

PEMROGRAMAN BERORIENTASI OBJEK DAN PENGENALAN BASIS DATA

KELAS : X RPL

GURU PENGAMPU : Hj. IKA ROKHASARI LESTARI, S.Pd, S.Kom, Gr.

PERTEMUAN KE – 1 :

A. PENGERTIAN PEMROGRAMAN BERORIENTASI OBJEK (OOP)

Pemrograman berorientasi objek (Inggris: *object-oriented programming* disingkat OOP) merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk *field* atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai *method*. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

Pemrograman berorientasi objek (*Object Oriented Programming* atau disingkat OOP) adalah paradigma pemrograman yang berorientasikan kepada objek yang merupakan suatu metode dalam pembuatan program, dengan tujuan untuk menyelesaikan kompleksnya berbagai masalah program yang terus meningkat. Objek adalah *entitas* yang memiliki atribut, karakter (*bahaviour*) dan kadang kala disertai kondisi (*state*) (Douglas, 1992).

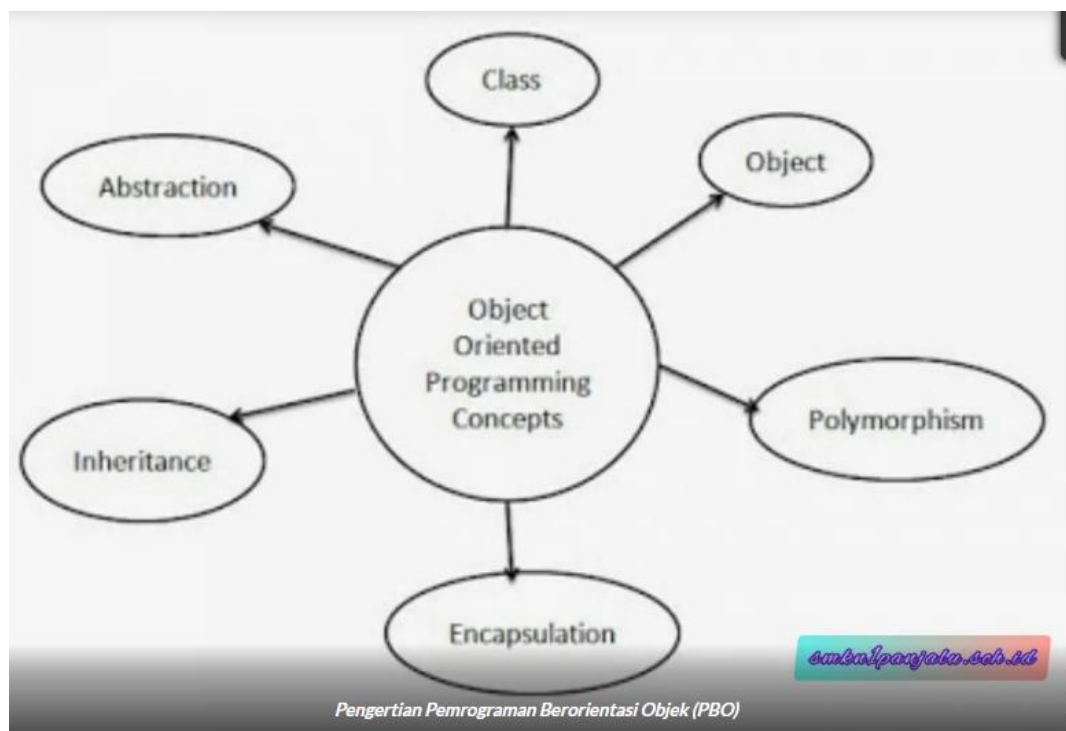
Pemrograman berorientasi objek ditemukan pada Tahun 1960, dimana berawal dari suatu pembuatan program yang terstruktur (*structured programming*). Metode ini dikembangkan dari bahasa C dan Pascal. Dengan program yang terstruktur inilah untuk pertama kalinya kita mampu menulis program yang begitu sulit dengan lebih mudah.

