

# ALGORITMA DAN PEMROGRAMAN

Mira Suryani, S.Pd., M.Kom | Erick Paulus, S.Si., M.Kom

S-1 Teknik Informatika  
Jatinangor, 13 September 2018



**From West Java for Indonesia to the World through SDGs**

[www.unpad.ac.id](http://www.unpad.ac.id)



# Tujuan

- Setelah mengikuti pokok bahasan ini mahasiswa dapat **memahami definisi, tujuan dan kegunaan** mempelajari **algoritma dan pemrograman**
- Mahasiswa dapat **memahami** tentang **sistem bilangan** decimal, biner, octal dan heksadesimal serta dapat melakukan konversi sistem bilangan dengan benar



# Pokok Bahasan

1. Pendahuluan Sistem Komputer
2. Algoritma
3. Pemrograman
4. Sistem Bilangan dan Konversi



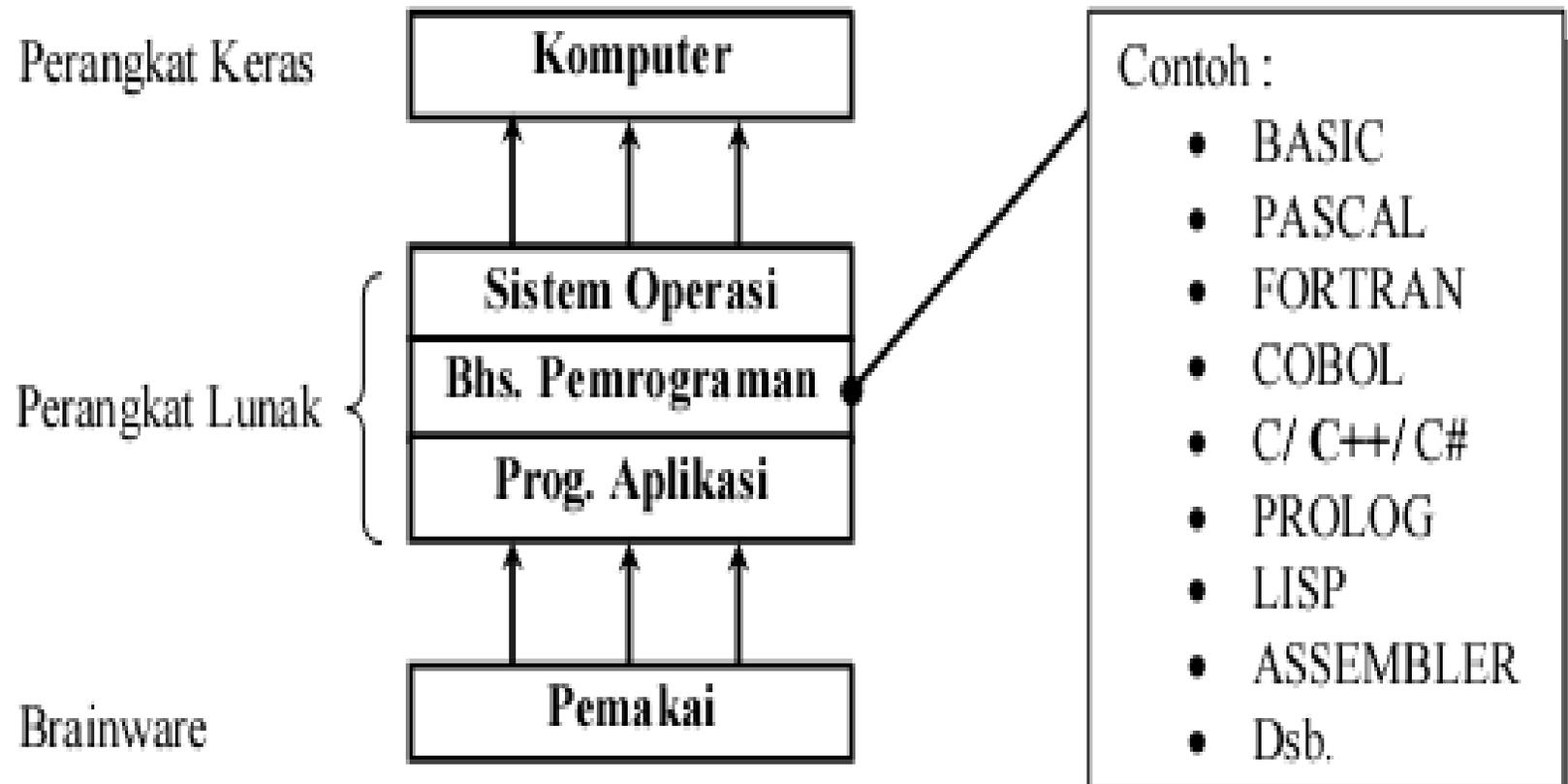
## *Pendahuluan Sistem Komputer*

- Komputer adalah **alat elektronik** untuk **mengolah data** dengan **menggunakan program** tertentu untuk **menghasilkan informasi** sebagai alat bantu **penyelesaian suatu masalah**





