**LAPORAN PRAKTIKUM**

49

**ALGORITMA DAN PEMROGRAMAN LANJUT**

**Pertemuan Ke – 11**



**DISUSUN OLEH :**

**HELDA LUDYA SAFITRI**

**175410186**

**TEKNIK INFORMATIKA**

**STMIK AKAKOM**

**YOGYAKARTA**

**2018**

**PERTEMUAN KE-11**

**COLLECTION**

## TUJUAN

1. Mahasiswa dapat menjelaskan dan menggunakan penyimpanan objek menggunakan Collection.
2. Mahasiswa dapat menggunakan metode pada kelas Collections untuk pengurutan, pencarian, mengacak, dan membalik elemen data.
3. **DASAR TEORI**

Collection adalah suatu objek yang bisa digunakan untuk menyimpan sekumpulan objek.objek yang ada dalam collection disebut elemen.collection menyimpan elemen yang bertipe objet, sehingga berbagai tipe object bisa disimpan dalam collection.

1. **PEMBAHASAN PRAKTIK**

**Praktik 1**

import java.util.\*;

public class Main {

public static void main(String[] args){

Scanner reader = new Scanner(System.in);

String input;

String[] array = new String[3];

int i = 0;

do{

System.out.println("Enter a word");

input = reader.nextLine();

array[i] = input;

i++;

}while(i<3);

System.out.println("The array now has the following :");

for(int j=0; j<array.length;j++){

System.out.print(""+array[j]);

}

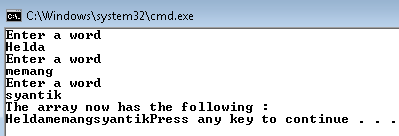
}

}

Script di atas merupakan syntax program penggunaan array dimana array bernama array dengan kapasitas data sebanyak 3. Data tersebut diinputkan oleh user melalui scanner dengan menggunakan perulagan do-while.

Pada do akan dikerjakan penginputan data oleh user kemudian pada while jika nilai i bernilai kurang dari 3 maka akan dijalankan perulangan for dimana akan menampilkan data yang terdapat pada array tadi.

Output :



**Praktik 2**

import java.util.\*;

public class cobaarray{

public static void main(String[]args){

Scanner reader = new Scanner(System.in);

String input;

ArrayList list = new ArrayList();

int i=0;

do {

System.out.println("Enter a word");

input = reader.nextLine();

list.add(input);

i++;

}while(i<3);

System.out.println("The array now has the following : "+list);

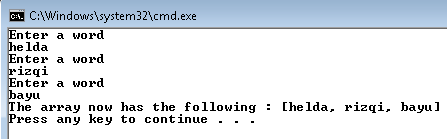
}

}

Script di atas merupakan syntax program menggunakan method ArrayList. Jika pada array praktikan harus menambahkan kapasitas data pada array maka pada ArrayList ini hal tersebut tidak perlu dilakukan.

Pada script di atas terdsapat penggunaan loop do-while, dimana pada do dilakukan penginputan data oleh user melalui scanner. Kemudian pada while jika i bernilai kurang dari 3 maka akan ditampilkan data dari array tersebut. Bisa dikatakan kapasitas data pada aray tersebut ditentukan dari kondisi yang ada pada while.

Output :



**Praktik 3**

import java.util.\*;

public class cobaarray2{

public static void main(String[] args){

Scanner reader = new Scanner(System.in);

String input;

ArrayList list = new ArrayList();

int i = 0;

do {

System.out.println("Enter a word");

input = reader.nextLine();

if(!input.equals("end"))

list.add(input);

i++;

}while(!input.equals("end"));

System.out.println("The array now has the following : "+list);

Collections.sort(list);

System.out.println("The array after sort : "+list);

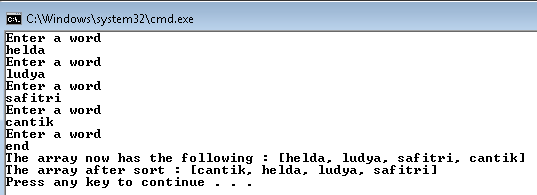
}

}

Script di atas merupakan syntax program yang hampir sama dengan praktik 2 tetapi praktikan memodifikasi program tersebut. Modifikasi terletak pada penggunaan if serta kondisi yang terdapat pada while.

