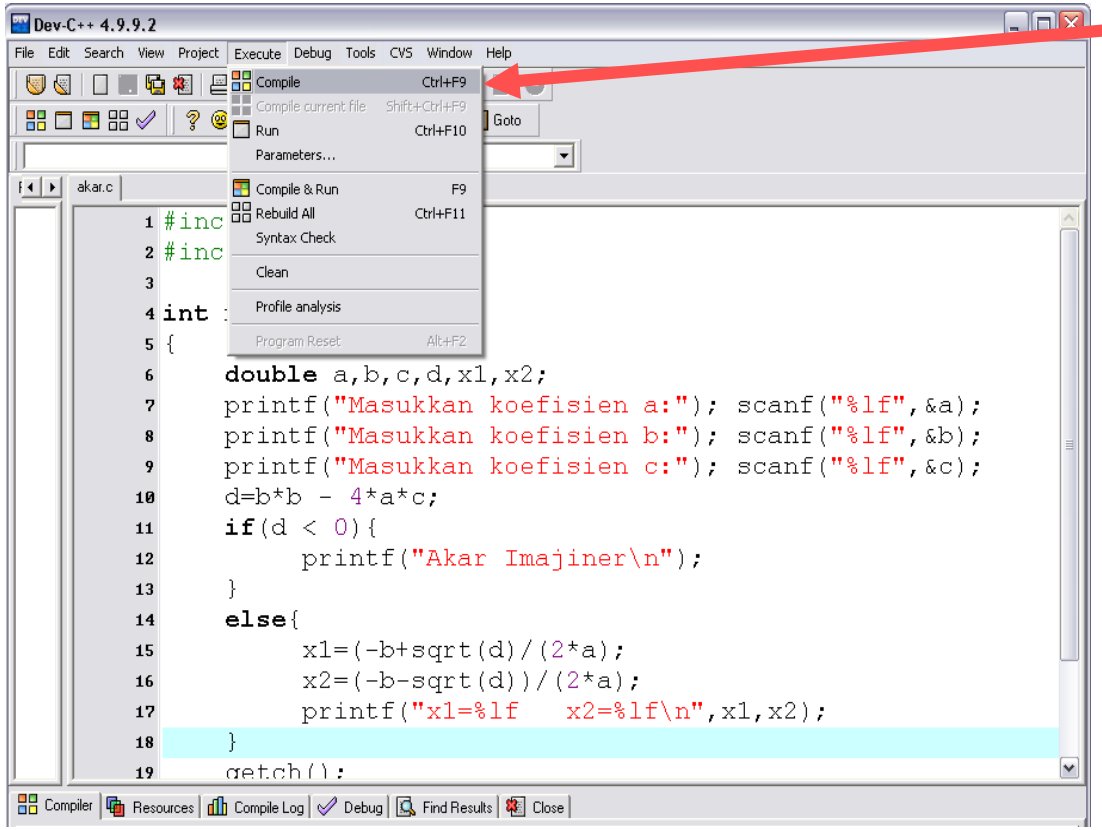
Executable code: => Run

Output Err

DOKUMEN TASI

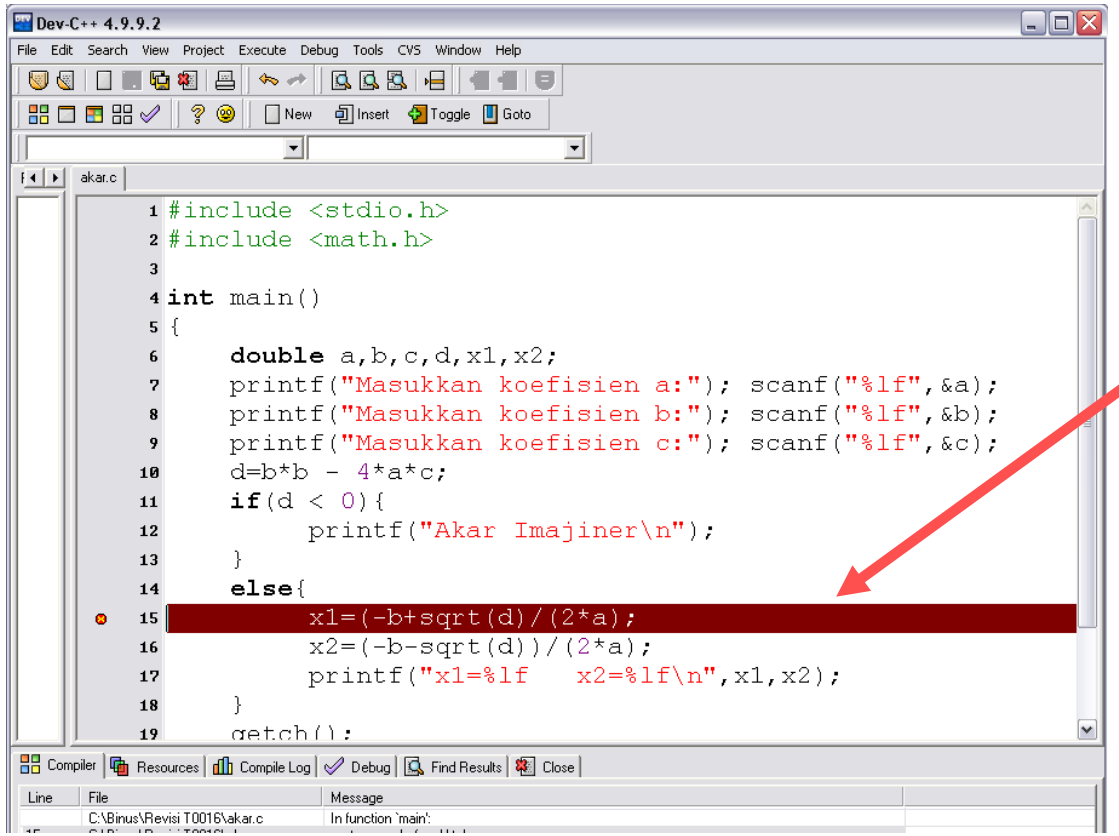
TAHAP PENGEMBANGAN ALGORITMA

DEFINISI MASALAH → MEMBUAT MODEL → RANCANG ALGORITMA → TULIS PROGRAM → COMPILE → Sintak Err → Executable code: => Run → Output Err → DOKUMEN TASI



