A logo of a university

Description automatically generated

NAMA : RAKASYAEL HIZKIA KOLONDAM

NIM : 2023105490

MATA KULIAH : TI0091 – PRAKTIKUM DATA TERSTRUKTUR

PRAKTIKUM KE - : 10

|  |  |  |
| --- | --- | --- |
| **No.** | **Kode** | **Penjelasan** |
| 1 | public void printGraph()  {  System.out.println(adjList);  } | Merupakan metode yang mencetak seluruh isi pada graph dengan menggunakan variabel bernama adjList yang merupakan implementasi dari hashmap. Sehingga pada dasarnya, graph ditampilkan menggunakan hashmap karena penyimpanan vertex dan edges pada graph dilakukan di dalam hashmap. |
| 2 | public boolean addVertex(String vertex) {  if (adjList.get(vertex) == null)  {  adjList.put(vertex, new ArrayList<String>());  return true;  }  return false;  } | Add vertex dilakukan dengan cara pengecekan pada setiap vertex yang sudah disimpan dalam adjList. Jika vertex dalam adjList masih kosong, maka akan dilakukan penambahan Vertex dengan cara menggunakan fungsi dari Hashmap (java.util.Hashmap) yaitu put atau menaruh key dan value. Key disini adalah vertex itu sendiri (Contoh A) dan values dari vertex merupakan setiap vertices yang terhubung langsung dengan vertex yang menjadi kunci. Lalu fungsi akan mengembalikan nilai true jika vertex yang dimasukkan belum ada dan akan dibuat baru, jika sudah ada dan terjadi duplikasi, maka akan return false. |
| 3 | public boolean removeEdge(String vertex1, String vertex2) {  if (adjList.get(vertex1) != null && adjList.get(vertex2) != null)  {  adjList.get(vertex1).remove(vertex2);  adjList.get(vertex2).remove(vertex1);  return true;  }  return false;  } | Metode ini akan menghapus edges pada graph dengan mengimplementasikan fungsi dari Hashmap. Pertama akan terjadi pengecekan apabila vertex pertama ada dalam graph atau tidak, dan vertex kedua ada dalam graph atau tidak, jika keduanya ada maka akan dihapus dari masing-masing list vertex yang terhubung (A ke B maka list edges A akan menghapus B, begitu juga sebaliknya.) |
| 4 | public boolean **removeVertex**(String vertex) {  *// Modifikasi bagin ini, return value dapat diubah*  *if*(adjList.**get**(vertex) **==** null){  *return* false;      }  *for*(ArrayList<String> x *:* adjList.**values**()){        x.**remove**(vertex);      }        adjList.**remove**(vertex);  *return* true;    } | Metode ini akan menghapus vertex yang dicari. Pertama akan dilakukan pengecekan dari vertex. Jika vertex yang ingin dihapus tidak ada maka akan langsung return false. Jika tidak, maka akan lanjut menggunakan for each loop dimana setiap vertex yang terhubung dengan vertex yang ingin dihapus, akan menghapus dari list nya masing-masing. Setelah itu akan menghapus vertex itu sendiri (A akan dihapus dari list setiap yang terhubung langsung dengan A. Jika sudah maka A itu sendiri akan dihapus sehingga key A sudah tidak ada lagi) lalu akan return true. |