Nama : Monica Tifani Zahara

NRP : 171 111 077 / TI

Praktikum pemrograman dasar 2

1. Bubble sort

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package modul3;

/**
 *
 * @author monica
 */
import java.util.Scanner;

class angka {

    public angka next;
    public int node;
}

public class bubblesort {

    static angka head;
    static int size = 0;

    public static void print() {
        angka current = head;
        while (current != null) {
            System.out.print(current.node + " ");
            current = current.next;
        }
        System.out.println("");
    }

    public static void insert(int new_node) {
        angka nilai = new angka();
        nilai.node = new_node;
        if (head != null) {
            angka datanya = head;
            while (datanya.next != null) {
                datanya = datanya.next;
            }
            datanya.next = nilai;
        } else {
            head = nilai;
        }
        size++;
    }
```

```java

    public static void removeLast() {
        if (head != null) {
            if (head.next == null) {
                head = null;
            } else {
                angka datanya = head;
                while (datanya.next.next != null) {
                    datanya = datanya.next;
                }
                datanya.next = null;
            }
        }
        size--;
    }

    public static void sort() {
        if (size > 1) {
            boolean wasChanged;
            do {
                angka current = head;
                angka previous = null;
                angka next = head.next;
                wasChanged = false;
                while (next != null) {
                    if (current.node > next.node) {
                        wasChanged = true;
                        if (previous != null) {
                            angka sig = next.next;
                            previous.next = next;
                            next.next = current;
                            current.next = sig;
                        } else {
                            angka sig = next.next;
                            head = next;
                            next.next = current;
                            current.next = sig;
                        }
                        previous = next;
                        next = current.next;
                    } else {
                        previous = current;
                        current = next;
                        next = next.next;
                    }
                }
            } while (wasChanged);
        }
    }

    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int bdata , isidata;
        System.out.print("Banyak data yang ingin diinput : ");
        bdata = in.nextInt();
        for (int i = 1; i <= bdata; i++) {
            System.out.print("Masukkan data ke "+i + " : ");
            isidata = in.nextInt();
            insert(isidata);
        }
        System.out.println("================ Data ================ ");
        print();
        System.out.println("======== After Bubble Sorting ========");
        sort();
        print();
        System.out.println("======= Menghapus Data Terakhir ======");
        removeLast();
        print();