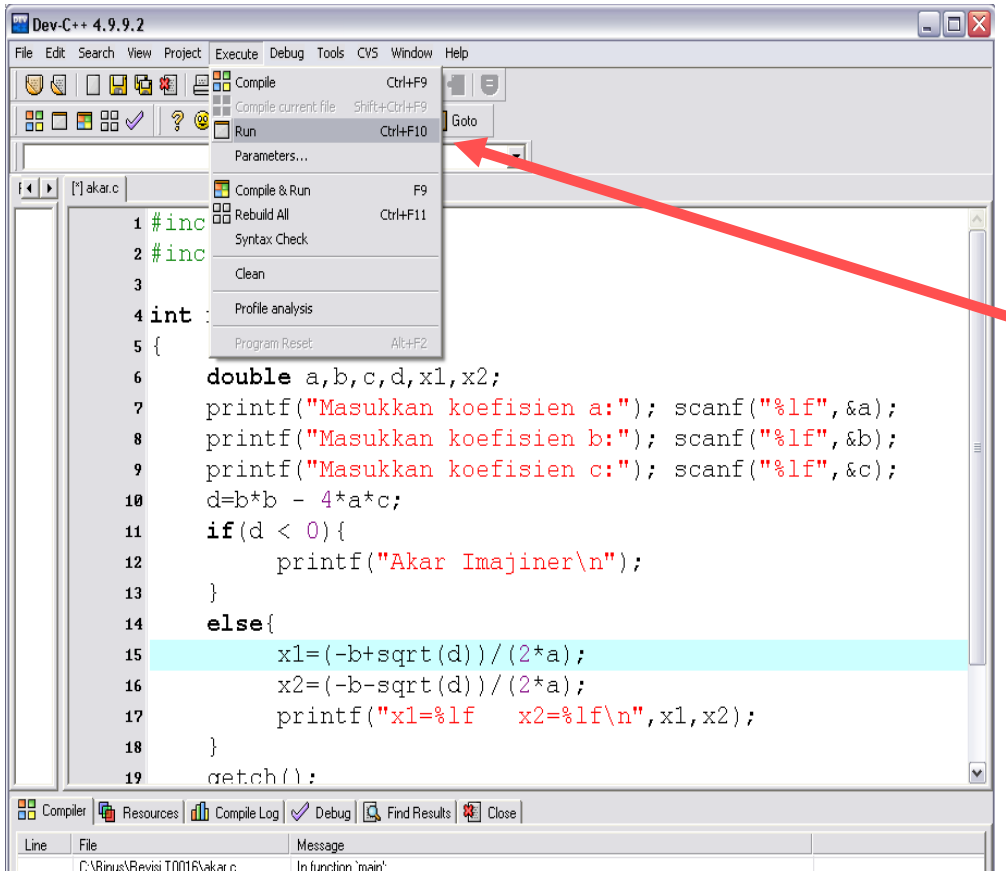
TAHAP PENGEMBANGAN ALGORITMA

DEFINISI MASALAH → MEMBUAT MODEL → RANCANG ALGORITMA → TULIS PROGRAM → COMPILE → Sintak Err → Executable code: => Run → Output Err → DOKUMEN TASI

