

# APLIKASI DESKTOP

## Generasi Bahasa Pemograman

- Name | Chalifa Chazar
- Email | [chalifa.chazar@gmail.com](mailto:chalifa.chazar@gmail.com)
- Website | <http://script.id>

# GENERASI BAHASA PEMOGRAMAN

- Generasi 1 (Machine Language)
- Generasi 2 (Assembly Language)
- Generasi 3 (Procedural Programming)
- Generasi 4 (Fourth-Generation Language – 4GL)
- Generasi 5 (Programming Language Based Object Oriented and Web Development)

## GENERASI 1 (MACHINE LANGUAGE)

- Memberikan perintah kepada komputer dengan menggunakan Bahasa Biner yaitu "0" dan "1"
- Sudah jarang di gunakan sekarang untuk membuat suatu perangkat lunak (digunakan untuk kepentingan pendidikan)
- Panjang dan sulit dimengerti
- Tergantung pada rangkaian mesin
- Tetapi prosesnya cepat
- Contoh: **01100101100110**

## GENERASI 2 (ASSEMBLY LANGUAGE)

- Pemrograman dengan bahasa pemrograman assembly (rakitan) merupakan generasi ke dua setelah generasi bahasa mesin
- Bahasa pemrograman assembly memiliki keyword yang lebih kompleks dari pada bahasa mesin yang hanya berisi karakter "0" dan "1" sehingga relatif lebih mudah dimengerti oleh programmer
- Assembler adalah nama bahasa pemrograman ( nama produk ) pada generasi ini

## GENERASI 2 (ASSEMBLY LANGUAGE)

- Kode-kode pada bahasa pemrograman assembly cenderung berupa singkatan yang dikenal dengan sebutan kode mnemonic seperti MOV, SUB, CMP, JMP, JGE, JL, LOOP, dan lain sebagainya
- Kode- kode tersebut hampir mirip dengan kode Terminal (pada sistem operasi Linux) atau kode Command Prompt (pada sistem operasi Windows), misalnya "mkdir" (make directory, untuk membuat folder baru)
- Cenderung lebih mudah diingat di bandingkan dengan bahasa biner

# CONTOH PEMROGRAMAN BAHASA ASSEMBLER

TASM	COMMENTS
MOV AX,1234	Place 1234 into AX
MOV AX,L1234	Contents of add. 1234 to AX
MOV AX,CS:1234	Move From Offset of CS
MOV Byte ptr ...	Move Byte string
Move Word ptr ...	Move word string
ret	Near return
ret	Far return

\***Turbo Assembler (TASM)** is a computer **assembler** (software for program development)

## GENERASI 3 (PROCEDURAL PROGRAMMING)

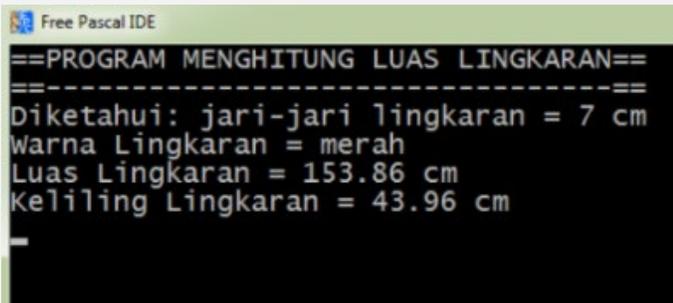
- Generasi ini mulai muncul bahasa pemrograman prosedural yang mengadopsi bahasa yang mirip dengan bahasa manusia namun masih mencampurkan unsur simbol/karakter khusus seperti {, }, ?, <<, >>, &&, ||, ; dan lain sebagainya
- Pada generasi ini juga mulai di kembangkan otomatisasi pengkodean agar programmer fokus pada fungsi utama program yang di kembangkan, unsur simbol/karakter khusus pun mulai di kurangi bahkan pada beberapa bahasa pemrograman telah di hilangkan
- Dikenal juga sebagai Bahasa pemograman tingkat tinggi (high level programming language)

## GENERASI 3 (PROCEDURAL PROGRAMMING)

- Bahasa pemrogramannya menyerupai bahasa manusia
- Mudah dipahami
- Membutuhkan **compiler** atau **interpreter**
- Contoh :
  - Pascal
  - Bahasa C
  - C++
  - Visual Basic (non . NET)

