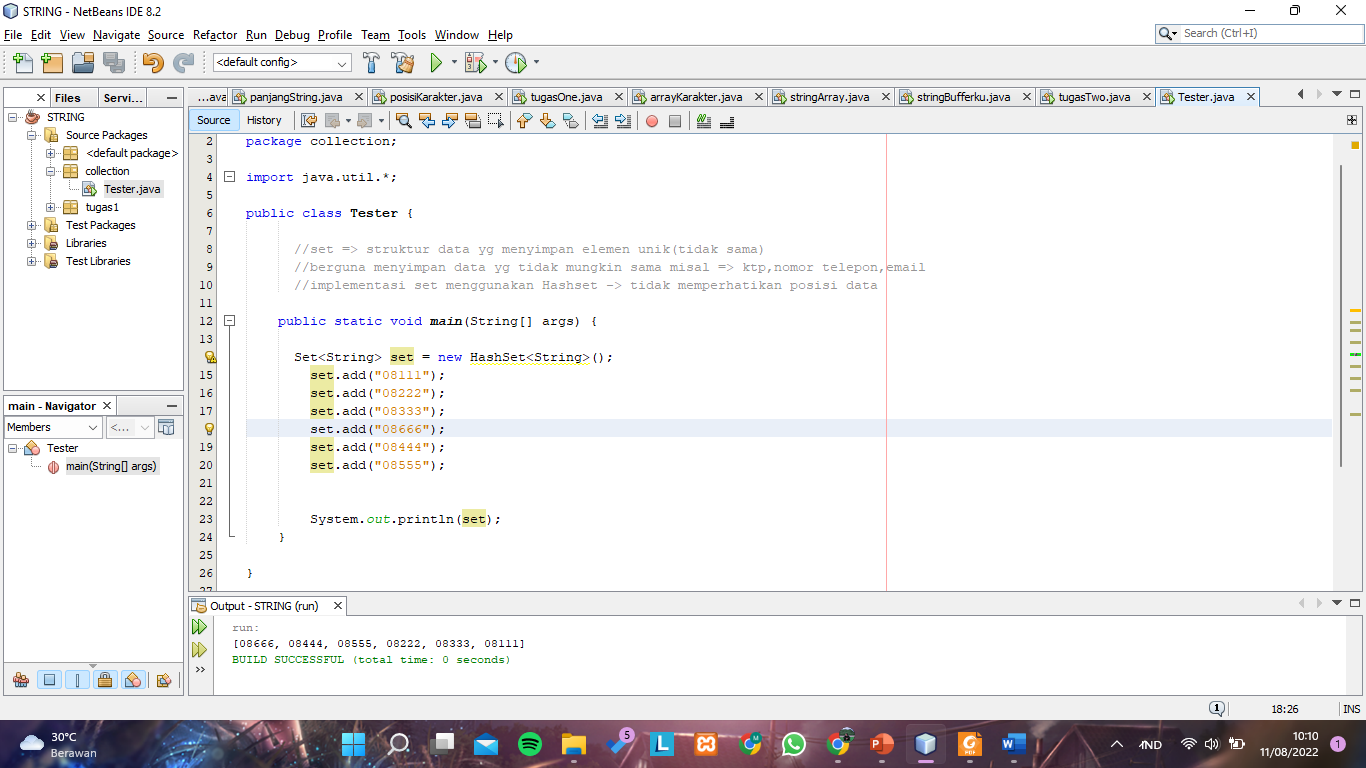
***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_collection\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**Nama : Muhamad Wahyu Saputra**

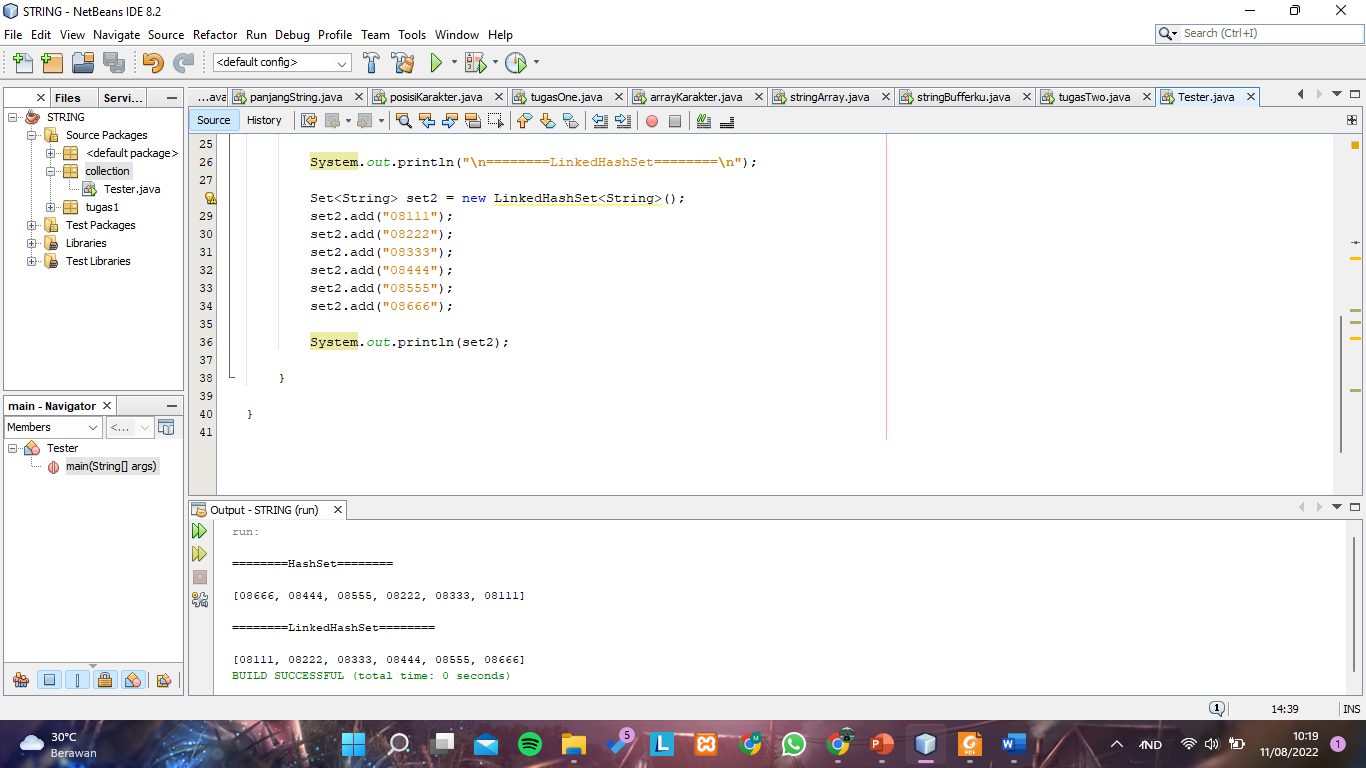
**Kelas : XII RPL B**

**Absen : 34**

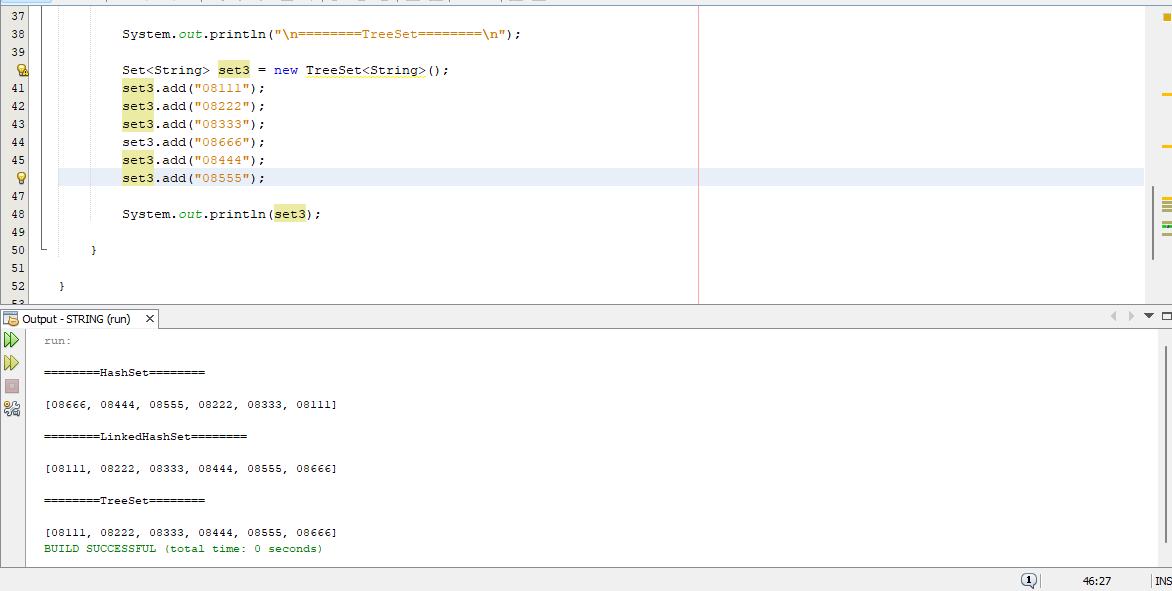
**1.Latihan SET**

 1.HashSet

Berdasarkan program&output diatas menggunakan HashSet hasil outputnya adalah acak