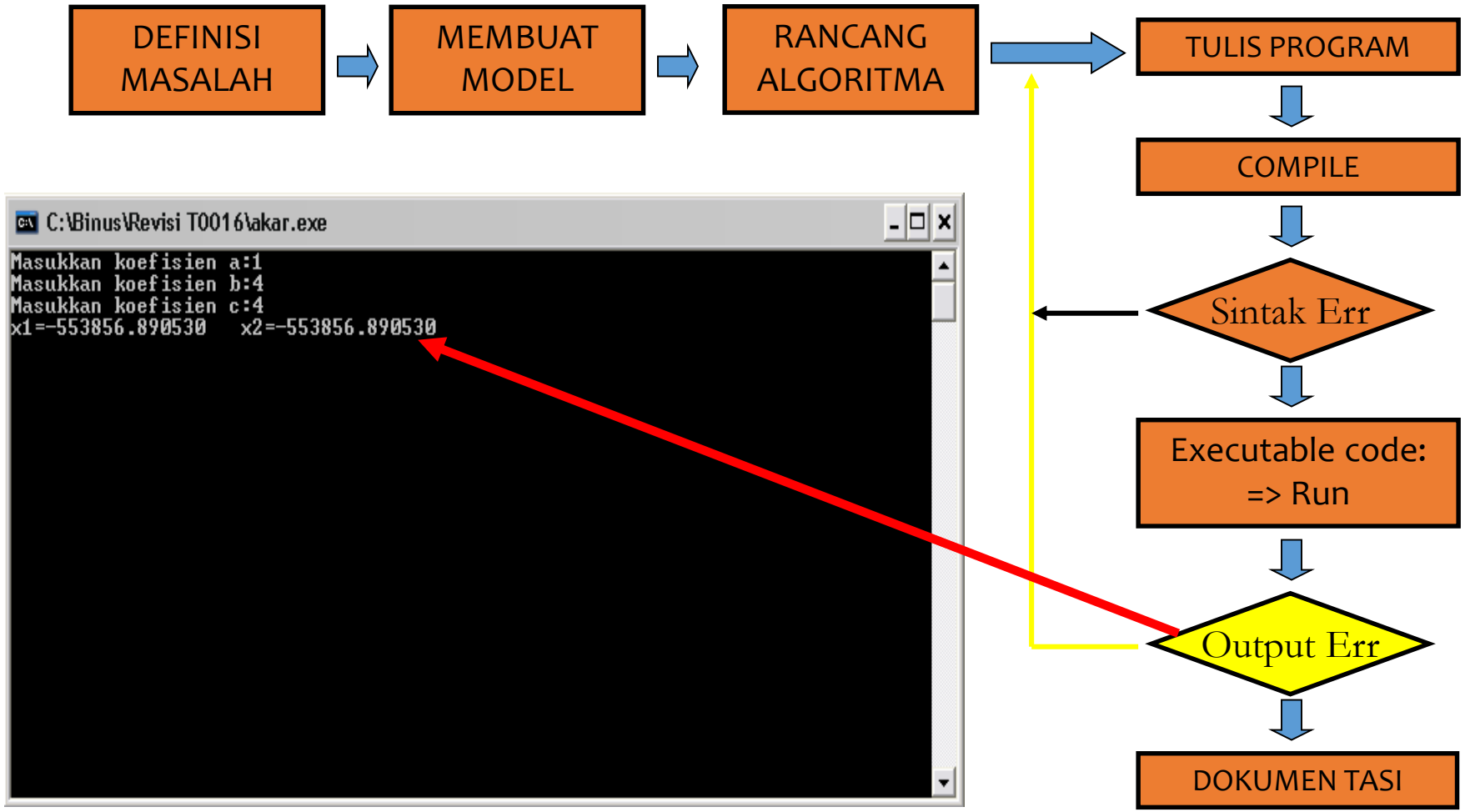


TAHAP PENGEMBANGAN ALGORITMA

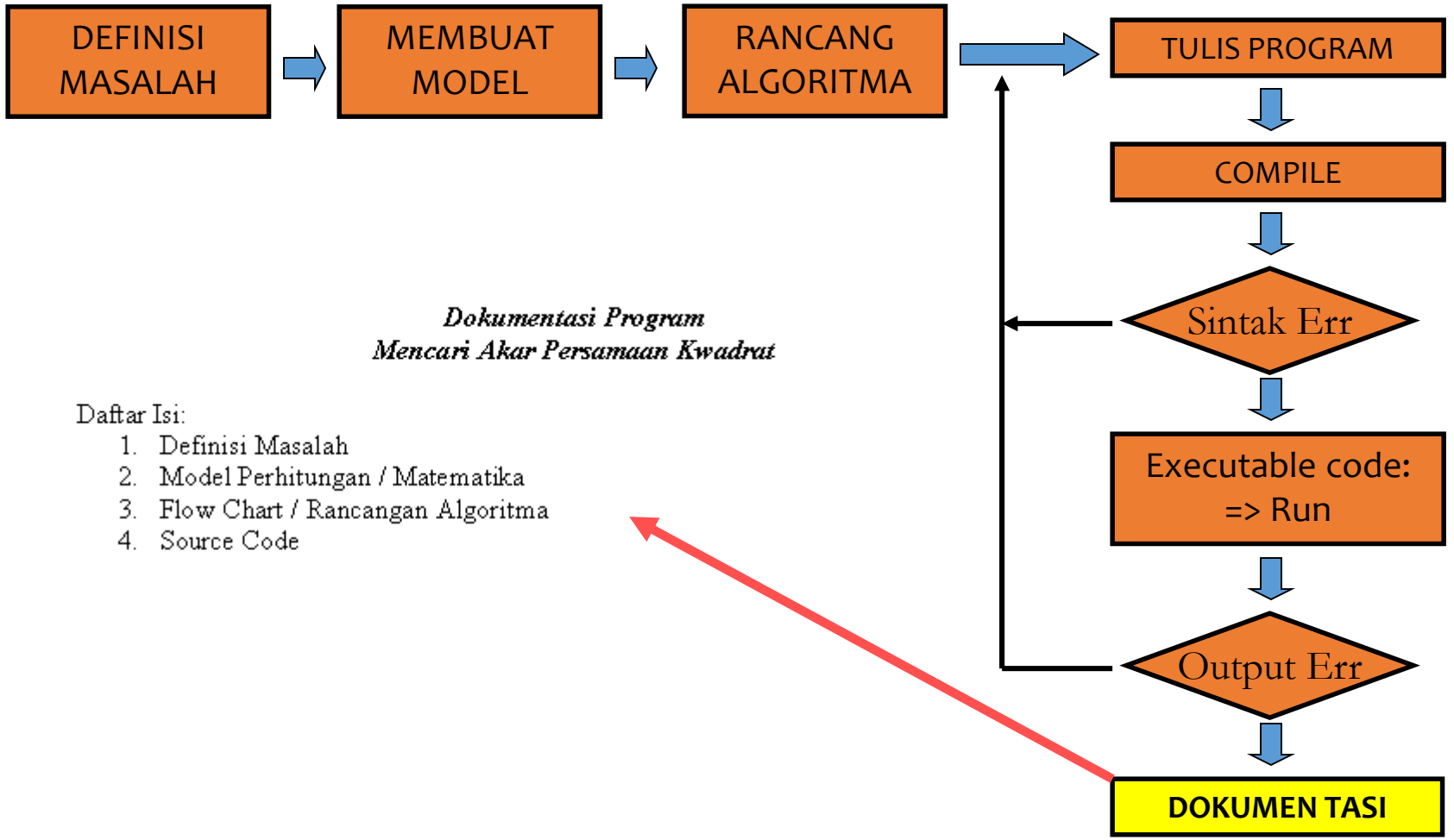
DEFINISI MASALAH → MEMBUAT MODEL → RANCANG ALGORITMA → TULIS PROGRAM → COMPILE → Sintak Err → Executable code: => Run → Output Err → DOKUMEN TASI