# Sistem Komputer



Operator, Programmer system, System Analyst,  
Administrator system, Network Engineer  
Technical Engineer, Web Developer, Animator, dll



# Komponen utama Perangkat Keras komputer

## – Piranti masukan dan Piranti keluaran (*I/O devices*)

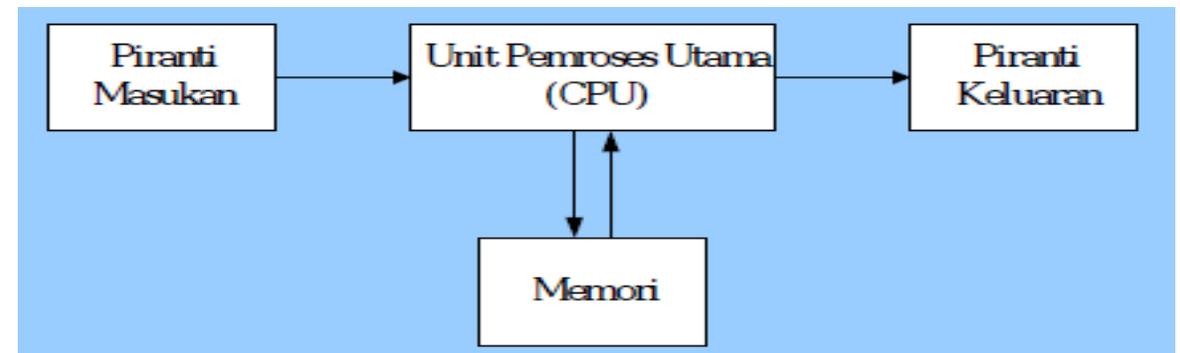
- Peralatan-peralatan yang digunakan untuk memasukkan data atau program ke dalam memori dan peralatan-peralatan yang digunakan untuk menampilkan hasil dari suatu operasi.

## – Unit pemroses utama (CPU = *Central Processing Unit*)

- Unit yang mengerjakan seluruh operasi-operasi dasar seperti operasi perhitungan, operasi perbandingan, operasi membaca dan operasi menulis (Aritmetika dan Logika)

## – Memori

- Komponen yang berfungsi sebagai penyimpan. Yang disimpan di dalam memori adalah program (berisi operasi-operasi yang akan dikerjakan oleh CPU) dan data atau informasi yang telah diolah oleh operasi-operasi





# Macam-macam perangkat lunak (Software)

- **Operating System (Sistem Operasi)**, adalah program komputer yang diperlukan untuk mengatur semua yang kegiatan sistem komputer., sejak komputer mulai dihidupkan hingga komputer siap dimatikan. Sistem Operasi mengendalikan penggunaan semua perangkat yang dipasang di komputer. Contoh *operating system* adalah UNIX, LINUX, Windows, MacOS, Android, dll
- **Language Software (Perangkat Lunak Bahasa)**, adalah bahasa yang digunakan untuk membuat program komputer. Karena digunakan untuk membuat program komputer, maka perangkat lunak bahasa komputer sering disebut juga dengan bahasa pemrograman (*programming language*).
- **Application Software (Program Aplikasi)**, yaitu program yang diterapkan pada suatu aplikasi tertentu. Komputer diciptakan untuk memenuhi beberapa kebutuhan, sehingga program aplikasi yang dibuat pun bermacam-macam, di antaranya program aplikasi untuk mengolah kata, mengolah data, menggambar, mengaransir musik dan lain sebagainya.



# Kaitan Komputer dan Permasalahan

Problematika → komputer → penyelesaian ( ? )

