

# **PERTEMUAN 14**

## **IMPLEMENTASI ALGORITMA KE DALAM BAHASA PEMROGRAMAN BERORIENTASI OBJEK**

# POKOK BAHASAN

Bahasa Pemrograman Yang Mendukung OOP

# BAHASA PEMROGRAMAN BERORIENTASI OBJEK

Bahasa pemrograman yang mendukung OOP antara lain:

1. SmallTalk
2. C++
3. Java, dll

# SMAL TALK

- Smalltalk adalah bahasa pemrograman berorientasi objek yang pertama kali populer.
- Smalltalk dikembangkan di Xerox PARC-Amerika Serikat.
- Ide dasar Smalltalk meliputi:
  1. Semua adalah objek
  2. Objek bisa saling berkomunikasi melalui messages atau berita
  3. Semua tersedia untuk dimodifikasi

# BAHASA C++

- **C++** dibuat pada tahun 1980an oleh Bell Labs sebagai pengembangan dari Bahasa Pemrograman C (yang dibuat oleh Brian W. Kernighan dan Denies M. Ricthie pada tahun 1972). Salah satu perbedaan yang paling mendasar dengan bahasa C adalah C++ mendukung konsep pemrograman berorientasi objek (Object Oriented Language).
- Tahun 1980 seorang ahli yang bernama **Bjarne Stroustrup** mengembangkan beberapa hal dari bahasa C yang dinamakan “C with Classes” yang berganti nama pada tahun 1983 menjadi C++.
- C++ merupakan bahasa pemrograman yang *case sensitive*.

# JAVA

- Java adalah bahasa pemrograman berorientasi objek yang dikembangkan oleh Sun Microsystems sejak tahun 1991.
- Bahasa Pemrograman Java pertama lahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh Joe Palrang.
- Java adalah bahasa pemrograman yang mirip dengan C/C++
- Keunggulannya adalah Java dapat berjalan di banyak platform perangkat keras (Multy-Platform).

# PHP

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama Form Interpreted (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilis kode sumber ini menjadi sumber terbuka, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan. Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: Hypertext Preprocessor.

Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai pada awal abad ke-21. Versi ini banyak dipakai disebabkan kemampuannya untuk membangun aplikasi web kompleks tetapi tetap memiliki kecepatan dan stabilitas yang tinggi. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar. Versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi objek. Versi terbaru dan stabil dari bahasa pemrograman PHP saat ini adalah versi 8.0.

# PYTHON

Python dibuat dan dikembangkan oleh Guido Van Rossum, yaitu seorang programmer yang berasal dari Belanda. Pembuatannya berlangsung di kota Amsterdam, Belanda pada tahun 1990. Pada tahun 1995 Python dikembangkan lagi agar lebih kompatibel oleh Guido Van Rossum. Selanjutnya pada awal tahun 2000, terdapat pembaharuan versi Python hingga mencapai Versi 3 sampai saat ini. Python mendukung beberapa paradigma pemrograman, termasuk pemrograman terstruktur (terutama, prosedural), berorientasi objek, dan fungsional.