TAHAP PENGEMBANGAN ALGORITMA



TAHAP PENGEMBANGAN ALGORITMA



DASAR PEMROGRAMAN TERSTRUKTUR

Tujuh operasi dasar komputer:

1. Membaca data (**Input**)
2. Menampilkan data (**Output**)
3. Melakukan perhitungan aritmetika (**Compute**)
4. Memberikan nilai ke suatu identifier (**Store**)
5. Membandingkan dan Memilih (**Compare / Selection**)
6. Melakukan pengulangan (**Loop / Repetition**)
7. Menyimpan data dalam variabel larik **(Array)**
8. Mengerjakan Fungsi atau Tugas tertentu **(Function / Method)**

PENYAJIAN ALGORITMA

- Algoritma bisa dibuat dengan:
  - **Teknik Tulisan** seperti : Structure english dan ***Pseudocode***.
  - **Teknik Visual** seperti : ***Flow chart***.

PSEUDOCODE

- *Outline* dari sebuah program komputer
- Ditulis dalam bahasa Inggris atau Indonesia sederhana
- Kata kunci (*keyword*) digunakan untuk menjelaskan **struktur kendali** (misalnya: “jika”, “ulangi”, “sampai”,”if”,”repeat”, “until”)

CONTOH ALGORITMA DGN PSEUDOCODE

Algoritma Menggunakan Kalkulator

Mulai

Nyalakan kalkulator
Kosongkan Kalkulator
Ulangi
    Input harga
    Tekan tombol Plus (+)
Sampai semua harga diinput
Tampilkan total harga
Matikan kalkulator