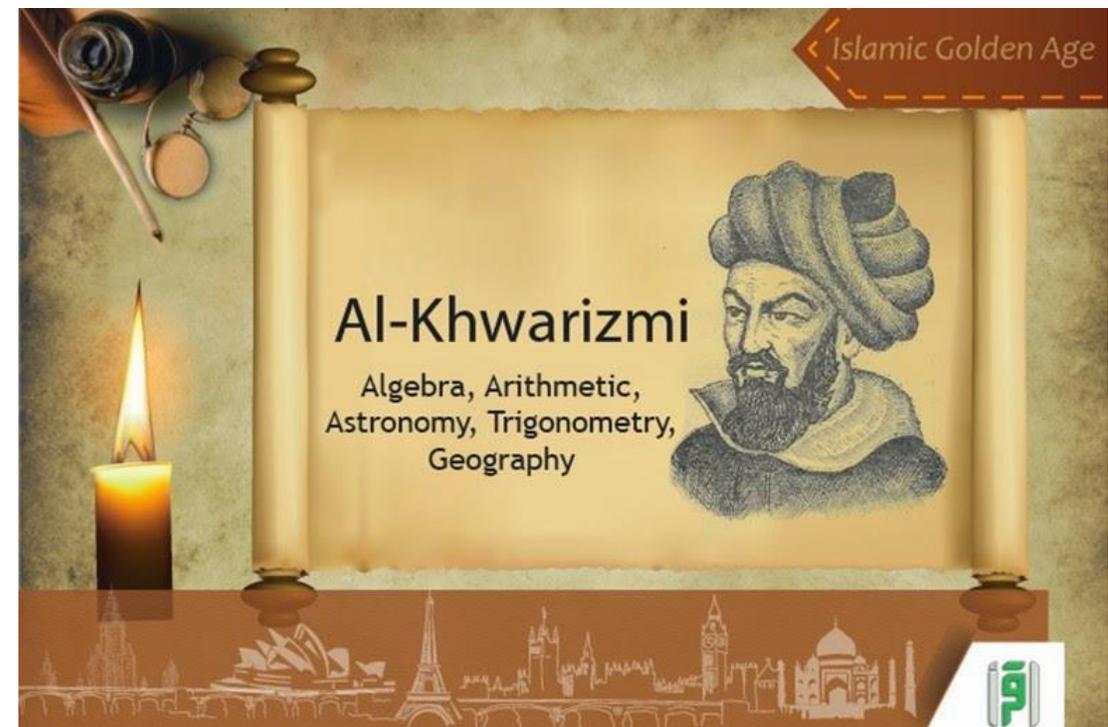
→ harus ditanamkan dalam bentuk program

- Perintah kepada komputer harus sistematis, rinci serta memenuhi kaidah logika tertentu.
- Perintah yang memiliki karakteristik seperti itu dinamakan algoritma.



# Algoritma

- Algoritma berasal dari kata **al-khawarizmi (Abu Ja'far Muhammad Ibnu Musa Al-Khawarizmi)**,
- Merupakan ahli matematika dan astronomi dari Persia, dan juga merupakan penulis buku “Aljabar wal Muqobala”.
- Beliau dianggap sebagai pencetus pertama algoritma karena di dalam buku tersebut Abu Ja'far menjelaskan langkah-langkah dalam menyelesaikan berbagai persoalan aritmatika (aljabar).





# Pengertian Algoritma

- Algoritma biasanya didefinisikan sebagai **rangkaian terurut langkah-langkah yang logis dan sistematis yang disusun untuk menyelesaikan suatu masalah.**
- Algoritma dikatakan sebagai jantung ilmu komputer atau informatika
- Pemakaian : Tidak cuma dalam ilmu komputer tetapi juga dalam bidang lain (kehidupan sehari-hari)



# Tujuan Algoritma

- Memberikan petunjuk tentang **langkah-langkah logika penyelesaian** masalah dalam bentuk yang mudah dipahami nalar manusia sebagai **acuan yang membantu dalam mengembangkan program komputer**.
- Pemahaman terhadap algoritma akan mencegah sejak dini kemungkinan terjadinya kesalahan logika pada program komputer yang dikembangkan.



# Algoritma login ke facebook

- User : Masukkan user name
- User : Masukkan Password
  - Sistem :Cek apakah username terdaftar?
  - Sistem :Jika **tidak terdaftar**, maka sistem akan mengeluarkan peringatan adanya kesalahan. Jika **terdaftar** maka, selanjutnya sistem akan mengecek apakah username dan password cocok (sesuai dengan database)?
  - Sistem :Jika Username dan password **cocok** maka user diperbolehkan masuk ke Facebook ybs. Jika **tidak cocok** maka sistem akan mengeluarkan peringatan adanya kesalahan
- User : Dapat mengakses halaman facebook ybs



Menurut Donald E. Knuth , ada beberapa ciri algoritma

1. Algoritma mempunyai **awal dan akhir**, suatu algoritma harus berhenti setelah mengerjakan serangkaian tugas. Dengan kata lain, suatu algoritma memiliki langkah yang terbatas.
2. Setiap langkah harus didefinisikan dengan tepat, sehingga tidak memiliki arti ganda, **tidak membingungkan (*not ambiguous*)**
3. Memiliki masukan (input) atau **kondisi awal**
4. Memiliki keluaran (output) atau **kondisi akhir**
5. **Algoritma harus efektif**, bila diikuti benar-benar maka akan menyelesaikan masalah.



