KUIS-1 PEMROGRAMAN BERORIENTASI OBJEK

Poin total 88/100

Email responden (**ptrprtm16@gmail.com**) dicatat saat formulir ini dikirimkan.

0 dari 0 poin

TOKEN *

24092024

IDENTITAS 0 dari 0 poin

NIM

2341720061

NAMA

ESA PRATAMA PUTRI

SAYA AKAN MENGERJAKAN KUIS INI DENGAN JUJUR, APABILA SAYA TERBUKTI BERBUAT CURANG MAKA SAYA BERSEDIA DIANGGAP TIDAK PERNAH MENGIKUTI KUIS INI *

- (●) SETUJU
- ( ) TIDAK SETUJU

SOAL PILIHAN GANDA DENGAN MULTI RESPON (BISA LEBIH DARI 1 JAWABAN BENAR) 88 dari 100 poin

PILIHLAH SEMUA JAWABAN YANG ANDA ANGGAP BENAR

Manakah dari pernyataan berikut yang merupakan pilar utama dari Pemrograman Berorientasi Objek? 0/2

- [x] Enkapsulasi
- [x] Polimorfisme
- [x] Pewarisan (Inheritance)
- [ ] Objek
- [x] Abstraksi

Apa yang dimaksud dengan "objek" dalam konteks OOP? 2/2

- [ ] Sebuah kumpulan data terkait
- [ ] Sebuah blueprint untuk membuat objek
- [ ] Sebuah tipe data primitif
- [ ] Sebuah fungsi atau prosedur
- [x] Sebuah instance dari sebuah kelas

Konsep apakah yang memungkinkan kita untuk menyembunyikan detail implementasi dari pengguna? 2/2

- [ ] Abstraksi
- [ ] Inheritance
- [ ] Polimorfisme
- [x] Enkapsulasi
- [ ] Pewarisan

Apa tujuan utama dari penggunaan Pemrograman Berorientasi Objek? 0/2

- [ ] Meningkatkan reusabilitas kode.
- [ ] Mempermudah pemeliharaan kode
- [x] Memodelkan masalah dunia nyata dengan lebih baik.
- [ ] Meningkatkan kecepatan eksekusi program
- [ ] Mengurangi penggunaan memori

Manakah dari pernyataan berikut yang BUKAN merupakan keuntungan dari OOP? 2/2

- [ ] Meningkatkan fleksibilitas kode
- [ ] Meningkatkan reusabilitas kode
- [x] Meningkatkan kompleksitas kode
- [ ] Memudahkan pengujian kode
- [ ] Meningkatkan modularitas kode

Apa yang dimaksud dengan "class" dalam Java? 2/2

- [ ] Sebuah fungsi atau prosedur
- [x] Sebuah blueprint atau template untuk membuat objek.
- [ ] Sebuah kumpulan data terkait.
- [ ] Sebuah tipe data primitif
- [ ] Sebuah instance dari sebuah objek

Bagaimana cara Anda membuat sebuah objek dari sebuah class di Java? 2/2

- [ ] Menggunakan kata kunci "public"
- [ ] Menggunakan kata kunci "static"
- [ ] Menggunakan kata kunci "class"
- [x] Menggunakan kata kunci "new"
- [ ] Menggunakan kata kunci "this"

Apa yang dimaksud dengan "method" dalam Java? 2/2

- [ ] Sebuah blueprint untuk membuat objek
- [x] Sebuah blok kode yang melakukan tugas tertentu
- [ ] Sebuah kumpulan data terkait
- [ ] Sebuah tipe data primitif
- [ ] Sebuah instance dari sebuah class

Apa perbedaan antara "constructor" dan "method" biasa di Java? 2/2

- [ ] Constructor tidak memiliki nama, sedangkan method biasa memiliki nama.
- [x] Constructor digunakan untuk menginisialisasi objek saat dibuat, sedangkan method biasa digunakan untuk melakukan tindakan pada objek.
- [ ] Constructor dapat dipanggil secara langsung, sedangkan method biasa harus dipanggil melalui objek
- [ ] Constructor memiliki tipe kembalian, sedangkan method biasa tidak.
- [ ] Constructor dapat memiliki parameter, sedangkan method biasa tidak

Apa kata kunci yang digunakan untuk mendefinisikan sebuah class di Java? 2/2

- [ ] method
- [x] class
- [ ] object
- [ ] this
- [ ] new

Apa tujuan utama dari enkapsulasi dalam OOP? 2/2

- [ ] Meningkatkan reusabilitas kode
- [ ] Meningkatkan kecepatan eksekusi program
- [x] Menyembunyikan detail implementasi dari pengguna
- [ ] Mengurangi penggunaan memori
- [ ] Mempermudah pemeliharaan kode.

