

Nama: Muhammad Kensya Kussyahputra Hidayatullah

NIM: 245150707111047

## A. Static Method

### Pertanyaan

1. Apakah yang disebut dengan static variabel? Dan apa fungsi dari static variabel serta kapan kita dapat menggunakan static variabel?
2. Mengapa pada main method harus dituliskan static? Jelaskan jawaban anda beserta dengan alasan!
3. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!
4. Jika pada tubuh method hitungPenjumlahan ditambahkan syntax hitungPerkalian(a,b) apa yang terjadi? Jelaskan?
5. Jika pada tubuh method hitungPerkalian ditambahkan syntax hitungPenjumlahan(a,b) apa yang terjadi? Jelaskan?
6. Tambahkan method non static dengan nilai balikan double untuk menghitung pembagian dengan parameter String nil dan String nil2, dan panggil method tersebut pada method main!

### Jawaban

1. Static variabel adalah variabel yang terikat pada kelas itu sendiri, bukan pada objek individual. Hanya ada satu salinan variabel ini yang dibagikan oleh seluruh instance kelas. Fungsinya utama adalah untuk berbagi data antar objek, menyimpan konstanta kelas, mengelola status global yang terkait dengan kelas, dan terkadang berperan dalam implementasi pola Singleton. Kita menggunakan static variabel ketika nilai variabel perlu sama untuk semua objek kelas, untuk mendefinisikan konstanta kelas, atau ketika kita ingin mempertahankan nilai di antara pembuatan objek yang berbeda dari kelas yang sama
2. Main method harus static di Java karena JVM (Java Virtual Machine) perlu mengeksekusinya sebagai titik awal program tanpa harus membuat objek dari kelas yang memuatnya terlebih dahulu. Sifat static memungkinkan JVM memanggil main langsung melalui nama kelas, karena pada awal eksekusi belum ada objek yang tersedia. Dengan demikian, static memastikan JVM memiliki akses langsung untuk memulai jalannya aplikasi Jav
3. (Dalam kode)
4. Jika pada method hitungPenjumlahan ditambahkan syntax hitungPerkalian(a,b) maka setiap kali method hitungPenjumlahan dipanggil maka juga akan memanggil

method `hitungPerkalian` dengan parameter adalah parameter yang dimasukkan ke method `hitungPenjumlahan`. Program akan mencetak hasil perkalian a dan b ke konsol sebelum kemudian menghitung dan mencetak hasil penjumlahan a dan b

5. Jika pada method `hitungPerkalian` ditambahkan syntax `hitungPenjumlahan(a,b)` akan terjadi error karena “non-static method `hitungPenjumlahan(int,int)` cannot be referenced from a static context”. Karena method `hitungPerkalian` bersifat static dan method `hitungPenjumlahan` bersifat non-static, dalam java method static tidak dapat mengakses method non-static secara langsung tanpa membuat objek dari kelas tersebut. Oleh karena itu, untuk memanggil `hitungPenjumlahan` dari dalam `hitungPerkalian`
6. (Dalam kode)

## B. Konstanta Final

Pertanyaan

1. Benahi kode `Vehicle1` dan `TestVehicle1` dan perbaiki jika menemui kesalahan!
2. Hapus separator “/” pada file `Vehicle1.java` pada baris 4-6 serta pada file `TestVehicle1.java` pada baris 6, apa yang terjadi dan jelaskan!
3. Pada file `Vehicle1.java` variabel `load` ubah menjadi konstanta final, apa yang terjadi, jelaskan!
4. Tambahkan keyword “static” pada file `Vehicle1.java` variabel `maxLoad`, apa yang terjadi dan jelaskan!

Jawaban

1. (Dalam kode)
2. Jika pada `Vehicle1.java` pada baris 4-6 serta pada file `TestVehicle1.java` pada baris 6 separator “/” dihilangkan maka akan terjadi error karena kalau pada file `Vehicle1.java` errornya terjadi pada `this.maxLoad = max;` yang merubah `maxLoad` padahal `maxLoad` adalah variabel final sehingga tidak bisa dirubah. Kalau pada file pada file `TestVehicle1.java` terjadi error karena pada saat melakukan instansiasi dengan `Vehicle1 vehicle2 = new Vehicle1(1000);` tidak ada konstuktor yang menerima parameter di file `Vehicle1`
3. Akan terjadi error karena pertama pada awalnya `load` tidak di inisialisasi dan akan ada error lagi pada method `addBox` merubah nilai `load` dengan `this.load = temp;`, jika `load` bersifat final maka nilai tersebut tidak dapat diubah setelah diinisialisasi, sehingga akan menghasilkan error

4. Jika keyword static ditambahkan pada variabel maxLoad di Vehicle1.java, maka maxLoad akan menjadi milik kelas, bukan milik setiap objek. Ini berarti semua instansi dari Vehicle1 akan berbagi satu salinan maxLoad, yang dapat diakses tanpa membuat objek, misalnya dengan Vehicle1.maxLoad. Jika maxLoad tetap final, nilainya tidak dapat diubah. Perubahan ini akan mengurangi fleksibilitas, tetapi berguna jika Anda ingin semua objek memiliki nilai maksimum yang sama