Oleh sebab itu maka :

- Setiap **instruksi** dalam algoritma harus **jelas dan bermakna unik/tunggal**.
- Seluruh rangkaian instruksi harus **selesai dalam waktu yang berhingga**.
- **Bersifat umum dan universal** artinya dapat dipakai menyelesaikan berbagai masalah yang tipenya serupa tanpa perlu dimodifikasi sekaligus bisa diimplementasikan dengan berbagai macam bahasa pemrograman.



# Contoh lain:

**Tabel 1.1. Contoh-Contoh Algoritma dalam Kehidupan Sehari-hari**

No.	Proses	Algoritma	Contoh Langkah dalam Algoritma
1	Membuat kue	Resep kue	Masukkan telur ke dalam wajan, kocok sampai mengembang
2	Membuat pakaian	Pola pakaian	Gunting kain dari pinggir kiri bawah ke arah kanan sejauh 5 cm
3	Merakit mobil	Panduan merakit	Sambungkan komponen A dengan komponen B
4	Kegiatan sehari-hari	Jadwal harian	Pukul 06.00: mandi pagi, pukul 07.00: berangkat kuliah
5	Mengisi voucer HP	Panduan pengisian	Tekan 888, masukkan nomor voucer

- Berikan contoh yang lain?



# Skema Dasar dalam Algoritma

- **Skema Runtunan (*sequence*)**

- Prosedur proses dalam algoritma yang dilakukan secara terurut langkah demi langkah.
- Sebuah urutan terdiri dari satu atau lebih instruksi. Tiap instruksi dilaksanakan secara berurutan sesuai dengan urutan pelaksanaan, artinya suatu instruksi akan dilaksanakan setelah instruksi sebelumnya telah selesai dilaksanakan.

- **Skema penyeleksian (*selection*)**

- Suatu instruksi dikerjakan jika suatu kondisi tertentu dipenuhi. Dengan adanya proses ini maka ada kemungkinan beberapa jalur aksi yang berbeda berdasarkan kondisi yang ada.

- **Skema Pengulangan (*looping*)**

- Suatu proses melakukan eksekusi suatu program secara berulang-ulang pada suatu blok instruksi tertentu yang terkendali.



# Penulisan (Notasi) algoritma

- Uraian deskriptif
- Diagram-alir (*flowchart*)
- *Pseudocode* (kode semu)



# Uraian Deskriptif

- Dengan notasi bergaya **uraian**, deskripsi setiap langkah dijelaskan dengan bahasa yang gamblang.
- Proses diawali dengan kata kerja seperti 'baca' atau 'membaca', 'hitung' atau 'menghitung', 'bagi' atau 'membagi', 'ganti' atau 'mengganti', dan sebagainya,
- pernyataan kondisional dinyatakan dengan 'jika ... maka ...'.
- Perulangan dengan 'Ulangi .. Selama ...'



Contoh : menyelesaikan permasalahan menghitung luas dan keliling suatu lingkaran

Algoritma Hitung\_Luas\_dan\_Keliling\_Lingkaran

Deskripsi :

- Masukkan jari-jari lingkaran ( $r$ )
- Hitung luas lingkaran dengan rumus  $L = p * r * r$
- Hitung keliling lingkaran dengan rumus  $K = 2 * p * r$
- Tampilkan luas lingkaran
- Tampilkan keliling lingkaran



# Diagram Alir / Flow Chart

- Penulisan algoritma dilakukan dengan menggunakan diagram-diagram.
- Setiap diagram mewakili satu instruksi / perintah tertentu.
- Urutan perintah dalam suatu algoritma digambarkan dengan anak panah ( dari suatu diagram ke diagram lain ).
- Tidak cocok untuk penulisan algoritma yang panjang karena menimbulkan kerumitan.
- Dari segi struktur pemrograman tidak dianjurkan untuk dipakai karena bentuk penulisannya jauh berbeda dengan implementasinya pada bahasa pemrograman tertentu.
- Sejak tahun 1980-an penulisan dengan diagram alir mulai ditinggalkan, kecuali untuk menuliskan langkah-langkah global sebuah algoritma.