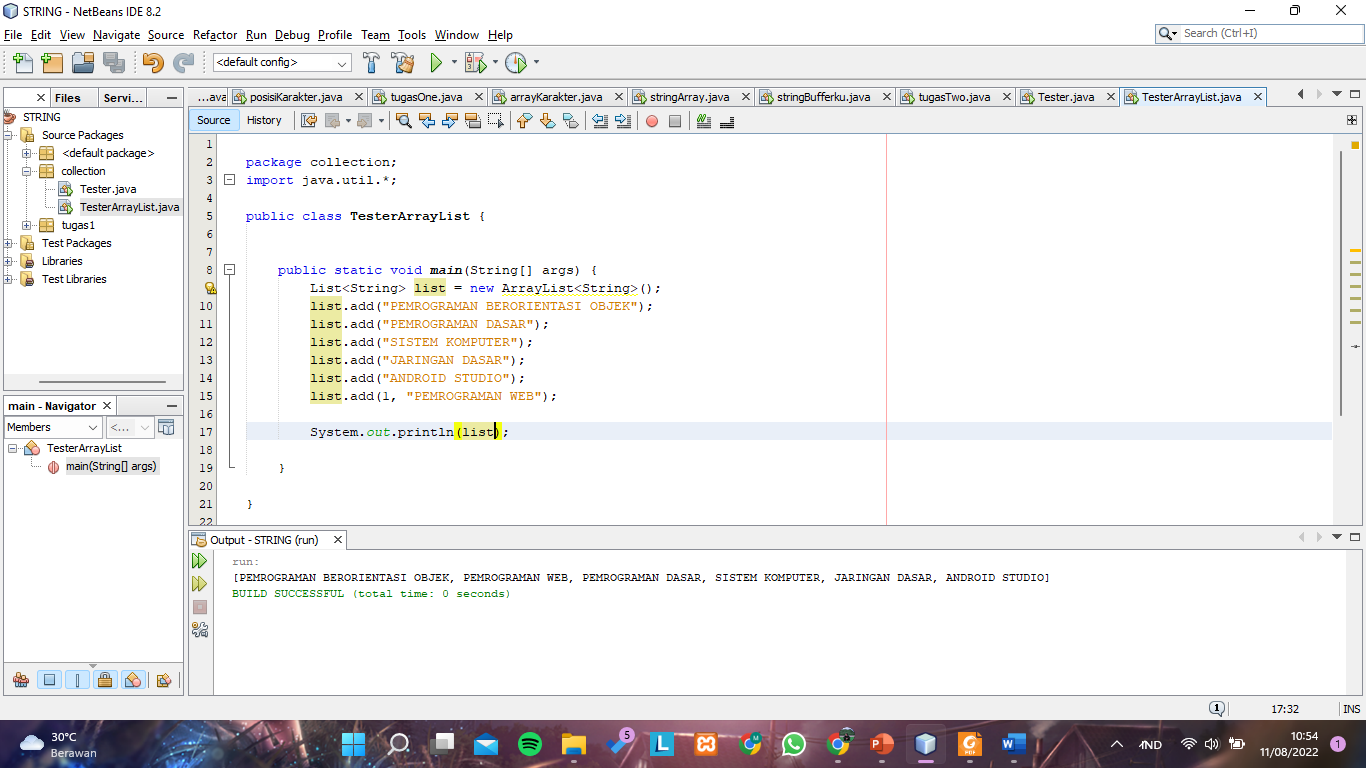
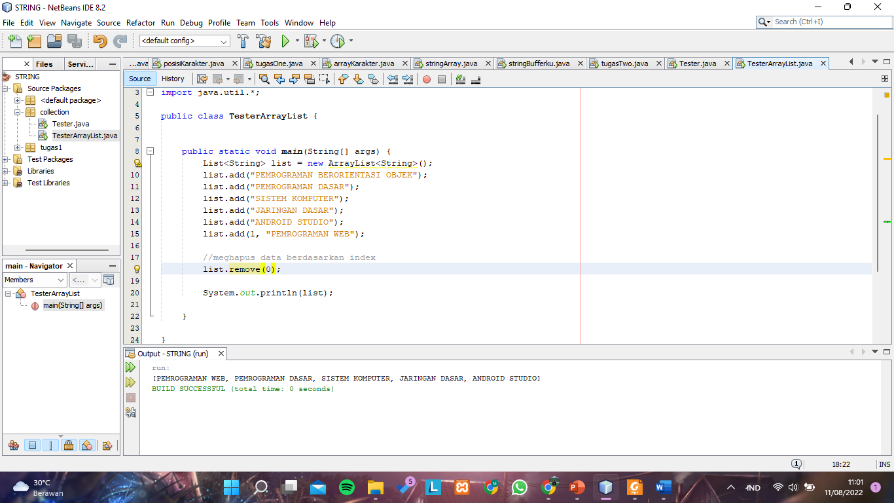
 2.LinkedHashSet

Jika menggunakan LinkedHashSet Akan urut sesuai inputan

3.TreeSet

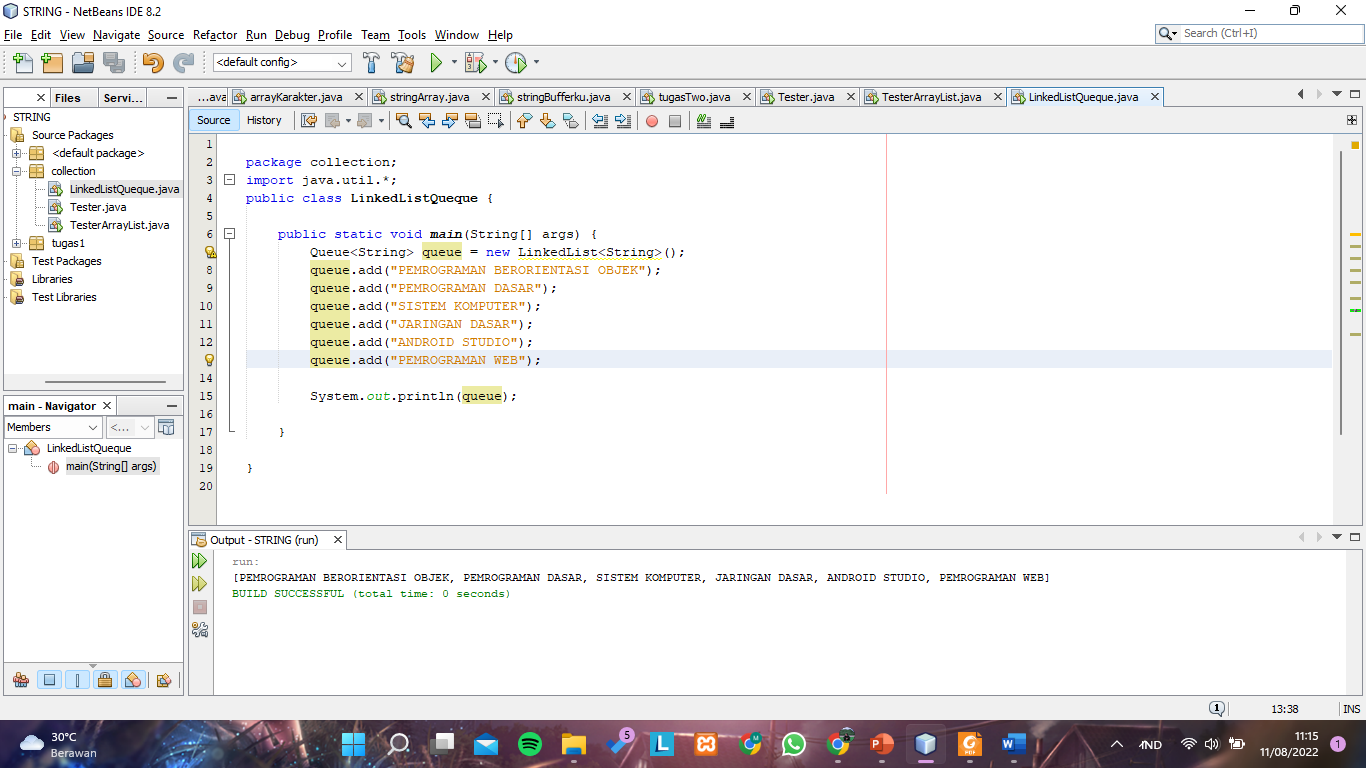
Ascending yaitu nilai diurutkan dari yg terkecil sampai yg terbesar jika huruf A -Z

