

```
1 // Tokenisasi menghitung jumlah kata unik
2 // Hasil sudah di-sort
3
4 import java.io.*;
5 import java.util.Map;
6 import java.util.Map.Entry;
7 import java.util.HashMap;
8 import java.util.StringTokenizer;
9 import java.util.LinkedHashMap;
10 import java.util.LinkedList;
11 import java.util.List;
12 import java.util.Iterator;
13 import java.util.Collections;
14 import java.util.Comparator;
15
16 public class tokenwordsort {
17     static String msg = "Harga minyak sekarang turun.
... Sebelumnya harga minyak naik. Sekarang harga minyak turun
... lagi. Kalau sudah turun (bisa juga tetap), harga minyak akan
... naik lagi. Harga minyak naik dan naik lagi.";
18
19
20 // Method untuk sorting data HashMap
21 //
22     private static HashMap<String, Integer>
... sortHashMap(HashMap<String, Integer> unsortMap, final boolean
... order) {
23         List<Entry<String, Integer>> list = new
... LinkedList<Entry<String, Integer>>(unsortMap.entrySet());
24
25         // Sorting the list based on values
26         Collections.sort(list, new Comparator<Entry<String,
... Integer>>()
27             {
28             public int compare(Entry<String, Integer> o1,
29                 Entry<String, Integer> o2)
30                 {
31                 if (order)
32                 {
33                 return
... o1.getValue().compareTo(o2.getValue());
```

```
34         }
35         else
36         {
37             return
... o2.getValue().compareTo(o1.getValue());
38
39         }
40     }
41 });
42
43     // Maintaining insertion order with the help of
... LinkedList
44     HashMap<String, Integer> sortedMap = new
... LinkedHashMap<String, Integer>();
45     for (Entry<String, Integer> entry : list)
46     {
47         sortedMap.put(entry.getKey(), entry.getValue());
48     }
49
50     return sortedMap;
51 }
52
53     public static void main(String a[]){
54         HashMap<String, Integer> TF = new HashMap<String,
... Integer>();
55         StringTokenizer st = new
... StringTokenizer(msg.toLowerCase(), ":", "(", ". ");
56         String term;
57         Integer freq;
58         while(st.hasMoreTokens()) {
59             term = st.nextToken();
60             if (!(TF.containsKey(term))) {
61                 TF.put(term, 0);
62             }
63             freq = TF.get(term);
64             TF.put(term, freq+1);
65         }
66         HashMap<String, Integer> sortTF = new HashMap<String,
... Integer>();
67         sortTF = sortHashMap(TF, false);
68         for(String key : sortTF.keySet()) {
```

```
69         System.out.println(key+" "+sortTF.get(key));  
70     }  
71 }  
72 }  
73
```