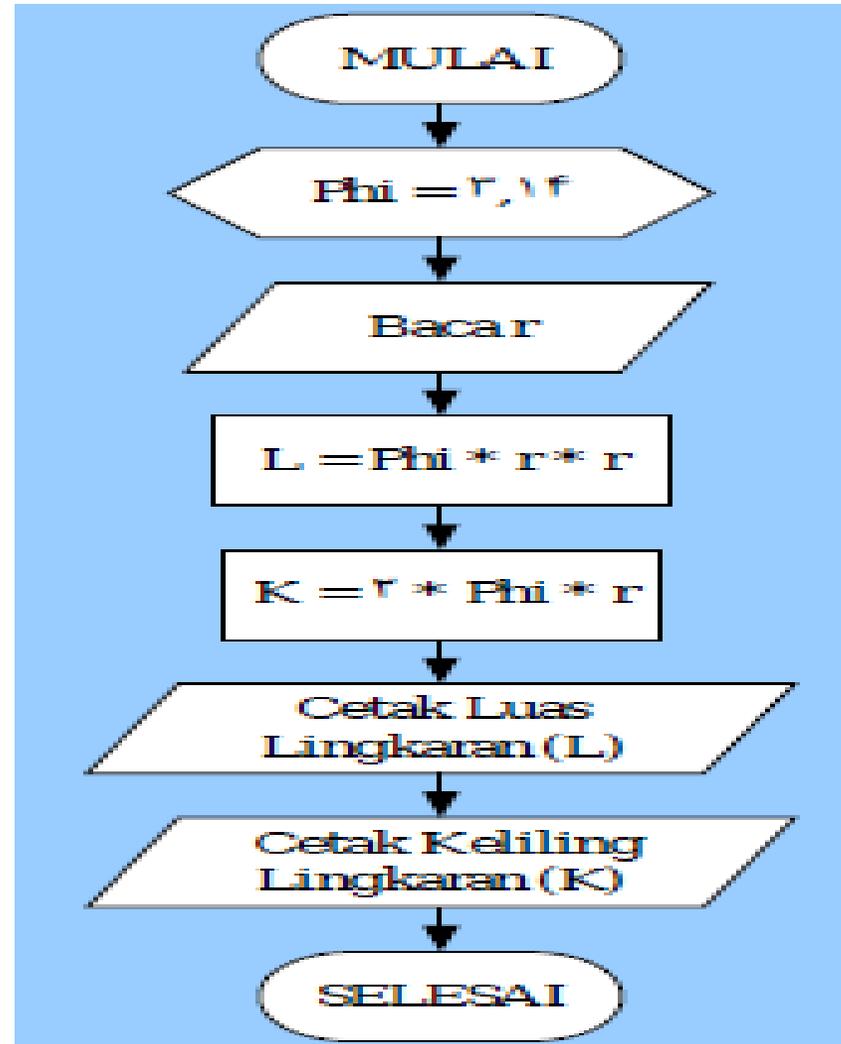


# Notasi Diagram Alir / Flow Chart

SIMBOL	NAMA	FUNGSI
	<b>TERMINATOR</b>	Permulaan/ akhir program
	<b>GARIS ALIR (FLOW LINE)</b>	Arah aliran program
	<b>PREPARATION</b>	Proses inisialisasi/ pemberian harga awal
	<b>PROSES</b>	Proses perhitungan/ proses pengolahan data
	<b>INPUT/ OUTPUT DATA</b>	Proses input/ output data, parameter, informasi
	<b>PREDEFINED PROCESS (SUB PROGRAM)</b>	Permulaan sub program/ proses menjalankan sub program
	<b>DECISION</b>	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	<b>ON PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	<b>OFF PAGE CONNECTOR</b>	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda



Contoh : menghitung luas dan keliling lingkaran





# Kode Semu / Pseudo Code

- Notasi yang **menyerupai notasi Bahasa pemrograman** tingkat tinggi, misalnya Bahasa Pascal dan C.
- Umumnya mempunyai notasi yang hampir mirip untuk beberapa instruksi dalam beberapa bahasa pemrograman, seperti notasi *if-then-else*, *while-do*, *repeat-until*, *read*, *write*, dan sebagainya



# 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** : menghitung luas sebuah luas persegi panjang

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

{Menghitung sebuah luas persegipanjang apabila panjang dan lebar persegipanjang tersebut diberikan}

### **Deklarasi**

{Definisi nama peubah/variabel}  
float panjang, lebar, luas

### **Deskripsi**

```
read(panjang,lebar)    // input / baca  
luas ← panjang * lebar  
write(Luas)           // output / tulis
```



FileView

- persegi

```

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

```

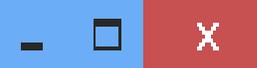
Panjang = 5
Lebar   = 3
Luas    = 15

```

Activate Windows  
Go to PC settings to activate Windows.



# Free Pascal IDE



File Edit Search Run Compile Debug Tools Options Window Help

```
[1] tes.pas 1=[↑]
uses crt;

var
    panjang, lebar, luas : real;

begin
    write('Masukkan panjang : '); readln(panjang);
    write('Masukkan lebar   : '); readln(lebar);

