**Nama : Zidan Mulyana**

**Npm : 0620101019**

**UTS PBO**

Bagian I Teori (75)

1. Jelaskan apa yang dimaksud dengan pemrograman berorientasi object ?

2. Jelaskan life cycle dari bahasa pemrograman java ?

3. Apa perbedaan antara object dan class ?

4. Apa kelebihan java dibandingkan pemrograman lainnya ?

5. Sebutkan beberapa konsep pemorgraman yang menjadi ciri khas OOP?

Jawab

1. Pemrograman berpandangan objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Seluruh 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,
2. Lifecycle adalah class yang menyimpan informasi tentang status siklus proses suatu komponen (seperti aktivitas atau fragmen) dan memungkinkan objek lain untuk mengamati status ini.

Lifecycle menggunakan dua enumerasi utama untuk melacak status siklus proses komponen teratribusinya

Kasus penggunaan yang umum adalah menghindari callback tertentu jika Lifecycle tidak sedang berstatus baik. Misalnya, jika callback menjalankan transaksi fragmen setelah status aktivitas disimpan, maka callback ini akan memicu error, dan kita tidak ingin memicu callback itu lagi.

Untuk mempermudah kasus penggunaan ini, class Lifecycle mengizinkan objek lain mengkueri status saat ini.

**Class**

Class merupakan konsep pokok bahasa pemograman berorientasi object, class merupakan tipe data abstract yang mendeskripsikan kelompok object / properti / atribut. Di dalam class terdiri dari :

* Variabel, variabel di dalam class lebih dikenal sebagai attribute atau properti atau object.
* Fungsi, dikenal dengan istilah method.

Di dalam java, untuk membuat class, berdasarkan standar konfrensi penamaan class, maka harus diawali dengan huruf besar contoh : Mahasiswa (ini sebagai class) mahasiswa (ini sebagai variabel atau objek)

**Class** merupakan konsep pokok di bahasa pemograman berorientasi object, demikian juga di Java. Kelas merupakan bentuk logis yang menjadi landasan bangun seluruh bahasa pemograman berorientasi object. Kelas mendefinisikan bentuk dan perilaku object. Kelas sungguh sangat penting di Java. Sembarang konsep/abstraksi yang diimplementasikan di Java harus di kapsulkan di kelas. Kelas adalah tipe data abstrak yang dilengkapi implementasi parsial atau total. Kelas mendeskripsikan kelompok object dengan properti-properti (atribut-atribut) serupa, perilaku (operasi-operasi) yang common, keterhubungan terhadap himpunan object-object lain dan semantik yang common. Kelas (class) merupakan contoh abstrak dari sebuah object yang telah terbentuk dari proses penyederhanaan. Dengan

**Objek**

**objek** adalah entitas dasar saat runtime. pada saat kode program dieksekusi, objek berinteraksi satu sama lain tanpa harus mengetahui detil data atau kodenya. interaksi antar objek ini dilakukan menggunakan suatu message. objek memiliki suatu siklus hidup, yaitu diciptakan, dimanipulasi, dan di hancurkan

Objek :

* Secara sederhana, objek merupakan segala sesuatu yang dapat dibedakan satu sama lainnya. Segala sesuatu yang ada di alam semesta ini adalah objek. Contohnya: manusia, mobil, hewan, tumbuhan, tempat, atau bahkan yang tidak bersifat fisik seperti kejadian atau konsep-konsep. Sehingga bisa disimpulkan bahwa objek tidak harus bersifat fisik, karena jika dikaitkan dengan OOP objek akan menjadi bentuk logis.Suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya.
* Objek dalam konsep OOP masih memiliki keadaan dan sifat seperti halnya objek di dunia nyata, karena pada dasarnya objek dalam OOP merupakan representasi dari dunia nyata. Objek dalam OOP merepresentasikan keadaan melalui variabel, sedangkan sifatnya direpresentasikan menjadi method. Method merupakan suatu fungsi (sub-routine) yang berhubungan dengan objek.
* Dalam konteks OOP, objek adalah instansiasi (yang dibentuk secara seketika) dari kelas pada saat eksekusi (seperti halnya deklerasi variabel pada pemograman prosedural).
* Jadi semua objek adalah instan dari kelas.
* Obyek merupakan hasil instansiasi dari suatu kelas.
* Sehingga kelas digunakan untuk menciptakan banyak objek
* Proses pembentukan obyek dari suatu class disebut dengan instantiation.
* Obyek disebut juga instances.