Terlihat pada do terdapat penginputan data oleh user, tetapi user hanya dapat menginputkan data jika inputannya tidak bernilai “end”. Kemudian pada while terdapat kondisi yang mengatakan jika input bernilai “end” maka perintah yang terdapat pada do tidak lagi dikerjakan dan akan menampilkan dat array yang telah diinputkan tadi. Kemudian program juga akan menampilkan data array yang telah diurutkan menggunakan method sort.

Output :



**Praktik 4**

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class MyList {

public static void main(String[] arg) {

//create a list using the

List<String> list = Arrays.asList("Lars", "Simon");

List<String> anotherList = new ArrayList<>();

anotherList.add("Lars");

anotherList.add("Simon");

for (String string : anotherList) {

System.out.println(string);

}

List<Integer> listInt = Arrays.asList(3,2,1,4,5,6,6);

for(Integer integer : listInt) {

System.out.println(integer);

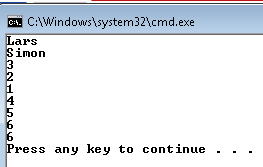
}

}

}

Script di atas merupakan syntax program penggunaan method ArrayList, Arrays, dan List. Terdapat 2 list, list yang petama berisikan data “lars” dan “simon”. List kedua berisikan ArrayList. Data yag terdapat pada ArrayList adalah 3, 3, 2, 1, 4, 5, 6, 6. Dat atersbut akan ditampilkan menggunakan scanner melalui perulangan for.

Output :



**Praktik 5**

import java.util.List;

import java.util.Arrays;

import java.util.Collections;

import java.util.ArrayList;

public class BinarySearchTest {

public static void main (String[] args) {

String[] colors = {"red", "white", "blue", "black", "yellow", "purple", "tan", "pink"};

List<String> list = new ArrayList<String>(Arrays.asList(colors) );

Collections.sort(list);

System.out.printf("Sorted ArrayList : %s\n", list);

printSearchResults (list, colors[3]);

printSearchResults (list, colors[0]);

printSearchResults (list, colors[7]);

printSearchResults (list, "aqua");

printSearchResults (list, "gray");

printSearchResults (list, "teal");

}

private static void printSearchResults (

List<String>list,String key) {

int result = 0;

System.out.printf("\nSearching for : %s\n", key);

result = Collections.binarySearch(list, key);

if(result>=0)

System.out.printf("Not Found (%d)\n", result);

}

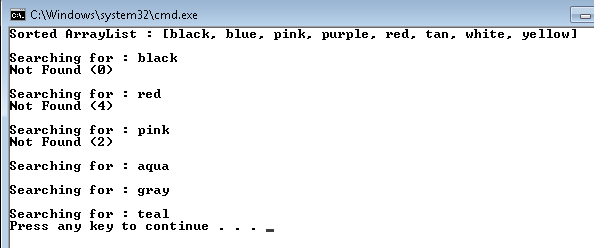
}

Script di atas merupakan syntax progam penggunaan method List, Arrays, Collections, dan ArrayList. Terdapat array colors dengan data seperti pada gambar. Kemudian terdapat List yang berisikan ArrayList dimana data dalam ArrayList adalah data yang sama dengan array colors. Kemudian data tersebut diurutkan menggunakna method sort.

Di bawahnya terdapat pencarian nilai berdasarkan index. Program akan mencari data dari array colors pada index ke-3,0, dan 7. Serta dilakukan pencarian berdasarkan nama data, terlihat terdapat pencarian “Aqua”, “Grey”, dan “teal”.