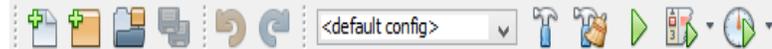
    luas := panjang * lebar;

    writeln('Luas Persegi Panjang : ', luas:7:2);

    repeat until keypressed;
end.
```

F1 Help F2 Save F3 Open Alt+F9 Compi

```
Free Pascal IDE
Free Pascal IDE Version 1.0.12 [2014/03/06]
Compiler Version 2.6.4
GDB Version GDB 7.4
Using configuration files from: C:\NFGC\2.6.4\bin\i386-win32\
Running "c:\fpc\2.6.4\bin\i386-win32\tes.exe"
Masukkan panjang : 5
Masukkan lebar   : 3
Luas Persegi Panjang : 15.00
```



Projects × Files Services

- BagiNol
  - Source Packages
    - baginol
      - Main.java
  - Test Packages
  - Libraries
  - Test Libraries
- CUIPersegi
  - Source Packages
    - cuipersegi
      - CUIPersegi.java
  - Libraries
- kalkulatorGUI
  - Source Packages
    - kalkulatorgui

main - Navigator ×

Members <empty>

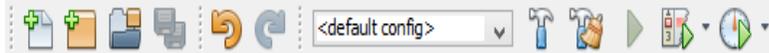
- CUIPersegi
  - main(String[] args)

```
16 * @param args the command line arguments
17 */
18 public static void main(String[] args) {
19     // TODO code application logic here
20     Scanner input=new Scanner (System.in);
21
22     float panjang,lebar,luas;
23
24     System.out.print("Masukkan panjang : ");
25     panjang = input.nextFloat();
26
27     System.out.print("Masukkan lebar  : ");
28     lebar = input.nextFloat();
29
30     luas=panjang*lebar;
31
32     System.out.println("Luas    : "+luas);
33
34
35 }
36 }
```

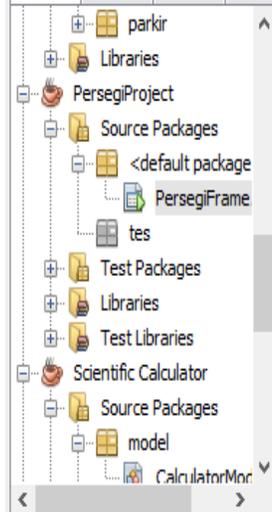
Output - CUIPersegi (run) ×

```
run:
Masukkan panjang : 5
Masukkan lebar  : 3
Luas    : 15.0
BUILD SUCCESSFUL (total time: 8 seconds)
```

Activate Windows  
Go to PC settings to activate Windows.



Start Page x Stack.java x Fkalkulator.java x FormUtama.java x PersegiFrame.java x CUIPersegi.java x Main.java x