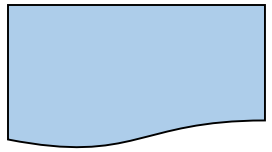
Selesai

FLOW CHART

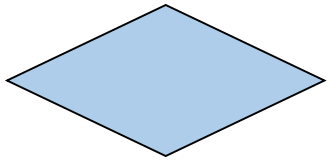
Terminator

Proses

Input/Output



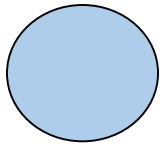
Dokumen



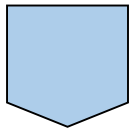
Pemilihan

Pengulangan

Arah



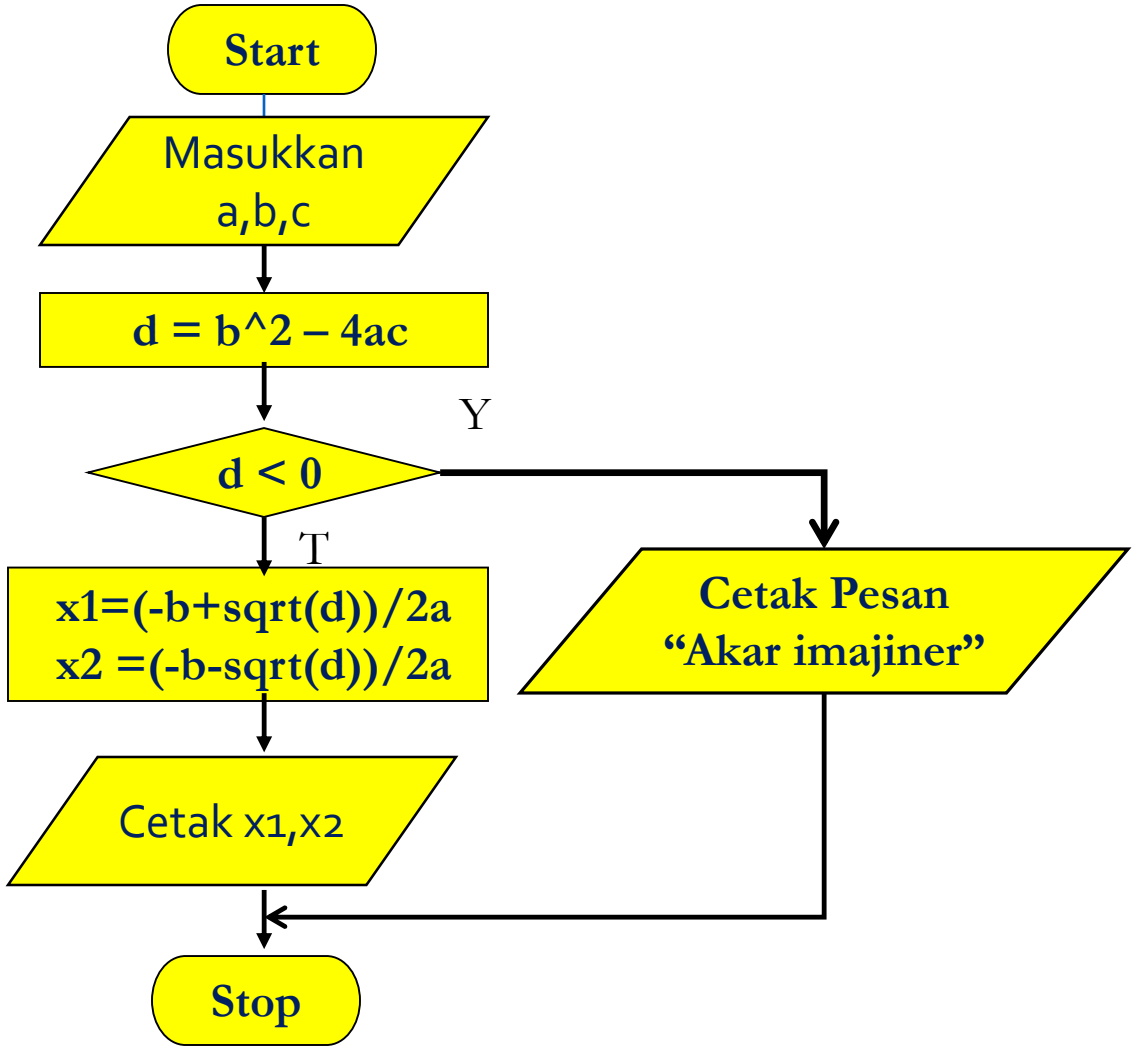
Konektor



Konektor antar halaman

Pemanggilan Procedure

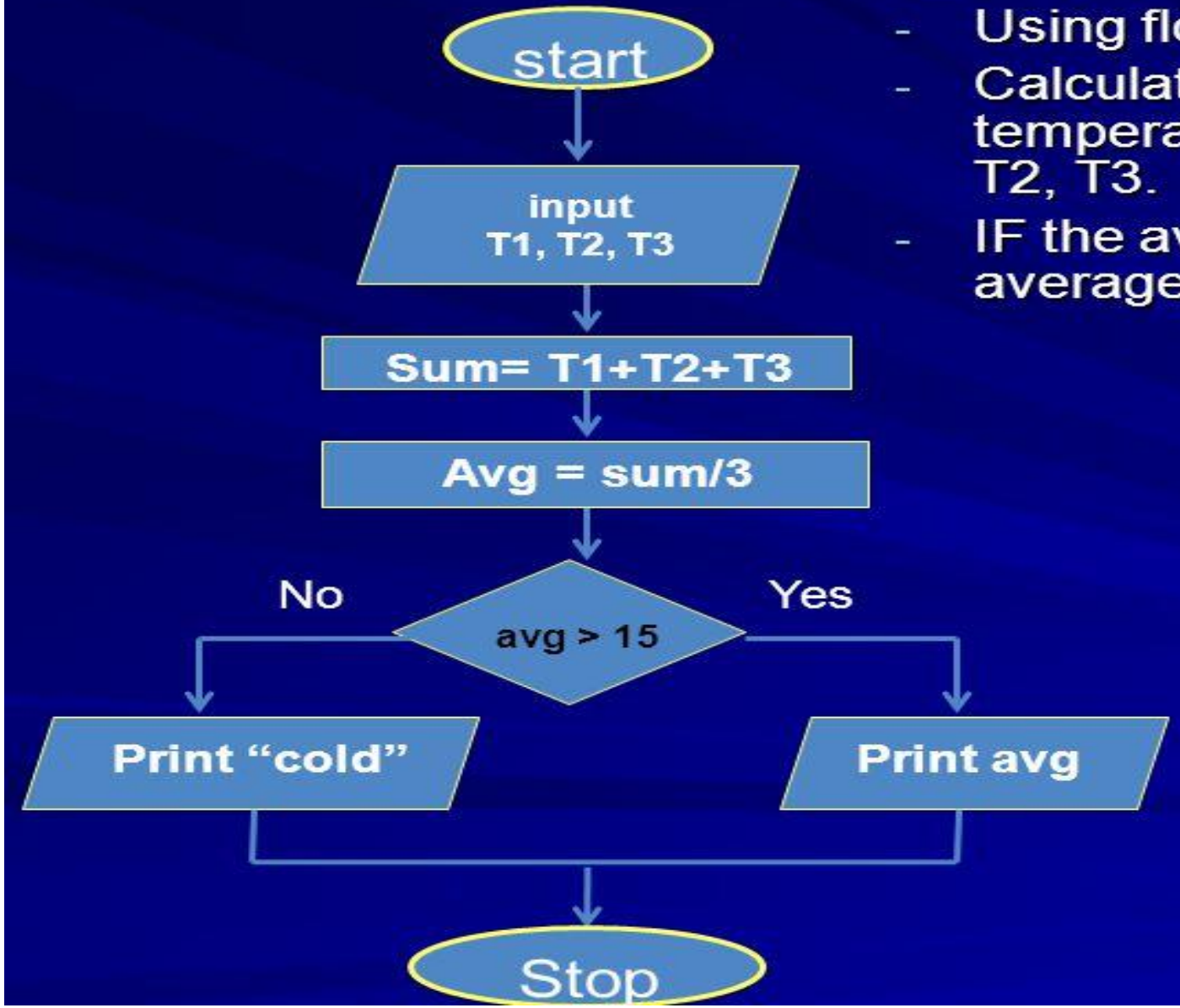
CONTOH FLOW CHART



Flow chart

Example:

- Using flow chart, Write suitable algorithm:
- Calculate and print the average temperature. Readings as the following T1, T2, T3.
- IF the average is greater than 15 print the average, else print the weather is cold.




```
Start
Input  T1,T2,T3

Sum = T1+T2+T3

Avg = sum/3

If avg > 15 then

      Print  avg

   Else

      Print "cold"
  End if
end
```

KRITERIA ALGORITMA YANG BAIK

1. Mempunyai **logika yang tepat** untuk memecahkan masalah.
2. Menghasilkan **output yang benar** dalam waktu yang singkat.
3. Ditulis dengan bahasa baku terstruktur sehingga **tidak menimbulkan arti ganda**.
4. Ditulis dengan format baku sehingga **mudah diimplementasikan** kedalam bahasa pemrograman.