Kemudian data yang dicari tesebut akan ditampilkan pada output. Jika data ditemukan maka akan ditampilkan juga ditemukan pada index data keberapa. Jika tidak ditemukan maka akan ditampilkan “Not found” .

Output :



1. **LATIHAN**

Latihan 1

import java.util.List;

import java.util.Arrays;

import java.util.Collections;

import java.util.ArrayList;

public class latihan {

public static void main(String[]args){

StringBuffer sb=new StringBuffer();

String[]hari={"senin","selasa","rabu","kamis","jumat","sabtu","minggu";

List<String> list= new ArrayList<String>(Arrays.asList(hari));

System.out.println("Data Pada List:"+list);

System.out.println("==================================");

Collections.reverse(list);

System.out.println("Data Setelah Dibalik:"+list);

System.out.println("==================================");

Collections.shuffle (list);

System.out.println ("Data Setelah Diacak"+list);

System.out.println("==================================");

Collections.sort(list);

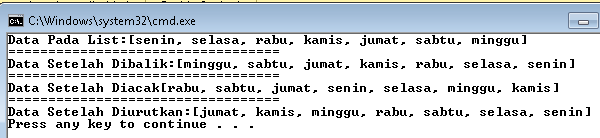
System.out.println("Data Setelah Diurutkan:"+list); }}

Script di atas merupakan program pengunaan objek list untuk daat menampilkan isi List, membalikkan data List, mengacak data List, dan mengurutkan data List.

Terlihat pada gambar penggunaan list.add yang bertujuan untuk menambahkan data dalam list, yang di dalam kurung merupakan data yang ditambahkan. Untuk menampilkan dat adari list, praktikan cukup menggunakan S.O.P saja.

Kemudian isi dari data List tersebut dibalik dengan menggunakan method reverse lalu List akan ditampilkan. Setelah itu praktikan mengacak isi data dari List menggunakna method shuffle yang kemudian ditampilkan menggunakna S.O.P. kemudian praktikan mengurutkan data pada List menggunakan .sort yang kemudian ditampilkan juga melalui S.O.P .

Output :



Latihan 2

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

public class MyList {

public static void main(String[] arg) {

//create a list using the

List<String> list = Arrays.asList("Lars", "Simon");

List<String> anotherList = new ArrayList<>();

anotherList.remove("Lars");

anotherList.remove("Simon");

for (String string : anotherList) {

System.out.println(string);

}

List<Integer> listInt = Arrays.asList(3,2,1,4,5,6,6);

for(Integer integer : listInt) {

System.out.println(integer);

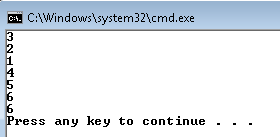
}

}

}

Pada latihan 2 sama dengan praktik 4, perbedaannya adalah pada statement Lars dan Simon ada selection remove yang berfungsi untuk menghapus atau menghilangkan, sehingga pada output Lars dan Simon tidak ditampilkan lagi.

Output :



1. **TUGAS**

import java.util.\*;

public class tugasalgoku{

public static void main(String[]arg){

Scanner reader = new Scanner (System.in);

String input,input2;

int index;

ArrayList list =new ArrayList();

int i = 0;

do {

System.out.print("Enter a word :");

input = reader.nextLine();

if(!input.equals("end"))

list.add(input);

i++;

}while(i<5);

System.out.println("Array Sebelum ditambahkan :"+list);

System.out.print("Masukkan kata tambahan pada index ke-3 :");

input2 = reader.nextLine();

list.add(3,input2);

System.out.println("Array setelah ditambahkan : "+list);

list.clear();