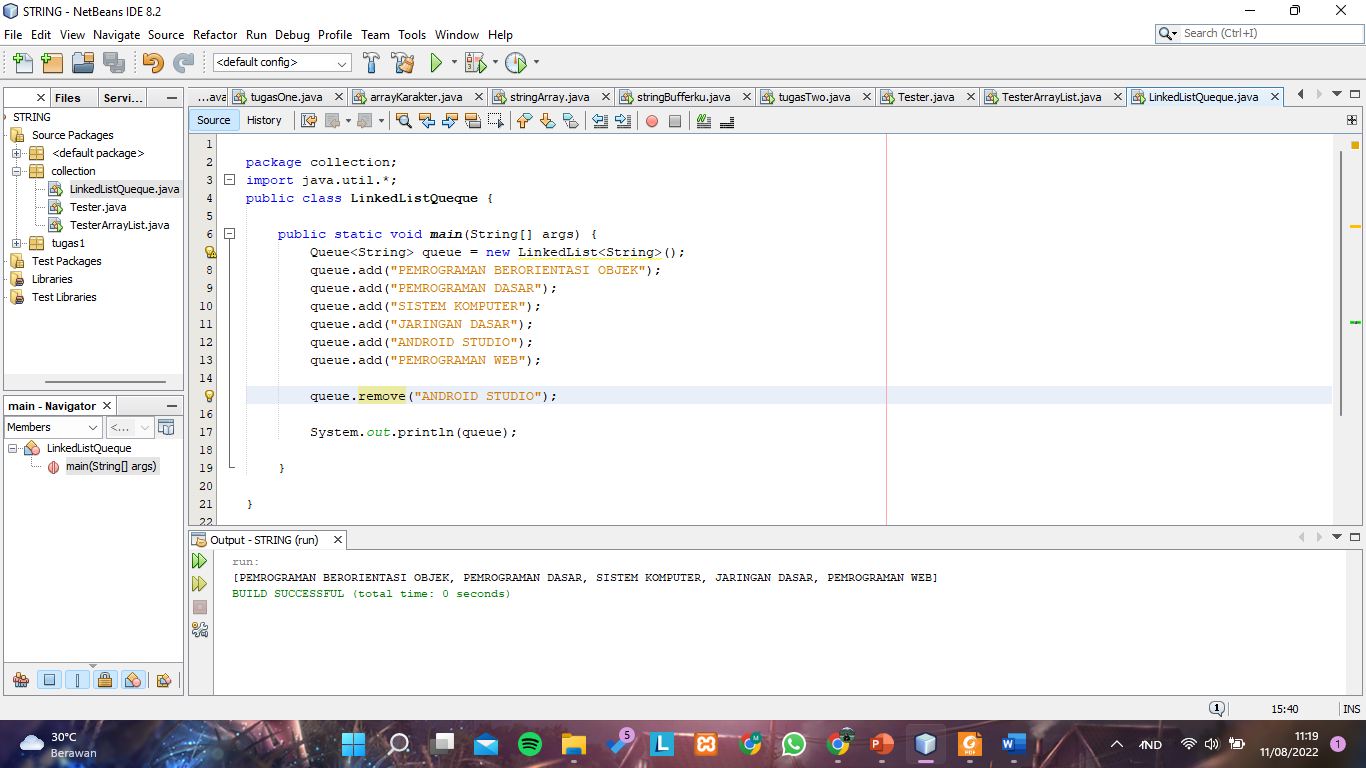
**2.Latihan LIST**

 1.ArrayList

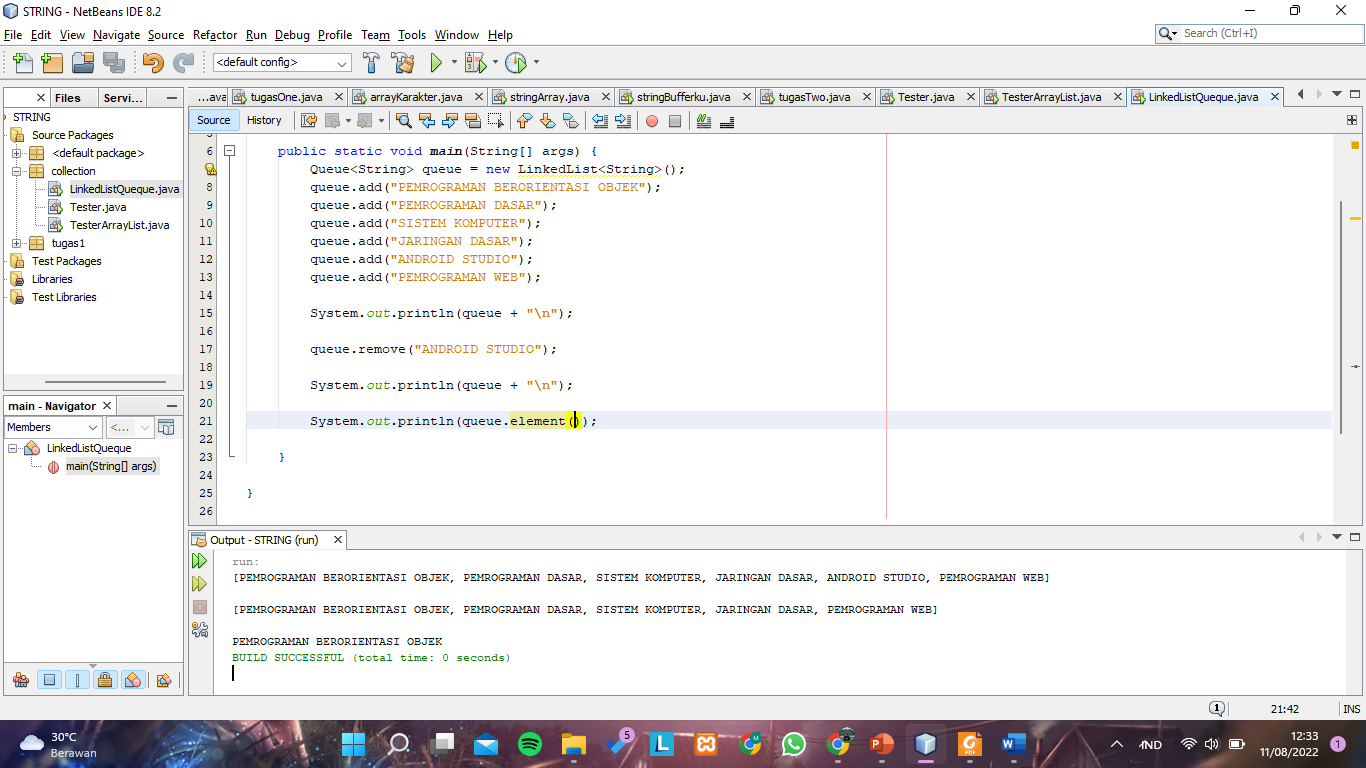
Pada output diatas setelah PBO yaitu pemrograman web, karena output dieksekusi   
 berdasarkan indeks ,dimanan pemrograman web berada pada indeks 1

Jika ingin menghapus data tertentu bisa menggunakan perintah remove disertai dengan  
 nomor indeks yg akan dihapus, pada array list juga dapat menambah data duplikat.