5. Semua operasi **didefinisikan dengan jelas** dan berakhir sesudah sejumlah langkah.

PEMROGRAMAN TERSTRUKTUR

- Pemrograman terstruktur merupakan pola penyusunan program komputer hanya dengan menggunakan tiga struktur kontrol yaitu:
  1. Sequence
  2. Selection
  3. Repetition

1. SEQUENCE

- Sequence merupakan urutan pengerjaan dari perintah/*statement* pertama sampai dengan perintah/*statement* terakhir.
- Umumnya bahasa pemrograman mempunyai sequence (urutan pengerjaan dari perintah / *statement*) mulai dari atas ke bawah dan dari kiri ke kanan.
- Top-down – dari atas ke bawah - mengalir

CONTOH SEQUENCE

- Contoh:

```
Cetak "Jumlah Mahasiswa"
Set Jumlah to 49
Cetak "Tambahan mahasiswa baru"
Baca mhs_baru
Jumlah = Jumlah + mhs_baru
Cetak "Jumlah Mahasiswa"
Cetak jumlah
```

- Penjelasan
  - Urutan pengerjaan adalah mulai dari urutan pertama sampai dengan urutan terakhir, jika mhs_baru diisi dengan 2, maka jumlah yang tercetak adalah 51

2. SELECTION

- Struktur Kontrol Selection adalah penggambaran sebuah kondisi dan pilihan diantara dua aksi.
- Statement Pertama akan dikerjakan jika kondisi bernilai benar, jika tidak maka akan mengerjakan perintah setelah keyword “else” (jika ada).