```
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() {
```

Output - PersegiProject (run) x

run:

Program Persegi Panjang

Persegi Panjang	
Masukkan Panjang Persegi Panjang :	<input type="text" value="5"/>
Masukkan Lebar Persegi Panjang :	<input type="text" value="3"/>
Luas Persegi Panjang :	<input type="text" value="15.0"/>





## Keuntungan menggunakan notasi *pseudocode*

- Kemudahan mengkonversi atau mentranslasi ke notasi bahasa pemrograman, karena terdapat korespondensi antara setiap *pseudocode* dengan notasi bahasa pemrograman.
- Korespondensi ini dapat diwujudkan dengan tabel translasi dari notasi algoritma ke notasi bahasa pemrograman apapun.

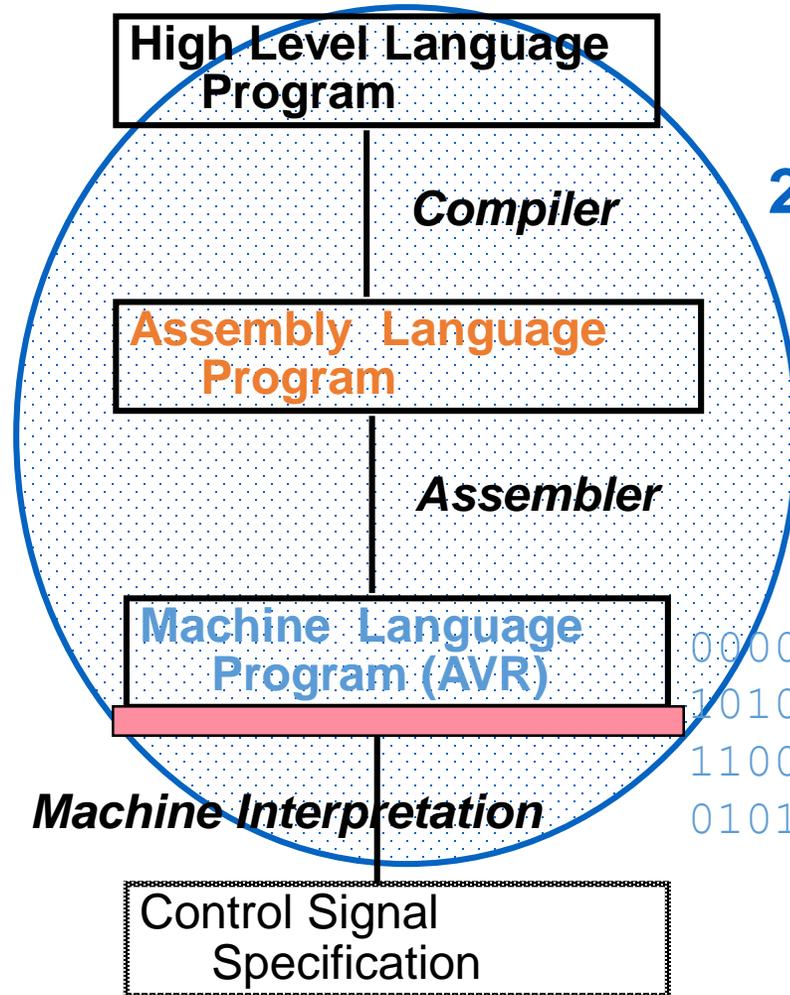


# Pemrograman

- 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 aktivitas yang berhubungan dengan **pembuatan program** dengan mengikuti **kaidah bahasa pemrograman** tertentu.
- Dalam konteks pemrograman terdapat sejumlah bahasa pemrograman seperti Pascal, Delphi, C, C++, C#, C++, dll.



# Tingkat Bahasa Pemrograman (Tinggi dan Rendah)



20210

**A = 25;**

**B = 8;**

**C = A + B;**

lds r1,0x100

lds r2, 0x102

add r1, r2

sts 0x104, r1

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```



# Top 10 Programming Language (IEEE Spectrum Ver)

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

Source: <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>

<https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2017>  **TOP 48 language**



# 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 Objek**
- Pemrograman **Berorientasi Fungsi**
- Pemrograman **Deklaratif**



# Pemrograman **Prosedural / Terstruktur**

- 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 dibagi-bagi menjadi prosedur dan fungsi.
  - Contoh: PASCAL dan C



- Pemrograman **Fungsional**
  - Berdasarkan teori fungsi matematika
  - Fungsi merupakan dasar utama program.
  - Contoh : Lisp, Matlab, R, dll
- Pemrograman **Berorientasi Obyek**
  - Pemrograman berdasarkan prinsip obyek, dimana obyek memiliki data/variabel/property dan method/event/prosedur yang dapat dimanipulasi
  - Contoh: C++, Object Pascal, dan C++.
- 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 dan menguji program (implementasi).**
  - 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	Algoritma	Pascal	C / C++
Assignment / Penugasan	←	:=	=
Masukan	input( )	read() readln()	scanf / cin
Keluaran	output()	write writeln	printf / cout



# *Belajar memprogram TIDAK SAMA dengan 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”).



# Sistem Bilangan

- Bilangan dapat disajikan dalam beberapa cara. Cara penyajiannya tergantung pada Basis(Base) bilangan tersebut.
- Terdapat 4 cara utama dalam penyajian bilangan yaitu :Desimal, Biner, Oktal dan Heksadesimal

Contoh :

<i>Desimal</i>	<i>Biner</i>	<i>Oktal</i>	<i>Heksadesimal</i>
126 <sub>10</sub>	1111110 <sub>2</sub>	176 <sub>8</sub>	7E <sub>16</sub>
11 <sub>10</sub>	1011 <sub>2</sub>	13 <sub>8</sub>	B <sub>16</sub>



# Sistem Bilangan Desimal

- Bilangan desimal adalah sistem bilangan yang berbasis 10.
- Hal ini berarti bilangan – bilangan pada sistem ini terdiri dari 0 sampai dengan 9.

contoh :

- $126_{10}$  (umumnya hanya ditulis 126)
- $11_{10}$  (umumnya hanya ditulis 11)



# Sistem Bilangan Biner

- Bilangan dalam bentuk biner adalah bilangan berbasis 2.
- Ini menyatakan bahwa bilangan yang terdapat dalam sistem ini hanya 0 dan 1.

contoh

- $1111110_2$