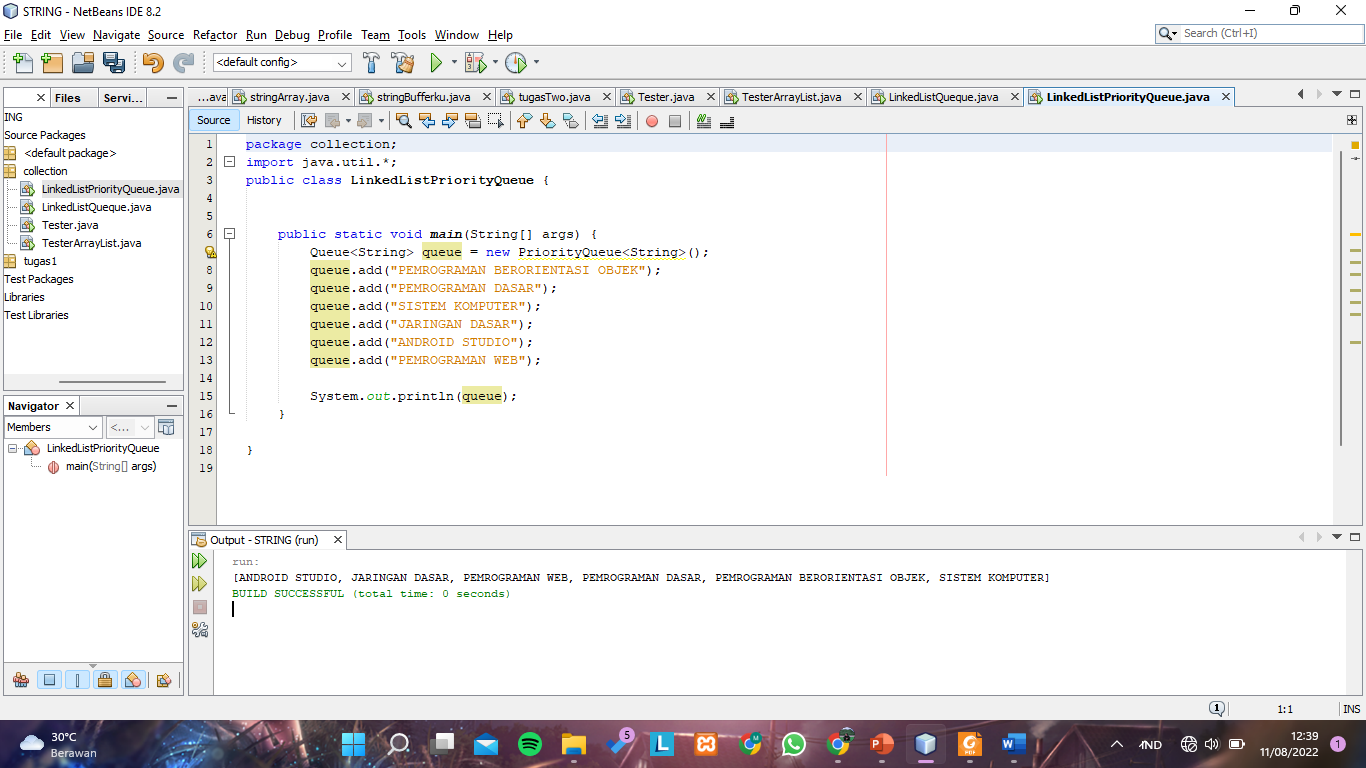
2.LinkedList Menggunakan Queque

Output berurutan sesuai inputan

Menghapus data menggunakan perintah remove(index)

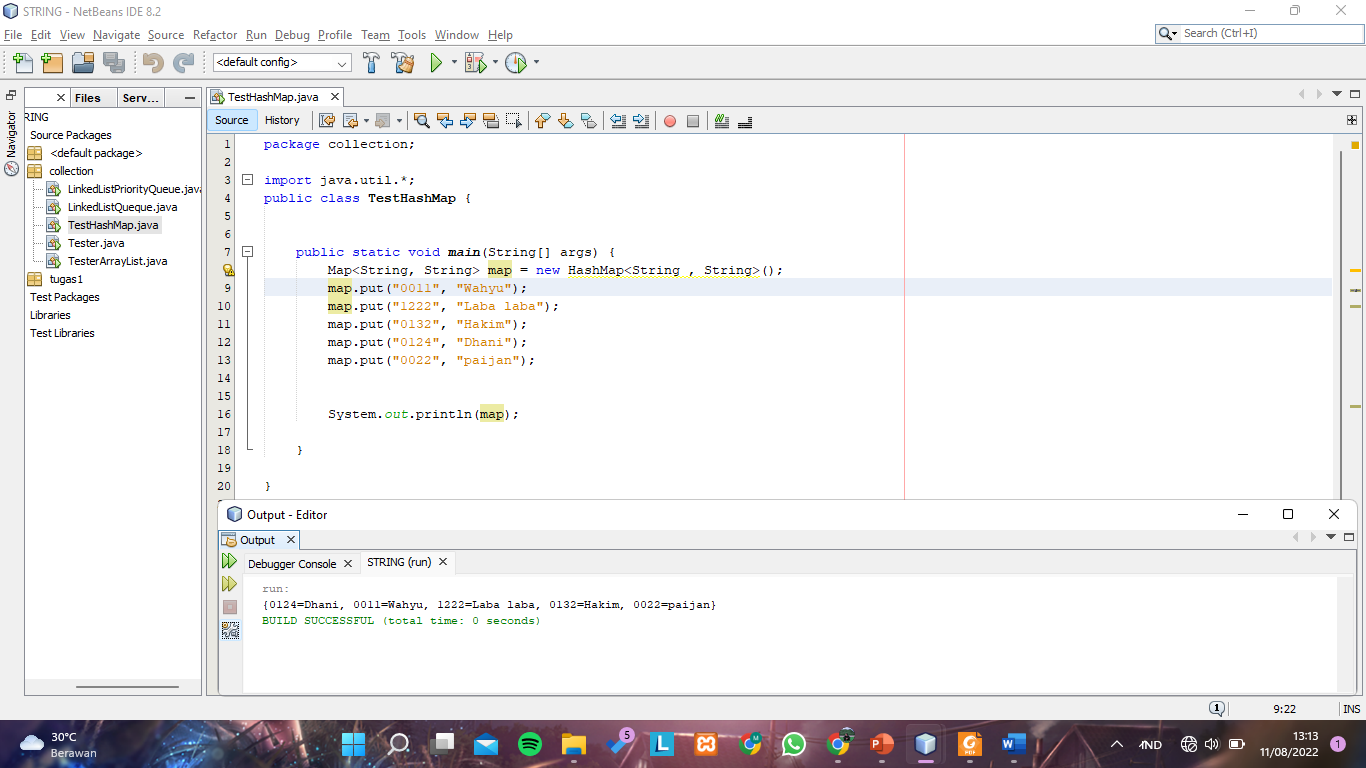


Mengambil data di posisi pertama menggunakan perintah .element()

3.PriorityQueue

Tergantung kebutuhannya dan bisa diatur,dimana terdapat kriteria yg dapat diatur di interface comparable