## CONTOH PEMROGRAMAN PASCAL

Hasilnya:



```
Free Pascal IDE
==PROGRAM MENGHITUNG LUAS LINGKARAN==
=====
Diketahui: jari-jari lingkaran = 7 cm
Warna Lingkaran = merah
Luas Lingkaran = 153.86 cm
Keliling Lingkaran = 43.96 cm
-
```

```
1 program hitung_lingkaran;
2
3 uses
4   crt;
5
6 type
7   warna=(merah, kuning, biru);
8
9 const
10  PI=3.14;
11
12 var
13   jari2:integer=7;
14   warna_lingkaran:warna=merah;
15
16 function luas_lingkaran:real;
17 begin
18   luas_lingkaran:=PI*jari2*jari2;
19 end;
20
21 procedure kel_lingkaran(jari2:integer);
22 begin
23   write('Keliling Lingkaran = ');
24   writeln(PI*(jari2+jari2):4:2, ' cm');
25 end;
26
27 begin
28   clrscr;
29   writeln('==PROGRAM MENGHITUNG LUAS LINGKARAN==');
30   writeln('=====');
31   writeln('Diketahui: jari-jari lingkaran = ',jari2,' cm');
32   writeln('Warna Lingkaran = ',warna_lingkaran);
33   writeln('Luas Lingkaran = ',luas_lingkaran:4:2, ' cm');
34   kel_lingkaran(jari2);
35   readln;
36 end.
```

# CONTOH PEMROGRAMAN BAHASA C

```
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int i;
6      for (i = 1; i <= 10; i++) {
7          printf("%i + %i = %i \n",i,i,i+i);
8      }
9      return 0;
10 }
```

Hasil kode program:

```
1 + 1 = 2
2 + 2 = 4
3 + 3 = 6
4 + 4 = 8
5 + 5 = 10
6 + 6 = 12
7 + 7 = 14
8 + 8 = 16
9 + 9 = 18
10 + 10 = 20
```

# PROSES PEMBUATAN PROGRAM

- Proses pembuatan program yaitu kita menulis kode sumber pada teks editor misalnya Notepad kemudian mengubahnya menjadi bahasa mesin yang bisa dieksekusi oleh CPU
- Proses pengubahan kode sumber (source code) menjadi bahasa mesin (machine language) ini terdiri dari dua macam yaitu:
  - Kompilasi (Compiler)
  - Interpretasi (Interpreter)

# KOMPILASI (COMPILER)

- Dalam proses kompilasi semua kode sumber dibaca terlebih dahulu dan jika tidak ada kesalahan dalam menulis program maka akan dibentuk kode mesinnya sehingga program bisa dijalankan
- Program yang melakukan tugas ini disebut Compiler
- Program hasil kompilasi akan berbentuk executable
- Program bisa langsung dijalankan tanpa harus memiliki Compiler di komputer yang menjalankan program tersebut
- Bahasa yang menggunakan teknik kompilasi misalnya bahasa C, C++, Pascal, Assembly dan masih banyak lagi

# INTERPRETASI (INTERPRETER)

- Bahasa yang menggunakan teknik interpretasi akan membaca kode sumber perbaris dan dieksekusi perbaris
- Jika ditemukan kesalahan dalam penulisan program maka di baris kesalahan itulah program akan dihentikan
- Program yang melakukan tugas ini disebut Interpreter
- Pada teknik interpretasi tidak akan dihasilkan program standalone, artinya untuk menjalankan program kita harus mempunyai kode sumbernya sekaligus interpreter program tersebut
- Bahasa yang menggunakan teknik interpretasi misalnya bahasa Perl, Python, Ruby dan masih banyak lagi

# KOMPILASI SEKALIGUS INTERPRETASI

- Ada juga bahasa pemrograman yang menghasilkan programnya dengan teknik kompilasi sekaligus interpretasi (misalnya bahasa JAVA)
- Dalam pembuatan program java kode sumber diubah menjadi bytecode
- Untuk menjalankan bytecode tersebut kita membutuhkan Java Runtime Environment (JRE) yang bertugas sebagai interpreter sehingga menghasilkan program dari bytecode tersebut.

## GENERASI 4 (FOURTH-GENERATION LANGUAGE – 4GL)

- Generasi ke-4 adalah generasi bahasa query terstruktur (SQL-Structured Query Language)
- SQL sebenarnya merupakan bahasa pemrograman namun penerapannya lebih banyak pada basis data (database) misalnya MySQL, Oracle Database, SQL Server, PostgreSQL, SQLite dan masih banyak lagi
- Pemrograman SQL umumnya di gunakan sebagai bahasa sekunder dengan library built-in atau terpisah
- Query yang di sematkan dalam kode suatu bahasa pemrograman tujuannya adalah untuk mendefinisikan basis data dan memanipulasi basis data (dikenal dengan **DDL** (Data Definition Language) dan **DML** (Data Manipulation Language))

# CONTOH BAHASA SQL

Perhatikan contoh Statement SQL berikut ini!

```
UPDATE data_mahasiswa
SET nilai_geografi = nilai_geografi+10
WHERE NIM = 61101045;
```

## Penjelasan UPDATE Statement

Statement tersebut berisi perintah untuk mengubah nilai record dengan kriteria yaitu sebagai berikut:

1. record tersebut terletak pada tabel data\_mahasiswa;
  2. record tersebut terletak pada field nilai\_geografi;
  3. record tersebut terletak sejajar dengan record yang memuat nilai 61101045 pada field NIM;
- Setelah record tersebut ditemukan, maka tambahkan nilai sebesar +10 pada record tersebut.