Bagaimana cara Anda dapat menerapkan konsep enkapsulasi di Java? 2/2

- [ ] Menggunakan kata kunci static
- [ ] Menggunakan kata kunci final
- [ ] Menggunakan kata kunci abstract
- [x] Menyediakan method getter dan setter untuk mengakses dan memodifikasi variabel.
- [x] Menggunakan modifier akses private untuk variabel dan method

Apa keuntungan dari menggunakan enkapsulasi? 2/2

- [x] Memudahkan perubahan implementasi tanpa mempengaruhi pengguna
- [ ] Meningkatkan kecepatan eksekusi program
- [ ] Mengurangi penggunaan memori
- [x] Meningkatkan keamanan data
- [x] Meningkatkan modularitas kode

Manakah dari modifier akses berikut yang memberikan tingkat akses paling terbatas di Java? 2/2

- [ ] public
- [ ] protected
- [x] private
- [ ] Tidak ada modifier (default)
- [ ] package

Apa yang dimaksud dengan "getter" dan "setter" dalam konteks enkapsulasi? 2/2

- [x] Method yang digunakan untuk mengakses dan memodifikasi variabel private secara terkontrol
- [ ] Method yang digunakan untuk menginisialisasi objek
- [ ] Method yang tidak memiliki tipe kembalian
- [ ] Method yang digunakan untuk melakukan tindakan pada objek
- [ ] Method yang dapat dipanggil secara langsung

Apa perbedaan antara "error" dan "exception" di Java? 2/2

- [ ] Error dan exception adalah hal yang sama
- [ ] Error adalah subclass dari Exception
- [x] Error adalah kondisi kritis yang biasanya tidak dapat dipulihkan, sedangkan exception adalah kondisi yang dapat ditangani oleh program
- [ ] Error terjadi saat kompilasi, sedangkan exception terjadi saat runtime
- [ ] Error dapat ditangani oleh program, sedangkan exception tidak

Manakah dari pernyataan berikut yang merupakan contoh dari "checked exception" di Java? 2/2

- [ ] ArithmeticException
- [x] IOException
- [x] SQLException
- [ ] RuntimeException
- [ ] NullPointerException

Bagaimana cara Anda menangani exception di Java? 2/2

- [x] Menggunakan blok try-catch-finally
- [ ] Menggunakan kata kunci throw
- [ ] Menggunakan kata kunci throws
- [ ] Menggunakan kata kunci final
- [x] Menggunakan blok try-catch

Apa tujuan dari blok finally dalam penanganan exception? 2/2

- [x] Kode di dalam blok finally akan selalu dieksekusi, baik terjadi exception atau tidak
- [ ] Blok finally hanya dieksekusi jika terjadi exception
- [ ] Blok finally hanya dieksekusi jika tidak terjadi exception
- [ ] Blok finally digunakan untuk melempar exception baru
- [ ] Blok finally digunakan untuk menangani exception yang spesifik

Kata kunci apa yang digunakan untuk melempar exception secara manual di Java? 2/2

- [ ] final
- [ ] catch
- [x] throw
- [ ] throws
- [ ] try

Apa fungsi utama dari UML Class Diagram? 2/2

- [ ] Memodelkan deployment sebuah sistem
- [ ] Memodelkan use case sebuah sistem
- [ ] Memodelkan aliran proses dalam sebuah sistem
- [ ] Memodelkan interaksi antara pengguna dan sistem
- [x] Memodelkan struktur dan hubungan antara kelas-kelas dalam sebuah sistem

Apa yang diwakili oleh sebuah kotak dalam UML Class Diagram? 2/2

- [ ] Sebuah use case
- [x] Sebuah kelas
- [ ] Sebuah hubungan
- [ ] Sebuah method
- [ ] Sebuah objek

Bagaimana cara Anda merepresentasikan hubungan pewarisan (inheritance) dalam UML Class Diagram? 0/2

- [ ] Menggunakan garis putus-putus
- [x] Menggunakan garis dengan panah terbuka mengarah ke kelas induk
- [ ] Menggunakan kata kunci Include
- [ ] Menggunakan kata kunci Extends
- [ ] Menggunakan garis dengan panah tertutup mengarah ke kelas