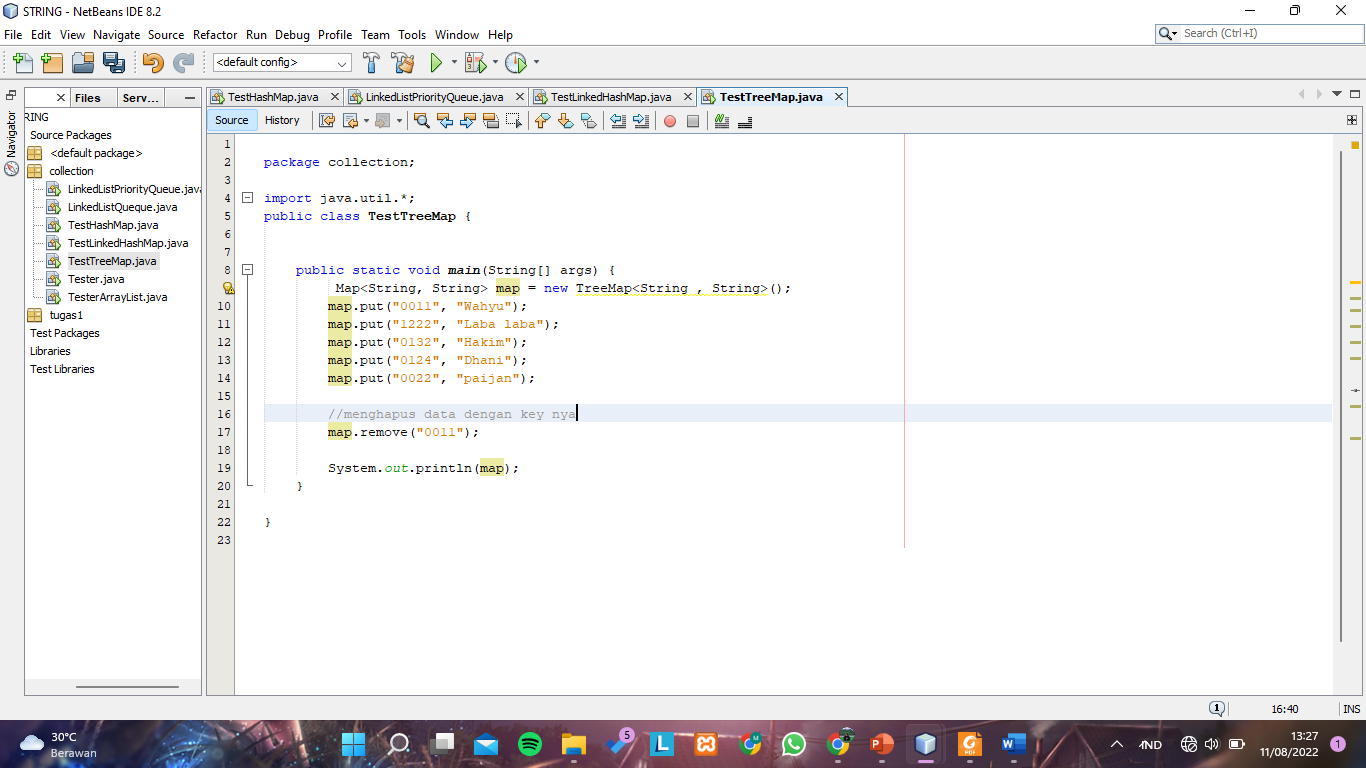
**2.Latihan MAP**

1.HashMap

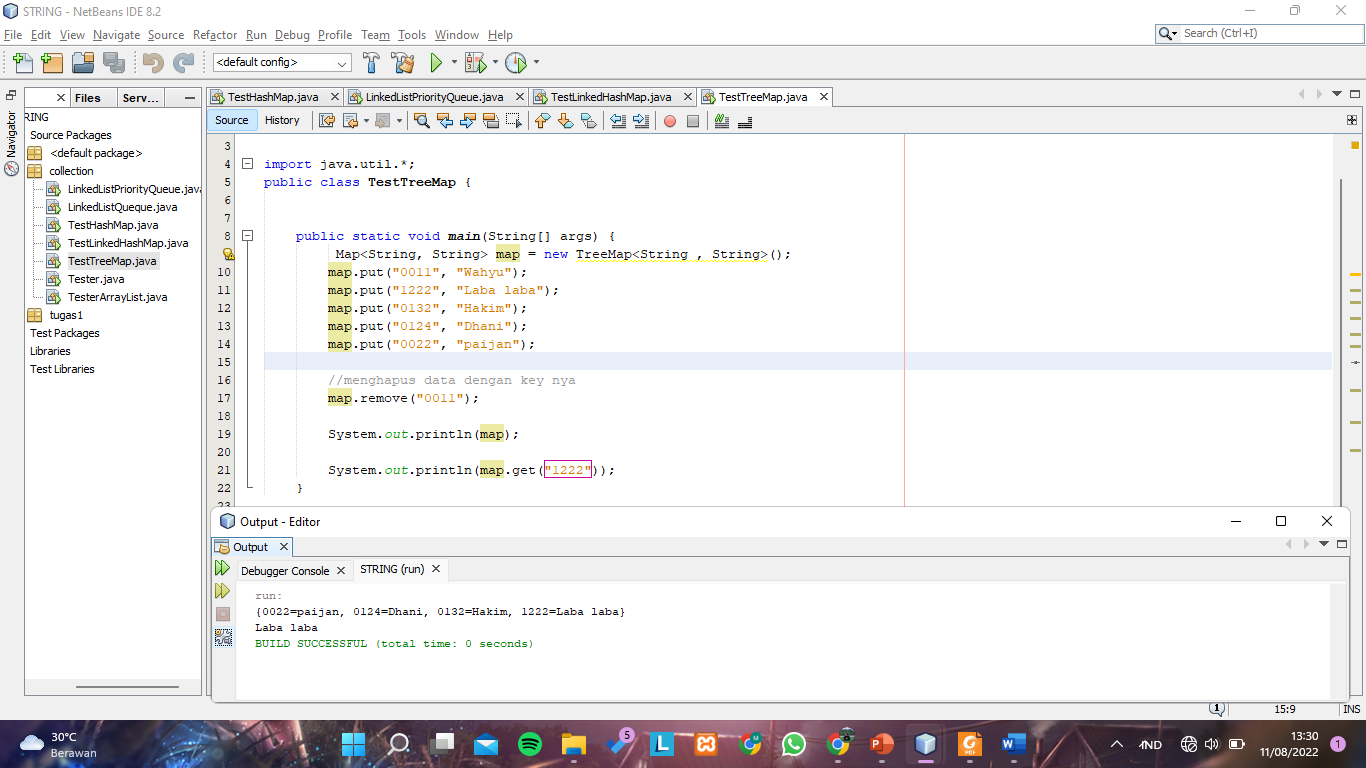
Output acak

2.Linked Hash Map

Data berurutan sesuai inputan

 3.TreeMap

Ascending nilai dari kecil ke besar dan menghapus data dengan key nya dan



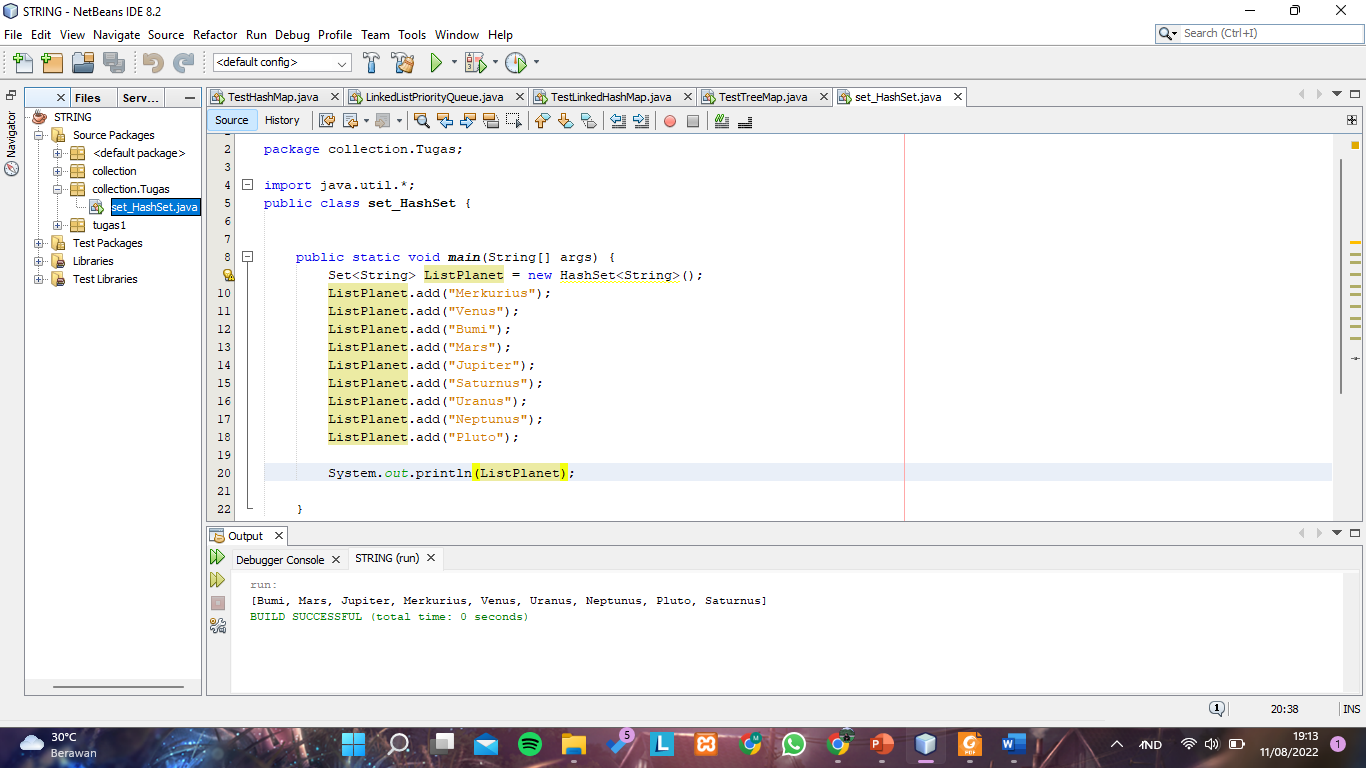
mengambil data dengan method get

TUGAS 1

**Collection Set**

Yaitu struktur data yg menampung elemen2 unik(tidak boleh sama),  
 Berguna untuk mengelola data yg tidak mungkin sama misal => (KTP,Email,no hp)

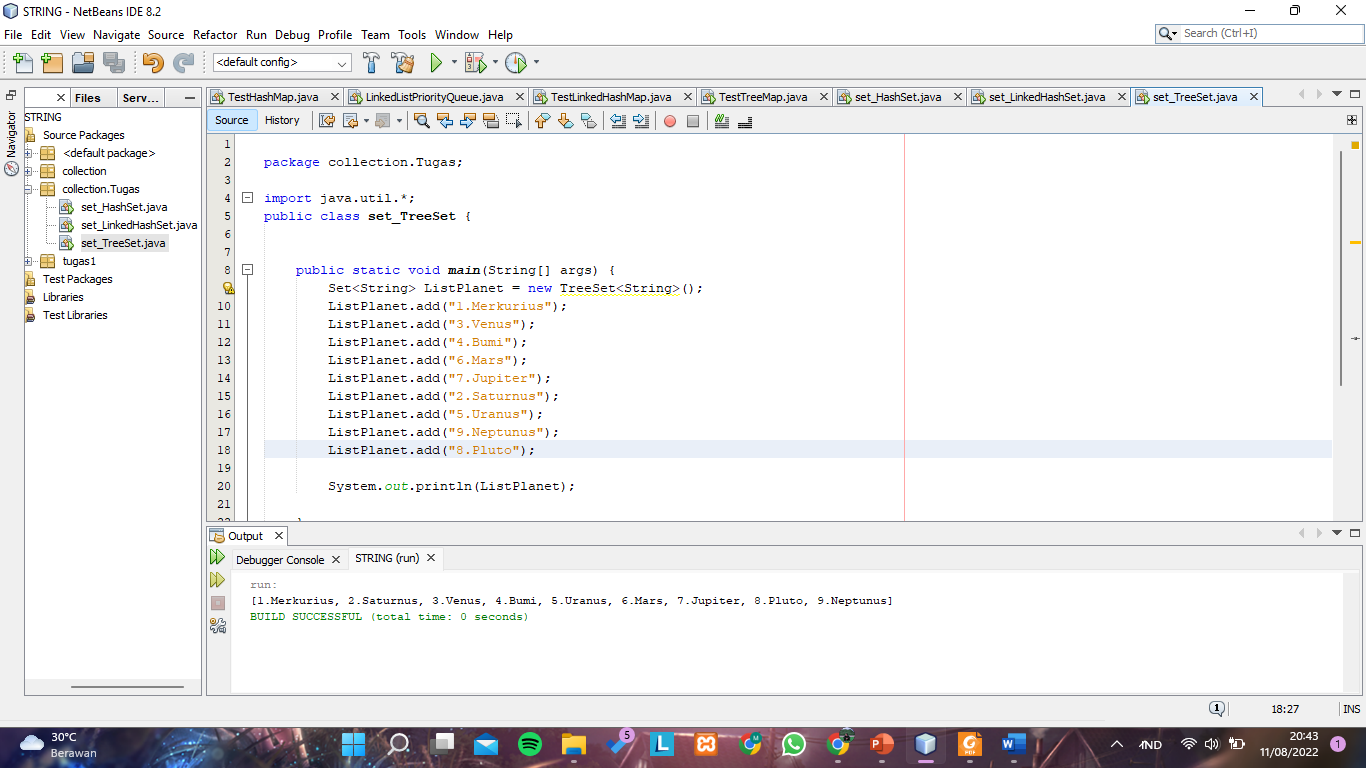
IMPLEMENTASI SET:

