**LAPORAN**

**PRAKTIKUM STRUKTUR DATA**

**Pertemuan ke – 14**

****

Disusun Oleh :

Rendra Eka Herlambang

175410028

**LABORATORIUM TERPADU**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER AKAKOM YOGYAKARTA**

**2018**

**HASHING**

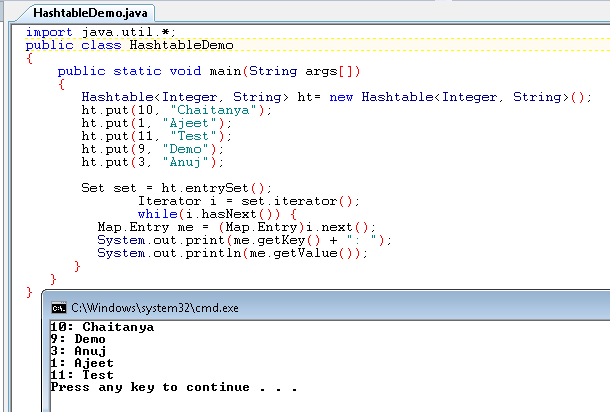
1. **TUJUAN**

-Mahasiswa dapat membuat program dengan hashing

- Mahasiswa mampu mengimplementasikan hashing

- Mahasiswa mampu mengimplemetasikan hashing dengan baik dan benar.

1. **PEMBAHASAN PRAKTIK**



Pembahasan

import java.util.\*;

public class HashtableDemo

{

//yaitu untuk membuat class HashtableDemo

public static void main(String args[])

{

Hashtable<Integer, String> ht= new Hashtable<Integer, String>();

//yaitu mendekrasikan data yang bertipe data String

ht.put(10, "Chaitanya");

ht.put(1, "Ajeet");

ht.put(11, "Test");

ht.put(9, "Demo");

ht.put(3, "Anuj");

//yaitu data yang akan di inputkan chaitanya,ajeet,Test,Demo dan anuj

Set set = ht.entrySet();

Iterator i = set.iterator();

while(i.hasNext()) {

///yaitu untuk melakukan perulangan banyak datanya

Map.Entry me = (Map.Entry)i.next();

System.out.print(me.getKey() + ": ");

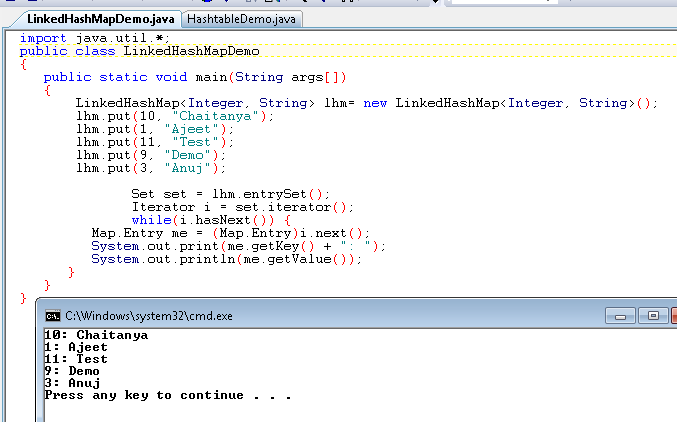
System.out.println(me.getValue());

}

}

}

//yaitu hasil keluaran data dari data 10 maka memenggal 11 jadi baru 9 3 1



Pembahasan

import java.util.\*;

public class LinkedHashMapDemo

{

//yaitu untuk membuat class LinkedHashMapDemo

public static void main(String args[])

{

LinkedHashMap<Integer, String> lhm= new LinkedHashMap<Integer, String>();

lhm.put(10, "Chaitanya");

lhm.put(1, "Ajeet");

lhm.put(11, "Test");

lhm.put(9, "Demo");

lhm.put(3, "Anuj");

//yaitu data yang akan di inputkan chaitanya,ajeet,Test,Demo dan anuj bedanya pada linkedHashMap

Set set = lhm.entrySet();

Iterator i = set.iterator();

while(i.hasNext()) {

///yaitu untuk melakukan perulangan banyak datanya

Map.Entry me = (Map.Entry)i.next();

System.out.print(me.getKey() + ": ");

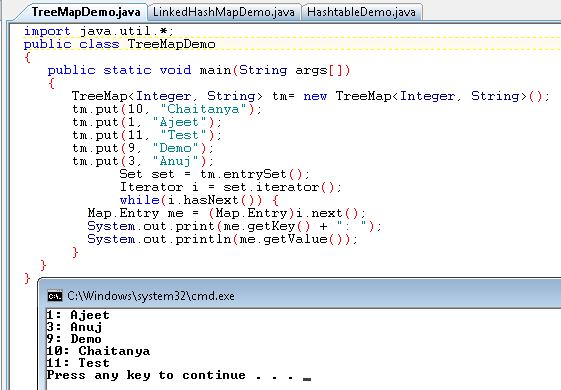
System.out.println(me.getValue());

}

}

}

//yaitu hasil keluaran datanya seperti data yang di inputkan tidak berubah dari nomer 10,1,11,9 dan 3