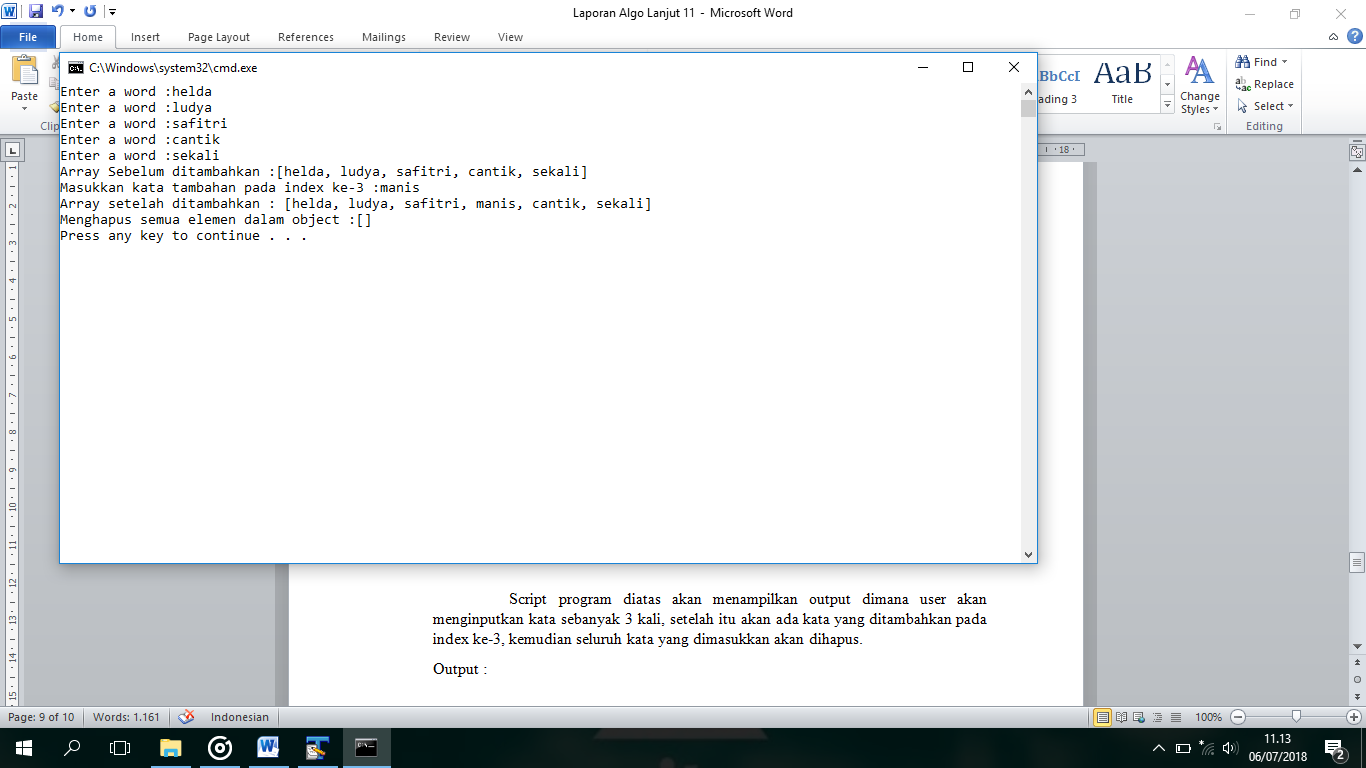
System.out.println("Menghapus semua elemen dalam object :"+list);

}

}

Script program diatas akan menampilkan output dimana user akan menginputkan kata sebanyak 3 kali, setelah itu akan ada kata yang ditambahkan pada index ke-3, kemudian seluruh kata yang dimasukkan akan dihapus.

Output :



1. **KESIMPULAN**

Berdasarkan pembahasan praktik diatas dapat disimpulkan bahwa penggunaan method pada collections mempermudah pembuat progam dalam membuat program yang efisien, lebih ringkas, serta lebih mudah dimengerti.

1. **LISTING**

Terlampir.