 1.HashSet

Implementasi HashSet menghasilkan output yg acak(tidak urut)sesuai inputan

 2.LinkedHashSet

Implementasi LinkedHashSet menghasilkan output urut sesuai inputan

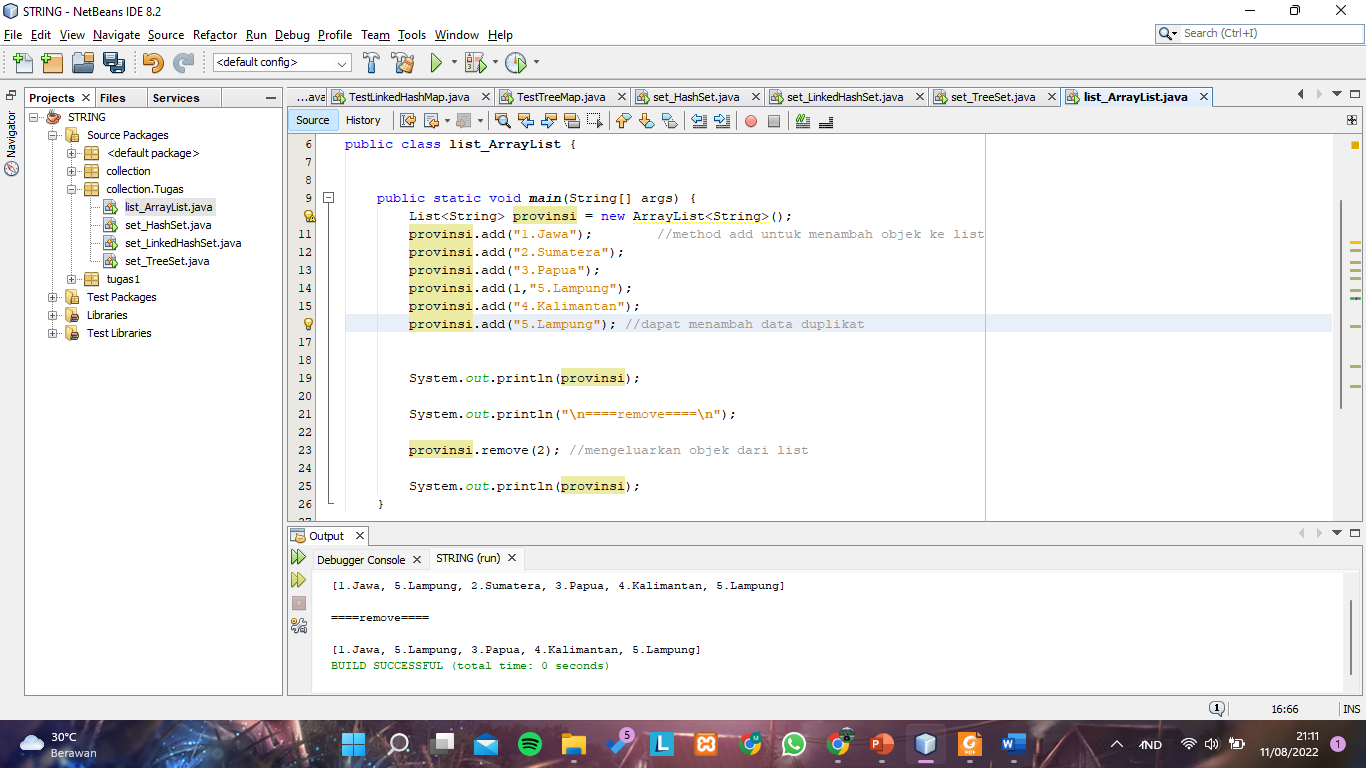
 3.TreeSet

Implementasi TreeSet menghasilkan nilai ascending/ nilai dari kecil ke besar

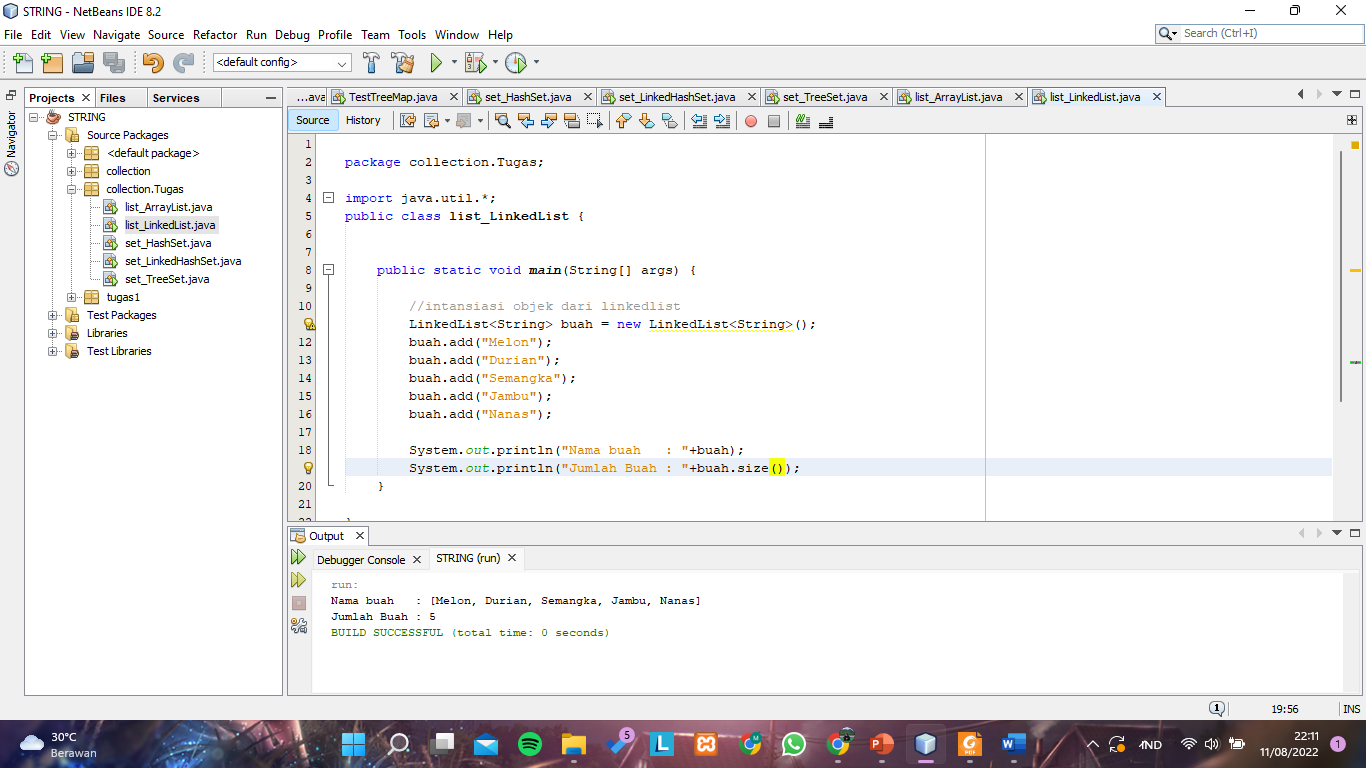
**Collection List**

\*List Menyimpan data secara sekuensial seperti array sehingga dapat diakses menggunakan sistem indexing.  
\*dapat menyimpan elemen duplikat,user dapat menentukan penyimpanan elemen  
\*berguna untuk mengelola data yg perlu memperhatikan posisi data   
contoh => (Presensi siswa, kursi bioskop)

