**Java Memory:**

**Stack vs Heap?**

**Apa Itu?**

**Stack** dan **Heap** adalah dua area utama memori dalam Java yang digunakan oleh Java Virtual Machine (JVM) untuk menyimpan objek dan data. Berikut adalah perbedaan antara Stack dan Heap:

1. Stack: Merupakan area memori yang digunakan untuk menyimpan variabel lokal, argumen method, dan alamat kembali dari method. Elemen-elemen dalam stack dapat ditemukan dengan cepat karena mereka diproses dengan cara **LIFO (Last In, First Out)**. Stack juga memiliki ukuran yang terbatas dan bisa mengalami **overflow** jika terlalu banyak elemen yang ditempatkan di dalamnya.
2. Heap: Merupakan area memori yang digunakan untuk menyimpan objek dan data yang dapat dibagikan oleh banyak method. Elemen-elemen dalam heap tidak ditemukan dengan cepat karena mereka tidak diorganisasi dengan cara tertentu. Ukuran heap tidak terbatas dan tidak ada risiko overflow selama memori fisik tersedia.

Dalam konsep objek-orientasi, objek ditempatkan dalam **heap**, dan referensi objek ditempatkan dalam **stack**. Ini memungkinkan objek dapat digunakan oleh banyak method dan mempertahankan nilainya meskipun method yang memanggil objek telah selesai dieksekusi.

“ If the **stack**, there is no space for creating the new objects, it throws the  java.lang.StackOverFlowError,

If the **heap** space is full, it throws the java.lang.OutOfMemoryError. “

- *javapoint*

**Dalam Java:**

Dalam Java, ada If there is no space for creating the new objects, it throws the java.lang.StackOverFlowError. beberapa tipe data yang dapat ditempatkan pada memori stack dan heap. Berikut ini adalah beberapa tipe data yang dapat ditempatkan pada memori stack dan heap:

1. **Memori Stack:**

1. Primitive data types seperti int, float, double, char, dan boolean.
2. Referensi dari objek yang tidak memiliki referensi tambahan.

2. **Memori Heap:**

1. Objek yang memiliki referensi tambahan.
2. Array[], atau Array dalam objek yang besar.
3. Objek dari kelas abstrak dan interface.

**Catatan**: Semua objek yang ditempatkan dalam memori heap memiliki sebuah referensi, sedangkan variabel primitive hanya memiliki nilai.

*Sumber: - CharGPT*

*- Kelas Terbuka (youtube) didalam playlist java dasar – pengenalan array*

*- JavaPoint (artikel)*