**Class :**

* class merupakan suatu konsep yang lebih tinggi lagi hierarkinya dari objek yang dihasilkan dari proses generalisasi objek-objek yang memiliki beberapa ciri yang sama. Dalam OOP, kelas merupakan suatu hasil pemodelan fakta-fakta dari suatu objek yang berguna bagi suatu aplikasi yang diprogram. Proses pemodelan fakta-fakta tersebut disebut dengan abstraksi.
* Dalam suatu sistem, sering kali ditemui beberapa obyek sejenis
* Beberapa obyek sejenis dapat di-identifikasi sebagai suatu kelas (class)
* Kelas mendefinisikan bentuk dan perilaku objek
* Kelas adalah  template/ prototype yang mendefinisikan Type objek
* Merupakan sarana pengkapsulan kumpulan data danmetode-metode yang beroperasi pada kumpulan data
* Kelas merupakan “cetakan” (blueprint) dari suatu obyek
* Dari suatu kelas kita dapat membuat obyek-obyek baru yang masing-masing dapat memiliki keadaan yang berbeda-beda.
* Dalam suatu sistem, sering kali ditemui beberapa obyek sejenis
* Beberapa obyek sejenis dapat di-identifikasi sebagai suatu kelas (class)
* Kelas mendefinisikan bentuk dan perilaku objek
* Kelas adalah  template/ prototype yang mendefinisikan Type objek
* Merupakan sarana pengkapsulan kumpulan data danmetode-metode yang beroperasi pada kumpulan data.
* Kelas merupakan “cetakan” (blueprint) dari suatu obyek
* Dari suatu kelas kita dapat membuat obyek-obyek baru yang masing-masing dapat memiliki keadaan yang berbeda-beda

1. Dibandingkan dengan berbagai jenis bahasa pemrograman lainnya memang keunggulan Java menjadi bahan pertimbangan begitu banyak orang yang memilih untuk menggunakannya. Pengaplikasian yang tepat tentu akan mendukung dihasilkannya aplikasi yang dapat digunakan dengan baik sesuai kebutuhan masyarakat dan memudahkan mereka dalam memenuhi kebutuhan penggunaan aplikasi

Berikut beberapa keunggulan Java yang bisa anda pertimbangkan dibandingkan bahasa pemrograman lainnya :

* Bersifat *multiplatform* sehingga memang mudah dan bisa diaplikasikan untuk berbagai jenis platform.
* Lebih mudah untuk dikembangkan karena berbagai aplikasi yang menggunakan Java sebagai basis dalam bahasa pemrogramannya maka akan memiliki potensi besar untuk dikembangkan lebih lanjut.
* Bahasa pemrograman yang digunakan tentunya akan berkaitan erat dengan kemampuan aplikasi tersebut saat digunakan. Maka bagi para programmer yang mementingkan atau berorientasi pada *usability* maka Java akan sangat mendukung hal tersebut secara optimal
* Membuat anda mudah dalam menyusun script yang akan digunakan dalam membuat sebuah program.
* Bersifat dinamis sehingga membuatnya menjadi bahasa pemrograman yang mudah untuk dikembangkan sesuai kebutuhan. Struktur kode yang sudah dibuat bisa dengan mudah dilakukan modifikasi dan dikembangkan sesuai kebutuhan user dan menjadi fasilitas yang sangat menguntungkan bagi para developer aplikasi
* Bahasa pemrograman ini lebih bersifat teknis sehingga berorientasi pada objek yang artinya aplikasi secara langsung dibangun menggunakan maka akan disesuaikan dengan objek atau tampilan dan interface dari aplikasi tersebut saat digunakan
* Akarnya aplikasi berbasis Android. Jika Anda menggunakan ponsel pintar berbasis Android sekarang maka Anda menggunakan aplikasi-aplikasi yang ditulis dengan menggunakan framework dari Java. Sedangkan pasar tablet dan ponsel pintar saat ini masih dikuasai oleh OS Android dibandingkan dengan iOS milik Apple.
* Java terus menerus dikembangkan. Java memang sudah berusia lebih dari 20 tahun, namun bukan berarti bahasa pemrograman ini sudah uzur dan kehilangan keunggulannya dan relevansinya untuk membuat aplikasi yang modern untuk di era maju sekarang ini.