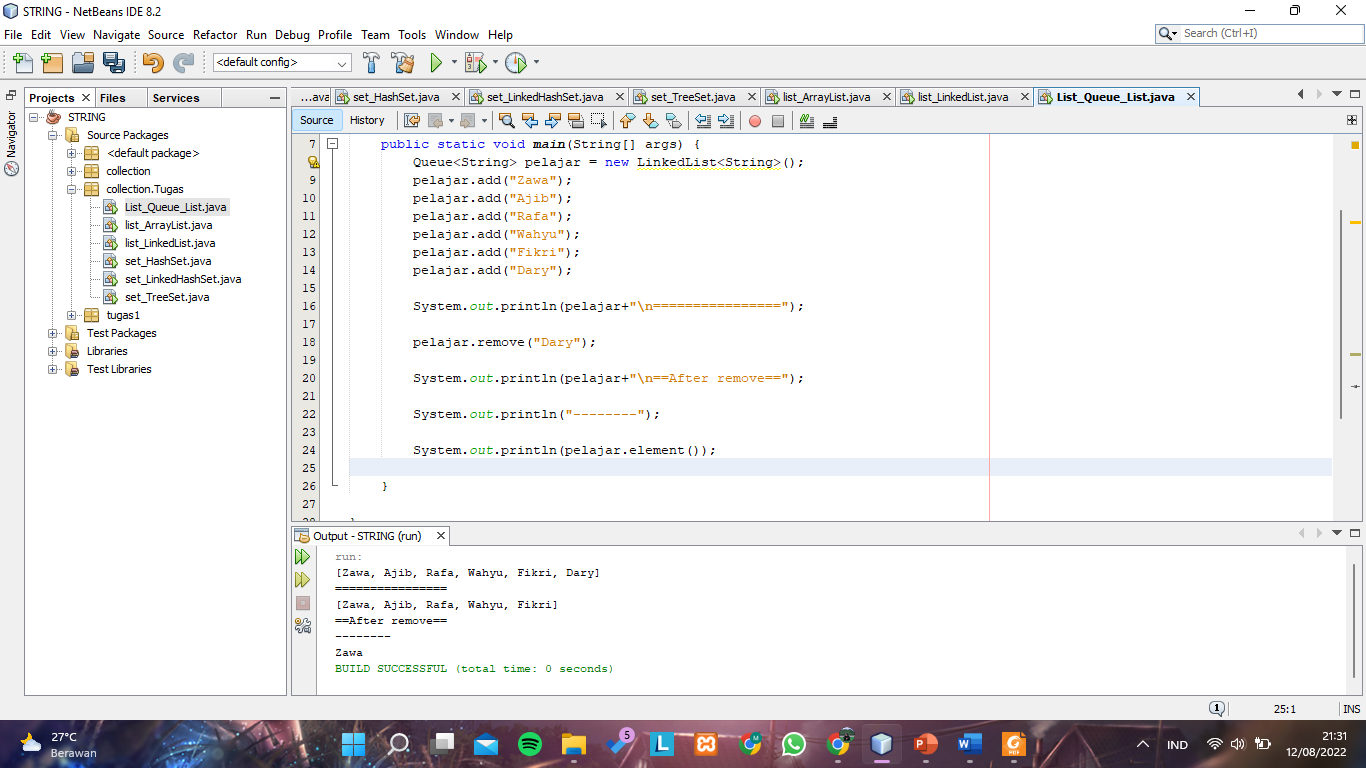
IMPLEMENTASI List:

 1.ArrayList

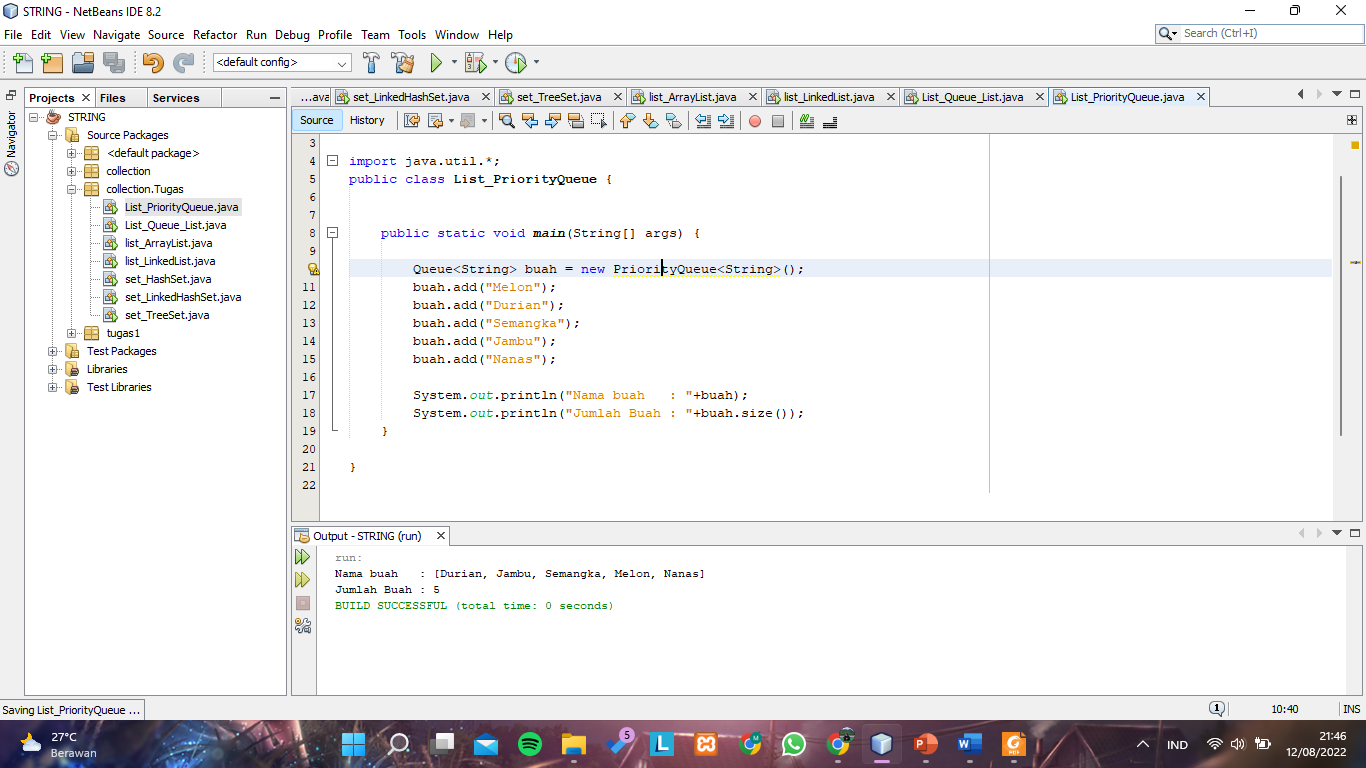
Output tersebut dipanggil berdasarkan index ,itulah sebabnya data LAMPUNG ditampilkan   
 setelah jawa,karena data lampung berada pada index 1.  
 Array list juga dapat menambah data duplikat dan jika ingin menghapus gunakan  
 perintah => remove(index data)

 2.LinkedList

Untuk menambahkan data pada Objek buah, kita menggunakan fungsi add(), lalu menampilkan nilainya pada statement System.out.println(), dan untuk melihat ukuran atau jumlah data yang terdapat pada Objek buah, kita bisa menggunakan fungsi size().

3.Queue dengan linked list

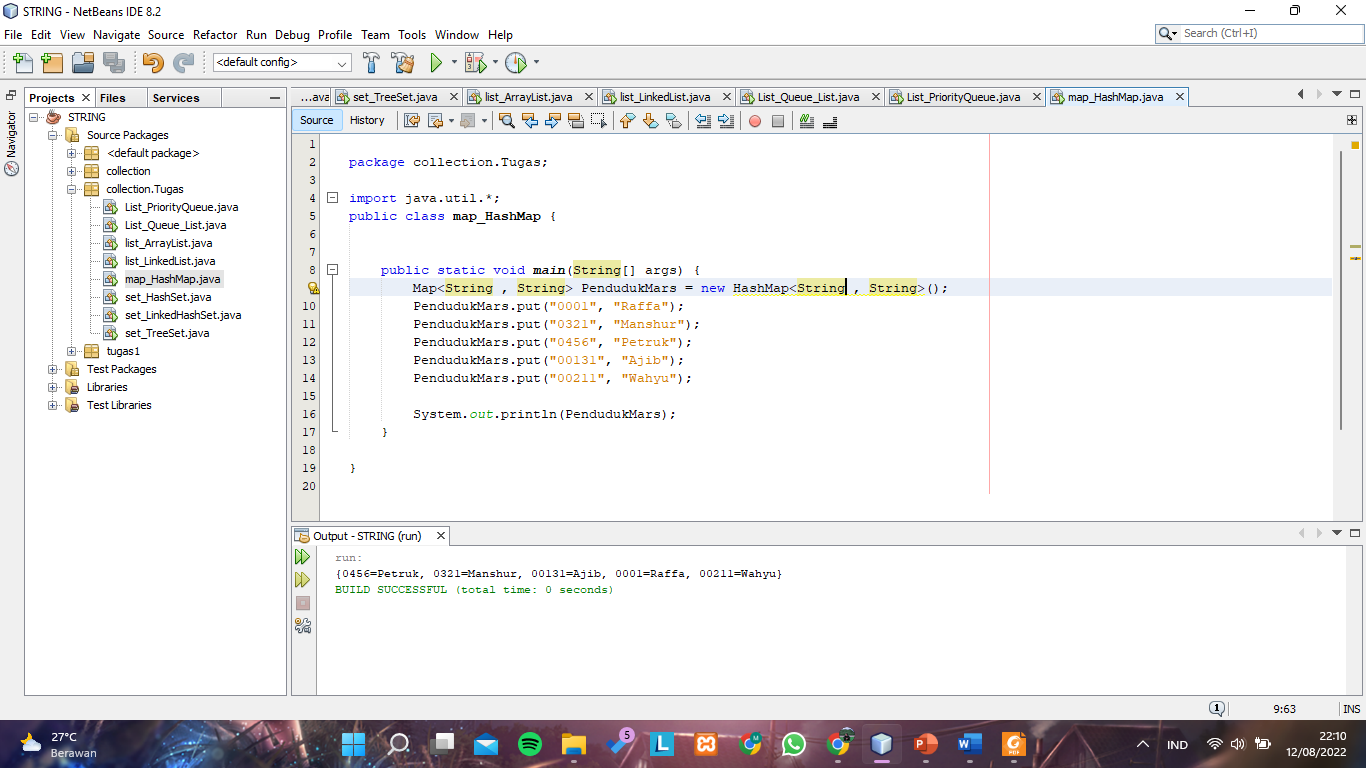
Output sesuai inputan,menghapus data perintah remove(),mengambil data awal menggunakan perintah element()