## GENERASI 5 (PROGRAMMING LANGUAGE BASED OBJECT ORIENTED AND WEB DEVELOPMENT)

- Generasi bahasa pemrograman yang ke-5 lebih menekankan pada aspek efisiensi dan penggunaan kembali (*re-useable*) modul-modul yang di buat dengan bahasa pemrograman tingkat tinggi tertentu
- Generasi ke-5 ini juga di kenal dengan nama “*intelligent programming*” (pemrograman kecerdasan) yang menekankan aspek otomatisasi dalam setiap prosesnya
- Contoh:
  - PHP
  - Visual Basic .NET (VB .NET)
  - Java
  - ASP

# CONTOH PEMROGRAMAN VB .NET

```
Public Class Form1
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        MsgBox("Hello world", MsgBoxStyle.Information, "Hello world!") ' Show a message that says
        "Hello world!".
    End Sub
End Class
```

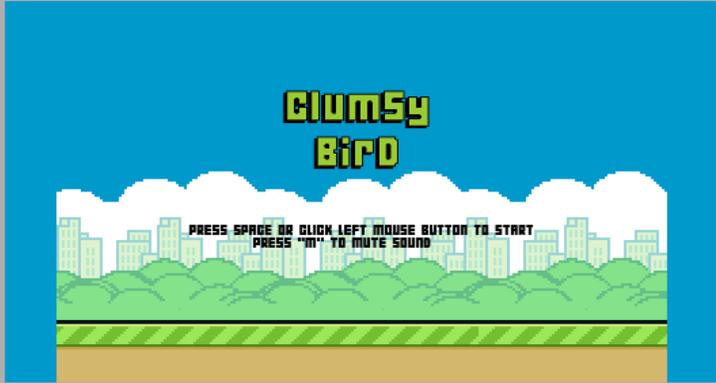
# CONTOH PENGGUNAAN BAHASA PHP UNTUK MEMBANGUN WEBSITE

</EDU> [Module](#) [Upload Assesment](#) [Research](#) [About](#)

## SCRIPT.ID

Wadah untuk berbagi ilmu, khususnya bidang Teknologi Informasi (TI). Semoga bermanfaat dan berguna dalam mendukung kemajuan pendidikan bangsa Indonesia.  
Your time is limited, so don't waste it living someone else's life - Steve Jobs -

Wow.. Hari ini tidak ada tugas..



**Upload Terbaru**

	Pertemuan 3 Old Meets New: Distributed Computing Big Data (KRY)	October 28th, 2020
	Pertemuan 3 Bahasa mesin (machine language) Aplikasi Desktop (MP)	October 28th, 2020
	Pertemuan 2 Examining Big Data Types Big Data (KRY)	October 20th, 2020
	Pertemuan 2 Software, aplikasi dan program Aplikasi Desktop (MP)	October 20th, 2020
	Pertemuan 1	

[Back to Top](#)

©Support by SCRIPT.ID (V.2) [LOGIN](#) [f](#) [t](#) [i](#)

Last update : October 2020 | [chalifa.chazar@gmail.com](mailto:chalifa.chazar@gmail.com)

## KESIMPULAN

- Meskipun setiap bahasa pemrograman dibuat untuk membuat program namun setiap bahasa dibuat dengan tujuan dan fungsi yang berbeda-beda
- Misalnya untuk membuat driver hardware kita tidak bisa menggunakan bahasa Visual Basic. Untuk membuat program berbasis sistem seperti driver kita bisa gunakan bahasa C atau Assembly
- Contohnya sistem operasi linux yang open source. Jika anda melihat kode sumbernya anda akan menemukan bahwa linux dibuat menggunakan bahasa C
- Sedangkan untuk pemrograman desktop kita bisa menggunakan Visual Basic. Bahasa tersebut dirancang oleh Microsoft untuk pemrograman desktop dengan tampilan GUI yang memukau

[chalifa.chazar@gmail.com](mailto:chalifa.chazar@gmail.com)  
| <http://script.id> |

</THANKS>