Apa yang terjadi jika Anda mencoba mengakses variabel private dari luar kelas yang mendefinisikannya? 2/2

- [ ] Variabel akan diubah menjadi public secara otomatis
- [x] Akan terjadi error kompilasi
- [ ] Program akan berjalan normal
- [ ] Program akan berjalan tanpa error
- [ ] Akan terjadi error runtime

Mengapa penting untuk menggunakan enkapsulasi saat mengembangkan aplikasi yang kompleks? 2/2

- [x] Meningkatkan keamanan data dengan membatasi akses langsung
- [x] Meningkatkan modularitas dan pemahaman kode
- [x] Memudahkan pengelolaan perubahan pada kode
- [ ] Mengurangi penggunaan memori
- [ ] Meningkatkan kecepatan eksekusi program

Dalam enkapsulasi, apa peran dari modifier akses protected? 0/2

- [ ] Memungkinkan akses hanya dari kelas itu sendiri
- [x] Memungkinkan akses dari semua kelas dalam paket yang sama dan subclass di paket berbeda
- [ ] Memungkinkan akses hanya dari kelas dalam paket yang sama
- [ ] Memungkinkan akses hanya dari subclass, baik dalam paket yang sama maupun berbeda
- [ ] Memungkinkan akses dari semua kelas dalam program

Bagaimana Anda bisa mengakses variabel private dari subclass yang berada di paket yang berbeda? 2/2

- [x] Menggunakan method getter yang disediakan oleh kelas induk
- [ ] Langsung mengaksesnya menggunakan nama variabel
- [ ] Langsung mengaksesnya menggunakan nama variabel disertai kata kunci static
- [ ] Menggunakan modifier akses protected pada variabel
- [ ] Tidak bisa, variabel private tidak bisa diakses dari subclass di paket berbeda

Apa perbedaan antara enkapsulasi dan abstraksi dalam OOP? 0/2

- [ ] Enkapsulasi menggunakan modifier akses, sedangkan abstraksi menggunakan interface
- [ ] Enkapsulasi dan abstraksi adalah konsep yang sama
- [ ] Enkapsulasi berfokus pada data, sedangkan abstraksi berfokus pada perilaku
- [ ] Enkapsulasi berfokus pada perilaku, sedangkan abstraksi berfokus pada data
- [x] Enkapsulasi menyembunyikan detail implementasi, sedangkan abstraksi menyembunyikan kompleksitas

Apa yang dimaksud dengan "unchecked exception" di Java? 2/2

- [x] Exception yang tidak perlu ditangani secara eksplisit, tetapi bisa terjadi saat runtime
- [ ] Exception yang disebabkan oleh kesalahan programmer
- [ ] Exception yang terjadi saat kompilasi
- [ ] Exception yang harus ditangani secara eksplisit menggunakan try-catch atau dinyatakan dengan throws
- [ ] Exception yang harus ditangani secara eksplisit menggunakan throws

Apa yang terjadi jika sebuah exception tidak ditangani oleh program Java? 0/2

- [ ] Exception akan ditangani oleh sistem operasi
- [x] Program akan berhenti secara abnormal dan menampilkan pesan error
- [ ] Exception akan diabaikan
- [ ] Program akan melanjutkan eksekusi secara normal
- [ ] Program tidak akan melanjutkan eksekusi secara normal

Apa keuntungan dari menggunakan mekanisme exception handling di Java? 2/2

- [x] Memungkinkan program untuk menangani error secara terstruktur
- [x] Memisahkan logika penanganan error dari logika bisnis utama
- [ ] Meningkatkan kecepatan eksekusi program
- [ ] Mengurangi penggunaan memori
- [x] Meningkatkan robustness program dengan mencegah crash mendadak

Apa kata kunci yang digunakan untuk mengakses anggota (atribut atau method) dari objek itu sendiri di dalam method? 2/2

- [ ] self
- [ ] new
- [x] this
- [ ] static
- [ ] class

Apa yang dimaksud dengan "static method" di Java? 2/2

- [ ] Method yang harus di-override oleh subclass
- [ ] Method yang selalu mengembalikan nilai null
- [x] Method yang terkait dengan kelas itu sendiri, bukan dengan objek tertentu
- [ ] Method yang harus memiliki parameter
- [ ] Method yang tidak bisa memiliki parameter

Apa perbedaan antara "pass by value" dan "pass by reference" dalam konteks pemanggilan method di Java? 2/2