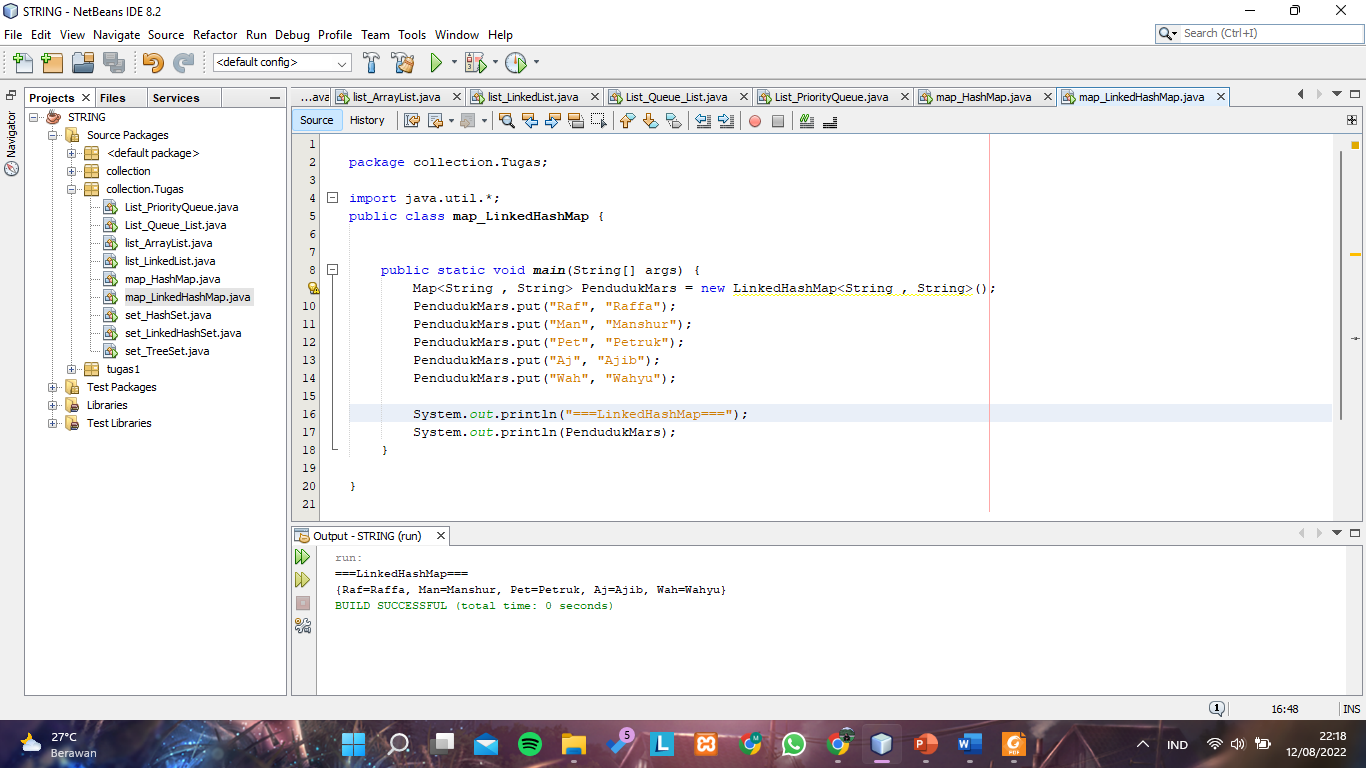
4.PriorityQueue

Tergantung kebutuhan + bisa diatur ,ada kriteria yg dapat diatur di interface comparable

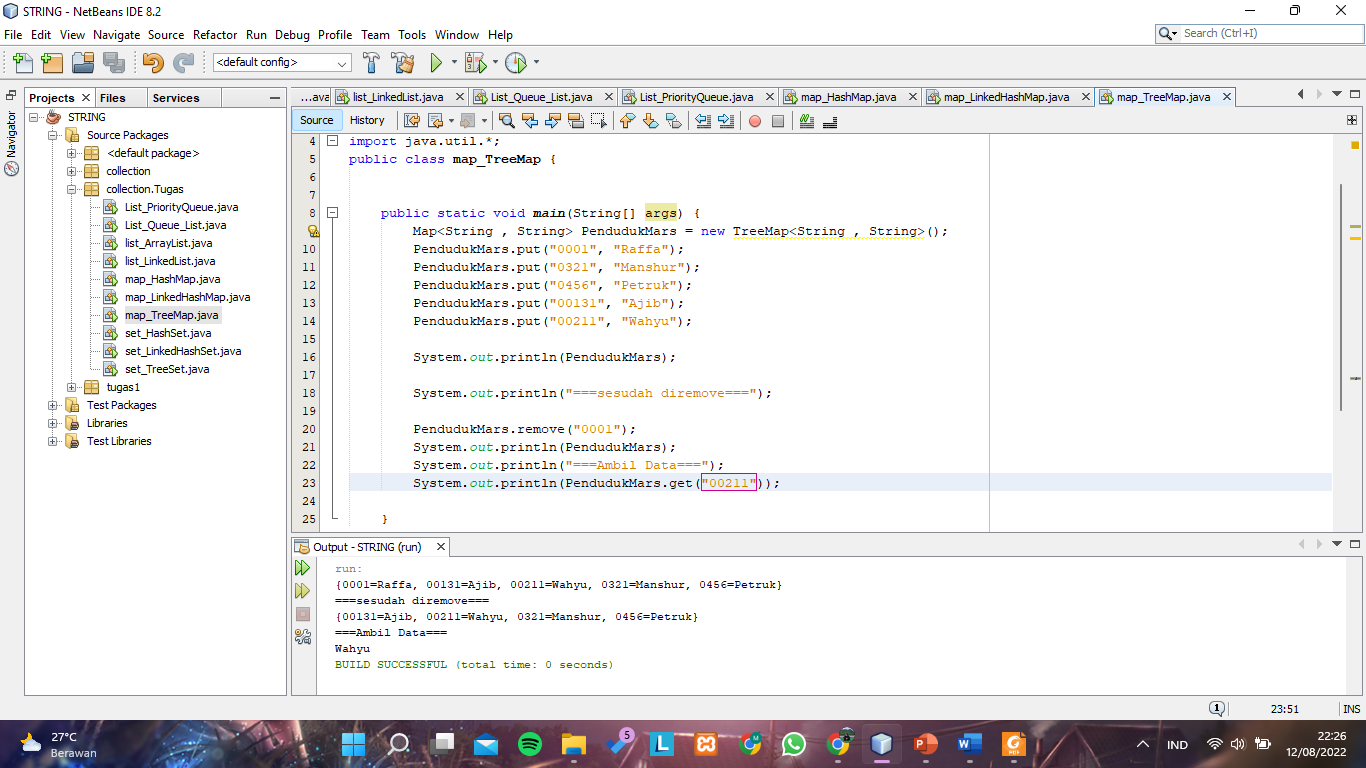
**Collection Map**

\*Merupakan container yg menyimpan elemen bersama dengan kuncinya(index),  
 map menggunakan sistem indexing yg disebut dengan Key  
 \*Kunci Harus Unik/tidak boleh sama misal => penduduk(Key : nomor KTP)

1.HashMap  
Output Menggunakan HashMap tidak memperhatikan posisi data(acak)

2.LinkedHashMap

Output Urut Sesuai Inputan

3.TreeMap

Output Ascending nilai dari kecil ke besar,remove kemudian masukkan key untuk menghapus data tertentu,method get untuk mengambil data dari key.

=========================================================================================()=========================================()=================================================--------------================================

---------------------------------------------(----------------------------)---------------------------------------------------------------------------------------------------------------------------------------------------------------------------