- $1011_2$

Nilai dari 2 pangkat n	128	64	32	16	8	4	2	1
Posisi bit ke-n	7	6	5	4	3	2	1	0



# Sistem Bilangan Oktal

- Bilangan dalam bentuk oktal adalah sistem bilangan yang berbasis **8**.
- Hal ini berarti bilangan–bilangan yang diperbolehkan hanya berkisar antara **0 – 7**.

contoh :

- $176_8$
- $13_8$



# Sistem Bilangan Heksadesimal

- Bilangan dalam sistem heksadesimal adalah sistem bilangan berbasis **16**.
- Sistem ini hanya memperbolehkan penggunaan bilangan dalam skala **0 – 9**, dan menggunakan **huruf A – F, atau a – f** karena perbedaan kapital huruf tidak memiliki efek apapun.

contoh :

- $7E_{16}$
- $B_{16}$



# Desimal ke Biner / Biner ke Desimal

- Untuk mengubah angka desimal menjadi angka biner digunakan metode pembagian dengan angka 2 sambil memperhatikan sisanya.
- Ambil hasil bagi dari proses pembagian sebelumnya, dan bagi kembali bilangan tersebut dengan angka 2.
- Ulangi langkah langkah tersebut hingga hasil bagi akhir bernilai 0 atau 1.
- Kemudian susun nilai–nilai sisa dimulai dari nilai sisa terakhir sehingga diperoleh bentuk biner dari angka bilangan tersebut



# Contoh : Desimal ke Biner

- 126 (10) = .... (2)

	<i>Hasil Bagi</i>	<i>Nilai Sisa</i>	
• 126 / 2 =	63	0 (LSB)	↑ penulisan hasil
• 63 / 2 =	31	1	
• 31 / 2 =	15	1	
• 15 / 2 =	7	1	
• 7 / 2 =	3	1	
• 3 / 2 =	1	1	
• 1 / 2 =	0	1 (MSB)	

- Hasil : 1 1 1 1 1 1 0 (2)



# Contoh: Biner ke Desimal

- 1 1 1 1 1 1 0 (2) = .... (10)?

1 1 1 1 1 1 0

Posisi bit ke-n 6 5 4 3 2 1 0

1	x	$2^6$	=	64
1	x	$2^5$	=	32
1	x	$2^4$	=	16
1	x	$2^3$	=	8
1	x	$2^2$	=	4
1	x	$2^1$	=	2
0	x	$2^0$	=	0
<b>Jumlah dalam desimal</b>				<b>126</b>



## Desimal ke Oktal/Heksadesimal & Oktal/Heksadesimal ke Desimal

- Konversi bilangan desimal ke bilangan oktal atau bilangan heksadesimal pada dasarnya sama dengan konversi bilangan desimal ke biner.
- Perbedaannya terletak pada bilangan pembagi.
- Jika pada konversi biner pembaginya adalah angka 2, maka pada konversi oktal pembaginya adalah angka 8, sedangkan pada konversi heksadesimal pembaginya adalah 16.



# Contoh: Desimal ke Oktal/Heksa

- $95(10) = \dots (8) ?$

				Hasil Bagi	Sisa Bagi
95	/	8	=	11	7 (LSB)
11	/	8	=	1	3
1	/	8	=	-	1 (MSB)



penulisan hasil

$$95(10) = 137(8)$$

- $31391(10) = \dots (16) ?$



# Biner ke Oktal dan Oktal ke Biner

- Untuk mengubah bilangan biner ke oktal, gunakan cara Representasi singkat (Shorthand Representation) yaitu kita pilah bilangan tersebut menjadi 3 bit bilangan biner dari kanan ke kiri.
- Tabel berikut ini menunjukkan representasi bilangan biner terhadap bilangan oktal :

<i>Digit Oktal</i>	<i>Representasi Biner</i>
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

contoh :

$$1111110_2 = ?_8$$

**001**

**111**

**110**

**1**

**7**

**6**



# Biner ke Heksadesimal dan Heksadesimal ke Biner

- Pengubahan bilangan Biner ke Heksadesimal dilakukan dengan pengelompokan setiap empat bit Biner dimulai dari bit paling kanan.
- Kemudian konversikan setiap kelompok menjadi satu digit Heksadesimal.

Tabel berikut menunjukkan representasi bilangan Biner terhadap digit Heksadesimal:

<i>Digit Heksadesimal</i>	<i>Representasi Biner</i>	<i>Digit Heksadesimal</i>	<i>Representasi Biner</i>
0	0000	A	1010
1	0001	B	1011
2	0010	C	1100
3	0011	D	1101
4	0100	E	1110
5	0101	F	1111
6	0110		
7	0111		
8	1000		
9	1001		

contoh :

$$11111102 = ?_{16}$$

**0111**

**7**

**1110**

**E**



**ANY  
QUESTIONS?**



**Sesi Berakhir**  
**TERIMA KASIH**