- [ ] Pass by value dan pass by reference adalah konsep yang sama di Java
- [ ] Pass by value mengubah nilai argumen asli, sedangkan pass by reference tidak
- [x] Pass by value menyalin nilai argumen ke parameter method, sedangkan pass by reference menyalin referensi objek ke parameter method
- [ ] Pass by value hanya bisa digunakan untuk tipe data primitif, sedangkan pass by reference hanya bisa digunakan untuk objek
- [ ] Pass by value dan pass by reference adalah konsep yang tidak dikenal di Java

Apa yang dimaksud dengan "method signature" di Java? 2/2

- [ ] Modifier akses dari method (public, private, protected)
- [ ] Komentar yang menjelaskan fungsi method
- [ ] Isi dari method, termasuk semua pernyataan di dalamnya
- [x] Nama method, tipe kembalian, dan daftar parameter
- [ ] Daftar parameter dari method, termasuk semua pernyataan di dalamnya

Selain private, modifier akses apa lagi yang bisa digunakan untuk mendukung enkapsulasi di Java? 2/2

- [ ] final
- [ ] package
- [x] protected
- [ ] public
- [ ] static

Kapan Anda harus menggunakan blok try-catch-finally daripada hanya menggunakan try-catch? 2/2

- [ ] Ketika Anda ingin menangani beberapa jenis exception yang sejenis
- [x] Ketika ada kode yang harus dieksekusi terlepas dari apakah terjadi exception atau tidak, misalnya menutup file atau koneksi database
- [ ] Ketika Anda ingin menangani beberapa jenis exception secara berbeda
- [ ] Ketika Anda tidak yakin jenis exception apa yang mungkin terjadi
- [ ] Ketika Anda ingin melempar exception baru dari dalam blok catch

Bagaimana cara Anda merepresentasikan atribut dan method dalam UML Class Diagram? 2/2

- [ ] Atribut ditulis di bagian atas kotak kelas
- [ ] Atribut dan method hanya dituliskan namanya saja, tanpa tipe data atau parameter
- [ ] Atribut ditulis dengan tanda + (public), - (private), atau # (protected), sedangkan method tidak memiliki tanda
- [ ] Method ditulis di bagian atas kotak kelas
- [x] Atribut ditulis di bagian tengah kotak kelas, sedangkan method ditulis di bagian bawah kotak kelas

Selain Class Diagram, apa saja jenis diagram lain yang termasuk dalam UML? 2/2

- [x] Use Case Diagram, Sequence Diagram, Activity Diagram, State Machine Diagram, dan lainnya
- [ ] Flowchart
- [ ] Gantt Chart, PERT Chart
- [ ] UML hanya terdiri dari Class Diagram
- [ ] ERD (Entity Relationship Diagram) dan DFD (Data Flow Diagram)

Apa perbedaan antara variabel instance dan variabel kelas di Java? 2/2

- [x] Variabel instance dimiliki oleh setiap objek dari kelas, sedangkan variabel kelas dimiliki oleh kelas itu sendiri dan dibagi oleh semua objek
- [ ] Variabel instance dideklarasikan di dalam method, sedangkan variabel kelas dideklarasikan di luar method
- [ ] Variabel instance bisa diakses dari method static, sedangkan variabel kelas tidak bisa
- [ ] Variabel instance dan variabel kelas adalah konsep yang sama
- [ ] Variabel variabel bisa diakses dari method static, sedangkan instance kelas tidak bisa

Apa kata kunci yang digunakan untuk mendeklarasikan konstanta di Java? 2/2

- [x] final
- [ ] static
- [ ] private
- [ ] this
- [ ] public

Bagaimana cara Anda merepresentasikan visibilitas (public, private, protected) dari atribut dan method dalam UML Class Diagram? 2/2

- [ ] Menggunakan warna berbeda untuk setiap tingkat visibilitas
- [ ] Visibilitas tidak direpresentasikan dalam UML Class Diagram
- [ ] Menggunakan garis tebal untuk public, garis tipis untuk private, dan garis putus-putus untuk protected
- [x] Menggunakan tanda + (public) atau - (private) di depan nama atribut atau method
- [x] Menggunakan tanda # (protected) atau ~ (package) di depan nama atribut atau method

Apa keuntungan menggunakan UML Class Diagram dalam pengembangan perangkat lunak? 2/2

- [x] Mendukung proses desain dan dokumentasi perangkat lunak
- [ ] Meningkatkan kecepatan eksekusi program
- [x] Memudahkan komunikasi dan kolaborasi antara anggota tim
- [ ] Mengurangi penggunaan memori
- [x] Membantu memvisualisasikan struktur dan hubungan antara kelas

