A logo of a university

Description automatically generated

NAMA : Rakasyael Hizkia Kolondam

NIM : 2023105490

MATA KULIAH : TI0091 – PRAKTIKUM DATA TERSTRUKTUR

PRAKTIKUM KE - : 9

|  |  |  |
| --- | --- | --- |
| **No.** | **Kode** | **Penjelasan** |
| 1 | public HashTable() {  dataMap = new Node[size];  } | Metode ini adalah konstruktor dari kelas HashTable yang berfungsi untuk assign ukuran dari sebuah hash table. |
| 2 | public void printTable() {  for(int i = 0; i < dataMap.length; i++)  {  System.out.println(i + ":");  Node temp = dataMap[i];  while(temp != null)  {  System.out.println(" {" + temp.key + "= " + temp.value + "}");  temp = temp.next;  }  }  } | Metode ini berfungsi untuk menampilkan hash table dengan cara iterasi dari indeks terlebih dahulu lalu menampilkan seluruh key dan value dari hash table pada indeks tertentu |
| 3 | private int hash(String key) {  int hash = 0;  char[] keyChars = key.toCharArray();  for(int i = 0; i < keyChars.length; i++) {  int asciiValue = keyChars[i];  hash = (hash + asciiValue \* 23) % dataMap.length;  }  return hash;  } | Metode ini berfungsi untuk menentukan indeks dari key yang dimasukkan dengan cara mengubah string menjadi integer dengan ASCII table lalu menggunakan rumus tertentu untuk mengubahnya menjadi hashcode dari string tersebut. Lalu hash code akan di modulo berdasarkan size dari hash table untuk mendapatkan nilai indeksnya. |
| 4 | public void set(String key, int value) {  int index = hash(key);  Node newNode = new Node(key, value);  if(dataMap[index] == null)  {  dataMap[index] = newNode;  }  else  {  Node temp = dataMap[index];  if (temp.key == key) {  temp.value += value;  return;  }  while(temp.next != null)  {  temp = temp.next;  if(temp.key == key)  {  temp.value += value;  return;  }  }  temp.next = newNode;  }  } | Metode ini berfungsi untuk mencarikan dan memasukkan value ke dalam hash table menggunakan kunci yang sudah diubah menjadi indeks. Lalu menggunakan metode chainning jika terjadi collision yaitu jika indeks sudah terisi maka value dari indeks akan menunjukkan alamat dari value selanjutnya yang dimasukkan seperti linked list. |
| 5 | public int get(String *key*) {  // Modifikasi bagian ini, silahkan ubah return value.  int index = hash(*key*);  Node head = dataMap[index];    while (head != null){  if (head.key.equals(*key*)) {  return head.value;  }    head = head.next;  }    return -1; } | Metode ini berfungsi untuk mendapatkan value dari key yang dimasukkan dengan cara menentukan indeks terlebih dahulu lalu dengan metode search seperti di linked list akan mencari satu persatu secara sekuen. |