

1. Saya tidak melihat method **createList** di dalam code **PairSum.java**, jadi mungkin saya asumsikan pertanyaan ini merujuk pada Hashmap pada variable **freqMap** yang digunakan dalam method **countPairs**

Hashmap dialokasikan di heap. Hashmap ini akan menyimpan pasangan key-value dari data type Integer.

Hashmap dalam internal mengelola alokasi memori untuk entri-entri yang disimpannya. Saat data baru ditambahkan menggunakan **freqMap.put()** object Integer baru mungkin dibuat dan disimpan di heap. Hashmap menggunakan array untuk menyimpan entri-entri baru. Jika jumlah entri melebihi kapasitas, Hashmap akan resizing (memperbesar ukurannya) dengan membuat array baru yang lebih besar dan menyalin entri-entri yang lama, yang juga akan menambah alokasi memory.

Ref: [Understanding Memory Occupation of a HashMap in Java - CodingTechRoom](#)

2. Ketika method **countPairs** selesai dieksekusi, variable **freqMap** yang ada didalam stack akan keluar dari scope, akibatnya tidak ada lagi referensi yang menunjuk ke object Hashmap yang berada di heap. Hashmap beserta semua value di disimpannya menjadi unreachable.

Garbage collector akan mendeteksi bahwa Hashmap dan objek-objek terkaitnya tidak lagi memiliki referensi yang aktif dan akan membebaskan memory yang mereka sudah tempati. Memory tersebut yang kemudian nantinya dapat digunakan kembali.

3. Tidak, jadi tidak ada potensi memory leak atau kebocoran memory di dalam code **Pairsum.java**

Memory leak terjadi Ketika objek-objek yang tidak lagi digunakan oleh app masih memiliki referensi yang aktif sehingga garbage collector tidak dapat membebaskannya.

Seperti yang sudah dijelaskan jawaban nomor 2, setelah method **countPairs** telah dieksekusi, **freqMap** dan semua referensi internalnya akan keluar dari scope. Garbage collector akan mendeteksi bahwa **freqMap** dan objek-objek terkaitnya tidak lagi memiliki referensi yang aktif dan akan membebaskan memory yang mereka sudah gunakan.