Apa peran dari aksesor (getter) dan mutator (setter) dalam enkapsulasi? 2/2

- [ ] Aksesor dan mutator adalah konsep yang sama
- [x] Aksesor digunakan untuk membaca nilai variabel private, sedangkan mutator digunakan untuk mengubah nilai variabel private
- [ ] Aksesor digunakan untuk mengubah nilai variabel private, sedangkan mutator digunakan untuk membaca nilai variabel private
- [ ] Aksesor dan mutator hanya bisa digunakan pada variabel public
- [ ] Aksesor digunakan untuk membaca dan mengubah nilai variabel private, sedangkan mutator digunakan hanya untuk membaca nilai variabel private

Mengapa penting untuk menggunakan aksesor dan mutator meskipun variabel sudah dideklarasikan sebagai private? 2/2

- [ ] Mengurangi penggunaan memori
- [x] Memberikan kontrol lebih terhadap akses dan modifikasi data
- [x] Memudahkan perubahan implementasi tanpa mempengaruhi pengguna kelas
- [x] Memungkinkan validasi data sebelum mengubah nilai
- [ ] Meningkatkan kecepatan eksekusi program

Apa yang terjadi jika Anda menempatkan beberapa blok catch setelah blok try? 2/2

- [ ] Akan terjadi error kompilasi
- [ ] Semua blok catch tidak akan dieksekusi, apapun jenis exception-nya
- [ ] Semua blok catch akan dieksekusi, terlepas dari jenis exception-nya
- [ ] Hanya blok catch terakhir yang akan dieksekusi
- [x] Blok catch akan dievaluasi berurutan, dan hanya blok pertama yang cocok dengan jenis exception yang akan dieksekusi

Apa perbedaan antara finally dan final di Java? 2/2

- [x] finally adalah blok kode yang selalu dieksekusi setelah blok try-catch, sedangkan final adalah modifier yang digunakan untuk mencegah perubahan nilai variabel, method overriding, atau pewarisan kelas
- [ ] finally dan final adalah kata kunci yang sama
- [ ] finally digunakan untuk melempar exception, sedangkan final digunakan untuk menangani exception
- [ ] finally hanya bisa digunakan dengan try-with-resources, sedangkan final bisa digunakan di mana saja
- [ ] final adalah blok kode yang selalu dieksekusi setelah blok try-catch, sedangkan finally adalah modifier yang digunakan untuk mencegah perubahan nilai variabel, method overriding, atau pewarisan kelas

Apa saja best practice dalam penanganan exception di Java? 2/2

- [x] Hindari penggunaan exception untuk alur kontrol normal
- [x] Gunakan custom exception untuk kasus-kasus khusus di aplikasi Anda
- [x] Tangani hanya exception yang bisa Anda pulihkan
- [x] Berikan pesan error yang jelas dan informatif
- [ ] Selalu gunakan blok try-catch untuk setiap baris kode

Apa yang harus Anda lakukan jika terjadi Error di Java? 2/2

- [ ] Menangani error tersebut menggunakan try-catch
- [x] Memperbaiki masalah mendasar yang menyebabkan error, karena biasanya Error tidak bisa dipulihkan
- [ ] Mengabaikan error tersebut
- [ ] Melempar error tersebut ke method pemanggil
- [ ] Menangani error tersebut menggunakan try-catch-finally

Apa saja teknik-teknik untuk mencegah terjadinya exception di Java? 2/2

- [x] Menggunakan try-with-resources untuk memastikan resource ditutup dengan benar
- [x] Menggunakan pernyataan if untuk memeriksa kondisi sebelum melakukan operasi yang berpotensi menimbulkan exception
- [x] Validasi input dari pengguna
- [ ] Menggunakan pernyataan while untuk mengulang input bila terjadi error
- [ ] Mengabaikan semua potensi error

PENOLAKAN 0 dari 0 poin

ANDA TIDAK DIPERKENANKAN MENGIKUTI KUIS INI KARENA TIDAK MENYETUJUI PERNYATAAN KESANGGUPAN UNTUK JUJUR DALAM MENGERJAKAN SOAL KUIS INI

PENUTUP 0 dari 0 poin

PASTIKAN SEMUA SOAL SUDAH ANDA JAWAB DENGAN BENAR

APAKAH ANDA YAKIN UNTUK MENYIMPAN PERMANEN JKAWABAN ANDA?

- (●) YA
- ( ) TINJAU ULANG JAWABAN SAYA