

PEMROGRAMAN BERORIENTASI OBJECT

NAMA : ILYAS RUKYAT

NIM : 201856006

SOAL

- 1. Buatlah resume tentang sejarah singkat lahirnya pemrograman JAVA**
- 2. Apa yang dimaksud dengan dengan compiler dan interpreter jelaskan konsep kerjanya serta alur proses kerjanya**
- 3. Jelaskan apa yang dimaksud dengan pemrograman berorientasi object serta konsep kerjanya (* penjelesan gunakan bahasa sendiri apabila mengutip referensi cantumkan kutipanya)**
- 4. Sebutkan dan Jelaskan apa saja yang diperlukan dalam proses pembuatan pemrograman**

JAVA

JAWABAN

1. Java secara resmi diperkenalkan oleh SUN pada dunia pada tanggal 23 Mei 1995. Sedangkan sejarah pembangunan java sendiri sudah dimulai sejak tahun 1991. Saat itu tim “ Stealth Project “ mengadakan pertemuan (brainstorming) untuk menciptakan suatu system software yang mampu berjalan pada alat-alat elektronik (small devices).

James Gosling berkonsultasi pada ide pembuatan bahasa pemrograman. Pada Juni 1991 , muncullah bahasa interpreter “Objek” yang menjadi cikal bakal dari Java. Kemudian secara resmi pada tahun 1995 Java diperkenalkan bersama browser HotJava, Java pun merambah k dunia web.

Kenyataan ini mungkin agak sedikit berbeda dengan ide pembuatan java pada awalnya. Internet ternyata membantu membuat Java terkenal seperti sekarang ini. Memang harus diakui karena semakin berkembangnya Internet, maka focus pemrograman saat ini mengarah ke pemrograman Internet itu sendiri.

Saat ini Java dibagi menjadi 3 framework atau teknologi yaitu J2SE untuk pemrograman aplikasi berbasis console dan desktop, kemudian J2EE berskala enterprise seperti aplikasi web-base (JSP dan Servlet), komponen (EJB), web service dan lain-lain. Kemudian framework yang terakhir adalah J2ME untuk pemrograman small device seperti handphone dan PDA.

2. Compiler Adalah suatu program yang menterjemahkan bahasa program (source code) ke dalam bahasa objek (object code). Compiler menggabungkan keseluruhan bahasa program dikumpulkan kemudian disusun kembali.

konsep kerja compiler serta alur proses kerjanya :

Compiler membaca sebuah source code dalam bentuk teks, menyatukan karakter-karakter yang berhubungan menjadi token,lalu memeriksa apakah token-token tersebut memenuhi grammar,setelah itu compiler akan memeriksa semantik input,dan membuat output dalam sebuah bahasa(yang umumnya adalah assembly maka proses berikutnya adalah assembling yang dilakukan dengan assembler untuk menghasilkan bahasa mesin,proses terakhir untuk membuat executable file dilakukan oleh linker.

Interpreter adalah dalam eksekusi yang bisa dilakukan dengan segera tanpa melalui tahap kompilasi, untuk alasan ini interpreter digunakan pada saat pembuatan program berskala besar.

konsep kerja interpreter serta alur proses kerjanya :

- » Kode program yang dibuat oleh programmer ke dalam bahasa mesin.
- » Mengeksekusi perintah baris demi baris dengan mengikuti logika yang ada.
- » Secara garis besar, Interpreter menterjemahkan baris per baris.

» Penyusunan program relatif lebih cepat dan bisa langsung diuji sekalipun masih ada beberapa kesalahan secara kaidah dalam program.

» Kode program tidak dapat dirahasiakan.

3. OOP (Object Oriented Programming) adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, dan objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil lagi. Saya ambil contoh Pesawat, Pesawat adalah sebuah objek. Pesawat itu sendiri terbentuk dari beberapa objek yang lebih kecil lagi seperti mesin, roda, baling-baling, kursi, dll. Pesawat sebagai objek yang terbentuk dari objek-objek yang lebih kecil saling berhubungan, berinteraksi, berkomunikasi dan saling mengirim pesan kepada objek-objek yang lainnya. Begitu juga dengan program, sebuah objek yang besar dibentuk dari beberapa objek yang lebih kecil, objek-objek itu saling berkomunikasi, dan saling berkirim pesan kepada objek yang lain.

konsep OOP

Abstrak Class

1. Kelas merupakan deskripsi abstrak informasi dan tingkah laku dari sekumpulan data.
2. Kelas dapat diilustrasikan sebagai suatu cetak biru(blueprint) atau prototipe yang digunakan untuk menciptakan objek.
3. Kelas merupakan tipe data bagi objek yang mengenkapsulasi data dan operasi pada data dalam suatu unit tunggal.
4. Kelas mendefinisikan suatu struktur yang terdiri atas data kelas (data field), prosedur atau fungsi (method), dan sifat kelas (property).

Encapsulation

1. Istilah enkapsulasi sebenarnya adalah kombinasi data dan fungsionalitas dalam sebuah unit tunggal sebagai bentuk untuk menyembunyikan detail informasi.
2. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.
3. Enkapsulasi menekankan pada antarmuka suatu kelas, atau dengan kata lain bagaimana menggunakan objek kelas tertentu.
4. Contoh: kelas mobil menyediakan antarmuka fungsi untuk menjalankan mobil tersebut, tanpa kita perlu tahu komposisi bahan bakar, udara dan kalor yang diperlukan untuk proses tersebut.

Inheritance

- 1.Kita dapat mendefinisikan suatu kelas baru dengan mewarisi sifat dari kelas lain yang sudah ada.**
- 2.Penurunan sifat ini bisa dilakukan secara bertingkat-tingkat, sehingga semakin ke bawah kelas tersebut menjadi semakin spesifik.**
- 3.Sub kelas memungkinkan kita untuk melakukan spesifikasi detail dan perilaku khusus dari kelas supernya.**
- 4.Dengan konsep pewarisan, seorang programmer dapat menggunakan kode yang telah ditulisnya pada kelas super berulang kali pada kelas-kelas turunannya tanpa harus menulis ulang semua kodekode itu.**

Polymorphism

- 1.Polimorfisme merupakan kemampuan objekobjek yang berbeda kelas namun terkait dalam pewarisan untuk merespon secara berbeda terhadap suatu pesan yang sama.**
- 2.Polimorfisme juga dapat dikatakan kemampuan sebuah objek untuk memutuskan method mana yang akan diterapkan padanya, tergantung letak objek tersebut pada jenjang pewarisan.**
- 3.Method overriding.**
- 4.Method name overloading.**

<https://idcloudhost.com> › panduan › mengenal-pengertian-dan-konsep-oop

- 4. 1.Tulis kode: Menulis kode dengan Netbeans maupun teks editor yang lain.**
- 2.Kompilasi: Mengubah kode menjadi bytecode.**
- 3. Eksekusi: Menjalankan program.**