Contoh Selection

- Contoh :

```
IF Hari=1 THEN
        Cetak "Senin"
    ELSE
        Cetak "Bukan hari Senin"
ENDIF
```

- Penjelasan
  - Tulisan "Senin" akan ditampilkan jika Hari bernilai 1, jika tidak maka tulisan "Bukan hari Senin" yang akan ditampilkan

3. REPETITION

- Beberapa *statement* / perintah dapat diulang dengan menggunakan struktur kontrol *repetition*.
- *Statement* / perintah akan tetap diulang selama kondisi perulangan memenuhi (jika menggunakan DOWHILE – ENDDO)

Contoh Repetition

- Contoh:

```
Bintang = 0
DOWHILE bintang < 5
   Cetak bintang
   bintang = bintang + 1
ENDDO
```

- Penjelasan:
  - Pertama kali bintang akan diisi dengan 0, setelah itu isi dari bintang akan dicetak sebanyak lima kali, sehingga tampilannya akan sebagai berikut:

    0 1 2 3 4

ALGORITMA

- Berbagai algoritma kemudian diciptakan untuk menyelesaikan berbagai persoalan komputasi, misalnya:
- Algoritma mengurutkan data **(Sorting)**
- Algoritma mencari data **(Searching)**
- Algoritma mengenali sesuatu **(Pattern Recognition)**
  - Mengelompokkan dalam kelas tertentu (Klasifikasi)
  - Mengelompokkan dalam cluster tertentu (Klastering)
- Algoritma Mencari Jarak Terdekat **(Graph)**
- **Belajar secara visual:** visualgo.net

BE A GOOD PROGRAMMER!

- Algoritma yang efektif dan efisien
- Bekerja dengan efisien
- Menyajikan dengan style

PROGRAMMING is FUN

CODE WITH STYLE

GOOD PROGRAMMING STYLE

- Gunakan komentar (comments) pada setiap file yang dibuat

```
/* ================================
Nama File   : Perkalian.java
Versi       : 1.1
Fungsi      : Untuk menghitung Perkalian
Programmer  : Nama (NIM) (Kelas)
Last Update : 2-Maret-2016
Update      :
  - V1.0 Membuat class perkalian
  - V1.1 merubah nama variabel
================================== */
```

GOOD PROGRAMMING STYLE

- Gunakan komentar (comments) pada bagian tertentu dari baris program yang dianggap perlu

```
// fungsi untuk menghitung luas lingkaran
public int LuasLingkaran(int R) // R variabel input
{
   return pi * R * R;
}
```

GOOD PROGRAMMING STYLE

- Gunakan nama identifier (nama kelas, nama variabel, nama method) dengan nama yang mempunyai makna dan kontekstual (sesuai dengan kegunaannya)

```
int Panjang;
double LuasLingkaran;
void CetakInfo();
class ContohMobilBeraksi
```

PROGRAMMING TIPS

- Simpan setiap file dalam folder yang mudah ditelusuri, diingat nama dan lokasinya
- Perbanyak latihan soal, RUN -> lihat apa yang terjadi, dan pelajari bagaimana cara kerja contoh kode tersebut
- Catat kejadian khusus yg belum bisa ditemukan solusinya, cari jawaban melalui internet atau tanya ke senior, asisten atau dosen

PENUTUP

- Belajar pemrogaraman (dasar) adalah belajar logika bagaimana suatu persoalan dapat diselesaikan dengan perintah-perintah program komputer yang terstruktur
- Kuasai konsep dan logika pemrograman, kuasai 1 Bahasa pemrograman → belajar Bahasa Pemrograman apapun akan menjadi mudah
- Investasikan waktu untuk terus belajar berbagai kasus algortima pemrograman → Practise, Practise, Practise
- Pengetahuan dan Ketrampilan Pemrograman adalah landasan utama bagi bidang-bidang IT lainnya

ANY QUESTIONS?