**Tugas Kedua**

*Java Vs C++*

****

Disusun Oleh:

Phoebe Ivana 2206820320

Darren Nathanael Boentara 2206059490

Darmawan Hanif 2206829175

Edgrant Henderson S. 2206025016

PROGRAM STUDI TEKNIK KOMPUTER

FAKULTAS TEKNIK

UNIVERSITAS INDONESIA

2023/2024

***Pembahasan mengenai Perbedaan penerapan OOP pada Bahasa Pemrograman Java dan C++***

Terdapat beberapa perbedaan dari penerapan OOP atau *Object Oriented Programing* dengan bahasa pemrograman *java* dan C++. Yang pertama adalah perbedaan terkait *inheritance.* Java mendukung hanya satu kali pewarisan langsung. Yang dimana, jika dilakukan *multiple inheritance*, akan mengakibatkan sulitnya pembuatan aplikasi dan compiler. Sehingga *Java* menggunakan *interface*. Sedangkan dalam C++, didukung untuk menggunakan pewarisan ganda (*multiple inheritance*). Yang dimana sebuah kelas dapat mewarisi lebih dari satu kelas dasar.

Perbedaan kedua dapat dilihat dari kesalahan akses memori. Yang dimana *java* memiliki mekanisme dalam mencegah kesalahan akses memori yaitu menggunakan *exceptions*. Sedangkan, C++ memberikan kebebasan kepada *user*. Hal ini membuat kesalahan dalam mengakses memori lebih banyak. Walaupun, dengan C++ dapat menghasilkan program lebih cepat, tetapi dapat lebih rentan dalam kesalahan.

Perbedaan ketiga dapat dilihat dari *garbage collection.* Yang dimana *java* memiliki *Garbage Collection* yang mengelola memori dengan menghapus objek yang tidak lagi diakses. Yang dimana, hal ini membantu untuk menjamin ketersediaan memori dengan membebaskan memori yang tidak terpakai. Sedangkan, C++ tidak memiliki *Garbage Collection* secara otomatis. Hal ini dapat memungkinkan fleksibilitas, tetapi memerlukan lebih banyak perhatian dalam manajemen memori.

Dengan demikian, perbedaan antara penerapan OOP pada Bahasa pemrograman Java dengan C++ dapat disimpulkan dalam bentuk table berikut.

|  |  |  |
| --- | --- | --- |
| Perbedaan | Java | C++ |
| *Inheritance* | Pewarisan tunggal didukung oleh Java dengan menggunakan kata kunci “*extends*”; kurang mendukung terjadinya *multiple inheritance*. | Mendukung pewarisan tunggal maupun ganda melalui kata kunci “class” dan “struct”. |
| *Garbage Collection* | Dimiliki secara otomatis oleh Java sedemikian rupa sehingga dapat mencegah terjadinya kebocoran pada memori, serta proses alokasi memori dapat dihandle oleh Java secara langsung. | Pengaturan manajemen memori harus dilakukan secara manual sedemikian rupa sehingga dibutuhkan kemahiran pemrogram dalam mengalokasikan dan mengelola memorinya sendiri. |
| Pointer | Umumnya, penggunaan pointer pada Java dilakukan secara implisit. | Penggunaan pointer pada C++ dilakukan secara eksplisit, sedemikian rupa sehingga proses manipulasi data pada memori dapat dilakukan secara mendetail. |
| Abstraksi | Mendukung; menggunakan kata kunci “*abstract*” dan “*interface*”. | Mendukung; menggunakan kata kunci “*abstract*” dan “*virtual*”. |
| *Polymorphism* | Mendukung; dilakukan melalui konsep *overriding* dan *interface*. | Mendukung; dilakukan melalui konsep *multiple dispatch*, *overriding*, dan lain sebagainya. |
| Keamanan | Cukup baik dikarenakan konsep enkapsulasi dapat diterapkan pada Java; terdapat metode keamanan seperti *sandboxing*. | Kurang baik dikarenakan rentan terhadap bug; umumnya berdampak pada kebocoran memori. |

***Laporan pembuatan aplikasi Java dengan menggunakan BlueJ***

Aplikasi kalkulator adalah program sederhana yang dapat melakukan operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian. Dalam tugas ini, kami telah membuat aplikasi kalkulator sederhana menggunakan bahasa pemrograman Java dan menggunakan BlueJ sebagai *IDE*. Berikut langkah-langkah yang kami lakukan untuk membuat aplikasi kalkulator.

1. Membuat Proyek BlueJ

Pertama-tama, kami membuat proyek baru di BlueJ dengan nama "Tugas2".

1. Membuat Kelas Kalkulator

Selanjutnya, kami membuat kelas baru dengan nama "Calc" di dalam proyek BlueJ.

1. Menuliskan Kode untuk Input dan Output Operasi Matematika

Selanjutnya, kami menuliskan kode-kode untuk operasi matematika dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian di dalam kelas "Calc". Berikut adalah kodenya.

A screenshot of a computer

Description automatically generated

1. Menguji Aplikasi Kalkulator

Langkah terakhir, kami meng*compile* kelas “Calc” tersebut, lalu melakukan pengujian hasil dengan input tertentu. Berikut salah satu hasil pengujian aplikasi.

A screenshot of a computer

Description automatically generated