    }
}
```

Running program

```
Output - PraktikumProdas2 (run)
run:
Banyak data yang ingin diinput : 9
Masukkan data ke 1 : 7
Masukkan data ke 2 : 12
Masukkan data ke 3 : 23
Masukkan data ke 4 : 5
Masukkan data ke 5 : 16
Masukkan data ke 6 : 3
Masukkan data ke 7 : 1
Masukkan data ke 8 : 8
Masukkan data ke 9 : 4
================ Data ================
7 12 23 5 16 3 1 8 4
======== After Bubble Sorting ========
1 3 4 5 7 8 12 16 23
======= Menghapus Data Terakhir ======
1 3 4 5 7 8 12 16
BUILD SUCCESSFUL (total time: 1 minute 20 seconds)
```

2. Quick Sort

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pertemuan3;
import java.util.Scanner;
public class LinkedList {

    LinkedListNode head;
    LinkedListNode tail;

    LinkedList() {
        this.head = null;
        this.tail = null;
    }


    /* First set a Node named current into head
    * while current is not null, print current.data, set current = current.next
    * print end of line
     */
    void print() {
        LinkedListNode current = this.head;
        while (current != null) {
            System.out.print(current.data + " ");
            current = current.next;
        }
        System.out.println("");
    }

    /* if LinkedList is empty, set new_node as head and tail
    * if LinkedList is not empty, set tail.next into new_node, set
      new_node.prev into tail, and make new_node a new tail
     */
    void push(LinkedListNode new_node) {
        if (this.head == null) {
            this.head = new_node;
            this.tail = new_node;
        } else {
            if (find_node_by_data(new_node.data) == null) {
                this.tail.set_next(new_node);
                this.tail = new_node;
            }

        }
    }

    /* if linked list is empty, set new_node as head and tail
    * if new_node < head, make it a new head
    * if new_node > tail, make it a new tail
    * otherwise traverse to the current position, and put new_node there
     */
```

```java
    void insert(LinkedListNode new_node) {
        if (this.head == null) {
            this.head = new_node;
            this.tail = new_node;
        } else if (new_node.data <= this.head.data) {
            this.head.set_prev(new_node);
            this.head = new_node;

        } else if (new_node.data >= this.tail.data) {
            this.tail.set_next(new_node);
            this.tail = new_node;

        } else {
            LinkedListNode position = head;
            while (position.data < new_node.data) {
                position = position.next;

            }
            LinkedListNode previous_position = position.prev;
            new_node.set_prev(previous_position);
            new_node.set_next(position);

        }
    }

    LinkedListNode find_node_by_data(int data) {
        LinkedListNode current = this.head;
        while (current != null) {
            if (current.data == data) {
                return current;
            }
            current = current.next;
        }
        return null;
    }
    LinkedListNode lastNode(LinkedListNode node)
    {
        while (node.next != null ) {
            node = node.next;
        }
//        System.out.println("last : " + node.data);
        return node;
    }

    void delete(LinkedListNode deleted) {
        if (deleted != null && this.head != null) {
            if (this.head == this.tail && deleted == this.head) {
                this.head = null;
                this.tail = null;
            } else if (deleted == this.head) {
                LinkedListNode new_head = this.head.next;
                this.head.set_next(null);
                new_head.set_prev(null);
                this.head = new_head;
            } else if (deleted == this.tail) {
                LinkedListNode new_tail = this.tail.prev;
                this.tail.set_prev(null);
                new_tail.set_next(null);
                this.tail = new_tail;
            } else {
                LinkedListNode deleted_prev = deleted.prev;
                LinkedListNode deleted_next = deleted.next;
                deleted.set_prev(null);
                deleted.set_next(null);
                deleted_prev.set_next(deleted_next);
            }
        }

    }
```

```
123.     //mengecek nilai pertama dan terakhir
124.     public void quickSort(LinkedListNode node)
125.     {
126.         LinkedListNode last = lastNode(node);
127.
128.         _quickSort(node, last);
129.     }
130.
131.     void _quickSort(LinkedListNode l, LinkedListNode h)
132.     {
133.         if(h != null && l != h && l != h.next)
134.         {
135.             LinkedListNode temp = partition(l, h);
136.             _quickSort(l, temp.prev);
137.             _quickSort(temp.next, h);
138.         }
139.     }
140.
141.     LinkedListNode partition(LinkedListNode l, LinkedListNode h)
142.     {
143.         int x = h.data;
144.         LinkedListNode i = l.prev;
145.
146.         for (LinkedListNode j=l; j != h; j=j.next) {
147.             if (j.data <= x) {
148.                 i = (i == null) ? l : i.next;
149.                 int temp = i.data;
150.                 i.data = j.data;
151.                 j.data = temp;
152.             }
153.         }
154.         i = (i==null) ? l : i.next;
155.         int temp = i.data;
156.         i.data = h.data;
157.         h.data = temp;
158.         return i;
159.     }
160.
161.
162.     public static void main(String[] args) {
163.        Scanner sc = new Scanner(System.in);
164.         int data;
165.         LinkedList a = new LinkedList();
166.         int bdata;
167.
168.         System.out.print("Banyak Data yang ingin di input : ");
169.         bdata = sc.nextInt();
170.         sc.nextLine();
171.
172.         for(int i=1; i<=bdata; i++)
173.         {
174.             System.out.print("Masukkan Data ke "+i + " : ");
175.             data = sc.nextInt();
176.             a.push(new LinkedListNode(data));
177.         }
178.         System.out.println("================ Data ================ ");
179.         a.print();
180.         System.out.println("======== After Quick Sorting ========");
181.         a.quickSort(a.head);
182.         a.print();
183.     }
184. }
185.
186.
```

Running program

```
run:
Banyak Data yang ingin di input : 10
Masukkan Data ke 1 : 7
Masukkan Data ke 2 : 5
Masukkan Data ke 3 : 4
Masukkan Data ke 4 : 8
Masukkan Data ke 5 : 12
Masukkan Data ke 6 : 3
Masukkan Data ke 7 : 1
Masukkan Data ke 8 : 2
Masukkan Data ke 9 : 70
Masukkan Data ke 10 : 69
================ Data ================
7 5 4 8 12 3 1 2 70 69
======== After Quick Sorting ========
1 2 3 4 5 7 8 12 69 70
BUILD SUCCESSFUL (total time: 33 seconds)
```