Ide dasar pada OOP adalah mengkombinasikan data dan fungsi untuk mengakses data menjadi sebuah kesatuan unit yang dikenal dengan nama objek. Objek adalah struktur data yang terdiri dari bidang data dan metode bersama dengan interaksi mereka untuk merancang aplikasi dan program komputer. Semua data dan fungsi di dalam paradigma ini

dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik peranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

B. KONSEP DASAR OOP:



Keterangan:

- **Kelas** — kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. *Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan*

domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

- **Objek** - membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
- **Abstraksi** - Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.
- **Enkapsulasi** - Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
- **Polimorfisme** melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode

yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

- **Inheritance.** Konsep inheritas mempunyai fungsi mengatur polimorfisme dan enkapsulasi dengan mengizinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Objek-objek ini dapat membagi dan memperluas perilaku mereka tanpa mengimplementasikan perilaku tersebut.
- Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Sebagai contoh anggap kita memiliki sebuah departemen yang memiliki manager, sekretaris, petugas administrasi data dan lainnya. Misal manager tersebut ingin memperoleh data dari bag administrasi maka manager tersebut tidak harus mengambilnya langsung tetapi dapat menyuruh petugas bag administrasi untuk mengambilnya. Pada kasus tersebut seorang manager tidak harus mengetahui bagaimana cara mengambil data tersebut tetapi manager bisa mendapatkan data tersebut melalui objek petugas administrasi. Jadi untuk menyelesaikan suatu masalah dengan kolaborasi antar objek-objek yang ada karena setiap objek memiliki deskripsi tugasnya sendiri.

C. KELEBIHAN DARI METODE PEMROGRAMAN BERORIENTASI OBJEK SENDIRI ADALAH SEBAGAI BERIKUT:

1. Maintenance program lebih mudah. Program yang dibuat dengan metode pemrograman berorientasi objek lebih mudah dibaca dan dipahami. Selain itu, pemrograman berorientasi objek dapat mengontrol kerumitan program hanya dengan cara mengizinkan rincian-rincian yang dibutuhkan oleh programmer.
2. Mudah dalam pengubahan program, bisa berupa penambahan atau penghapusan fitur atau objek tertentu. Contoh perubahan yang bisa dilakukan antara lain penambahan dan penghapusan data di dalam suatu database.
3. Objek-objek di dalam program dapat digunakan sesering mungkin oleh programmer. Programmer dapat menyimpan objek-objek yang dirancang ke dalam sebuah

module, yang dapat disisipkan ke dalam baris kode baru. Penambahan bisa dilakukan dengan sedikit perubahan atau tanpa perubahan pada kode program utama.

D. BAHASA PEMROGRAMAN YANG MENDUKUNG OOP

Berikut ini adalah Bahasa pemrograman yang mendukung OOP antara lain seperti:

1. Visual Foxpro
2. Java
3. C++
4. Pascal (bahasa pemrograman)
5. SIMULA
6. Smalltalk
7. Ruby
8. Python
9. PHP
10. C#
11. Delphi
12. Eiffel
13. Perl
14. Adobe Flash AS 3.0

DAFTAR REFERENSI:

<https://dosenit.com/kuliah-it/pemrograman/contoh-pemrograman-berorientasi-objek>

https://id.wikipedia.org/wiki/Pemrograman_berorientasi_objek

<https://smkn1panjalu.sch.id/pengertian-pemrograman-berorientasi-objek-pbo/>

Tugas Individu:

1. Baca dan pahami materi tentang OOP di atas
2. Buatlah kesimpulan tentang OOP sesuai dengan pemahaman Anda
3. Carilah informasi sebanyak banyaknya tentang:
 - a. Sejarah OOP
 - b. Pengertian OOP minimal 5 pengertian dari sumber yang berbeda
 - c. Manfaat OOP
 - d. Contoh bahasa pemrograman yang mendukung OOP selain yang disebutkan di atas.
 - e. Contoh program OOP sebanyak 5 program
4. Ketik kesimpulan dan masing masing informasi yang Anda dapat sesuai dengan tugas yang diberikan di atas menggunakan Microsoft Word dengan ketentuan:
 - a. Font: Arial Narrow
 - b. Font size: 12
 - c. Page layout: Paper A4, Margin: Top 3cm, Bottom 3cm, Left 4cm, Right 3cm (ketentuan ini seperti yang digunakan di materi ini)
 - d. Line spacing: 1,5 lines
5. Simpan dokumen Anda dengan nama file: tugas1_PBO_kelas_Namasiswa
6. Kirim hasil tugas di link <https://forms.gle/QR2bHrWUNkGr1dDo7> ini
7. Selamat belajar