Pembahasan

import java.util.\*;

public class TreeMapDemo

{

//yaitu untuk membuat class TreeMapDemo

public static void main(String args[])

{

TreeMap<Integer, String> tm= new TreeMap<Integer, String>();

tm.put(10, "Chaitanya");

tm.put(1, "Ajeet");

tm.put(11, "Test");

tm.put(9, "Demo");

tm.put(3, "Anuj");

//yaitu data yang akan di inputkan chaitanya,ajeet,Test,Demo dan anuj bedanya pada TreeMap

Set set = tm.entrySet();

Iterator i = set.iterator();

while(i.hasNext()) {

///yaitu untuk melakukan perulangan banyak datanya

Map.Entry me = (Map.Entry)i.next();

System.out.print(me.getKey() + ": ");

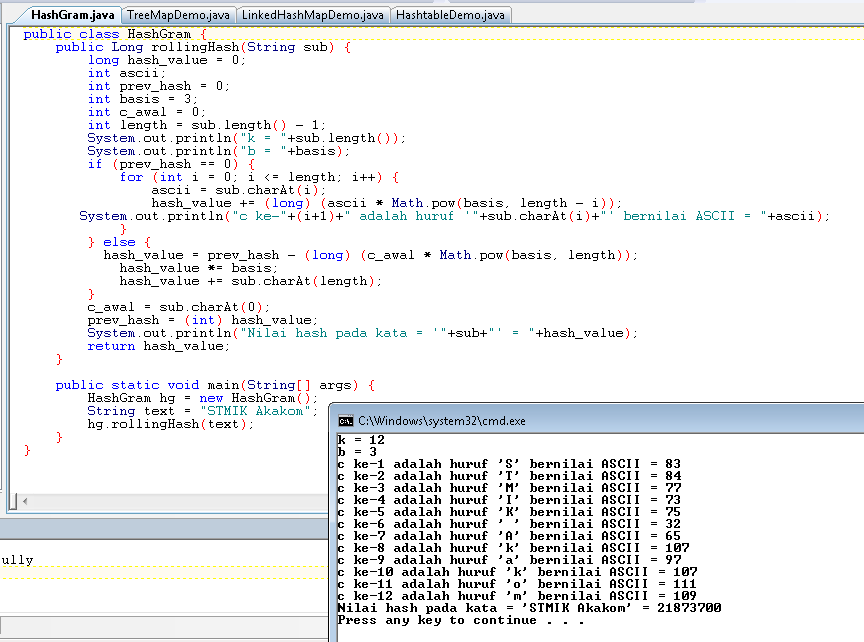
System.out.println(me.getValue());

}

}

}

//yaitu hasil keluaran atau output yaitu data dari kecil ke besar,maka 1,3,9,10 dan 11.



Pembahasan

public class HashGram {

//yaitu untuk membuat class HashGram

public Long rollingHash(String sub) {

long hash\_value = 0;

int ascii;

int prev\_hash = 0;

int basis = 3;

int c\_awal = 0;

int length = sub.length() - 1;

//pada method Long rollingHash maka nillai long hash dari 0, prev\_hash =0 basis 3 dan nilai awal=0

System.out.println("k = "+sub.length());

System.out.println("b = "+basis);

//yaitu hasil keluaran k maka di tambah banyaknya datanya ,dan b yaitu basis datanya.

if (prev\_hash == 0) {

for (int i = 0; i <= length; i++) {

ascii = sub.charAt(i);

hash\_value += (long) (ascii \* Math.pow(basis, length - i));

System.out.println("c ke-"+(i+1)+" adalah huruf '"+sub.charAt(i)+"' bernilai ASCII = "+ascii);

}

//yaitu untuk perulangan prev\_hash ==0 maka melakukan perulangan nilai awal i<= banyaknya data maka nambah 1 secara otomatis,maka hash\_value banyak jumlah data ascii maka datanya di kurang i,

} else {

hash\_value = prev\_hash - (long) (c\_awal \* Math.pow(basis, length));

hash\_value \*= basis;

hash\_value += sub.charAt(length);

}

//jika tidak maka hash\_value = prev\_hash – long dan nilai awal \*Math.pow, \*= basis maka di tambahkan banyaknya data.

c\_awal = sub.charAt(0);

prev\_hash = (int) hash\_value;

System.out.println("Nilai hash pada kata = '"+sub+"' = "+hash\_value);

return hash\_value;

}

//yaitu mendekrasikan nilai awal sampai akhir .

public static void main(String[] args) {

HashGram hg = new HashGram();

String text = "STMIK Akakom";

hg.rollingHash(text);

}

}

//maka yaitu gabungan nilai pada ascii pada kalimat stmik akakom.

KESIMPULAN

Hashingtransformasi aritmatik sebuah string dari karakter menjadi nilai yang merepresentasikan string aslinya. Menurut bahasanya, *hash* berarti memenggal dan kemudian menggabungkan. *Hashing* digunakan sebagai metode untuk menyimpan data dalam sebuah array agar penyimpanan data, pencarian data, penambahan data, dan penghapusan data dapat dilakukan dengan cepat. Dengan adanya hashing kita akan mengetahui nilai-nilai pada kode perkalimat.