1. Object Oriented Programming (OOP) adalah suatu pemprograman yang berorientasi objek. Ciri-ciri atau karakteristik pemrograman berorientasi objek, antara lain :

1. Abstraksi (abstraction)

2. Pembungkusan (encapsulation)

3. Pewarisan (inheritance)

4. Polimorfisme (polymorphism)

Class

Definisi Class yaitu template untuk membuat Objek. Class merupakan prototipe atau blue print yang mendefinisikan variabel-variabel dan method-method secara umum. Objek merupakan hasil instances dari suatu Class. Proses pembentukan objek dari suatu Class disebut sebagai instances. Objek disebut juga sebagai instances.

Properti

Dalam Class, Atribut sering disebut sebagai Variabel. Atribut dibedakan menjadi dua jenis, yaitu Instance Variable dan Class Variable.

Instance Variable adalah Atribut untuk setiap Objek dari Class yang sama. Setiap objek mempunyai dan menyimpan nilai Atributnya sendiri. Jadi, setiap Objek dari Class yang sama boleh mempunyai nilai yang sama atau berbeda.

Method

Behavior atau tingkah laku adalah hal-hal yang bisa dilakukan oleh Objek dari suatu Class. Behavior dapat digunakan untuk mengubah nilai Atribut suatu Objek, menerima informasi dari Objek lain, dan mengirim informasi ke Objek lain untuk melakukan suatu tugas (task). Contohnya seperti pada gambar diatas terdapat Objek manusia memiliki Behavior/Method jalan, lari, lompat, marah dan sebagainya.

Encapsulation (Pembungkusan)

Salah satu ciri penting OOP adalah Encapsulaption. Definisi Encapsulation secara teoritis adalah pembungkusan variabel dan method dalam sebuah obyek yang terlindungi serta menyediakan interface untuk mengakses variabel tersebut. Variabel dan method yang dimiliki oleh suatu objek, bisa ditentukan hak aksesnya.

Inheritance (Pewarisan)

Inheritance merupakan pewarisan atribut dan method dari sebuah class ke class lainnya. Class yang mewarisi disebut superclass dan Class yang diwarisi disebut subclass. Subclass bisa berlaku sebagai superclass bagi class lainya, disebut sebagai multilevel inheritance.

Sebagai contoh terdapat class mobil sport dan mobil minibus . Mobil termasuk superclass. Mobil sport dan mobil minibus termasuk subclass. Hal ini dikarenakan mobil sport dan mobil minibus memiliki variabel dan method yang dimiliki oleh class mobil, yaitu sama-sama memiliki variabel/properti nomor kendaraan, nomor mesin dan memiliki memiliki method bisa berjalan, berbelok, mengerem.

Polimorfisme

Polimorfisme adalah kemampuan suatu objek untuk mempunyai lebih dari satu bentuk. Polimorfisme tidak bergantung kepada pemanggilan subrutin. Metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesan tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan 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.