Praktikum 13

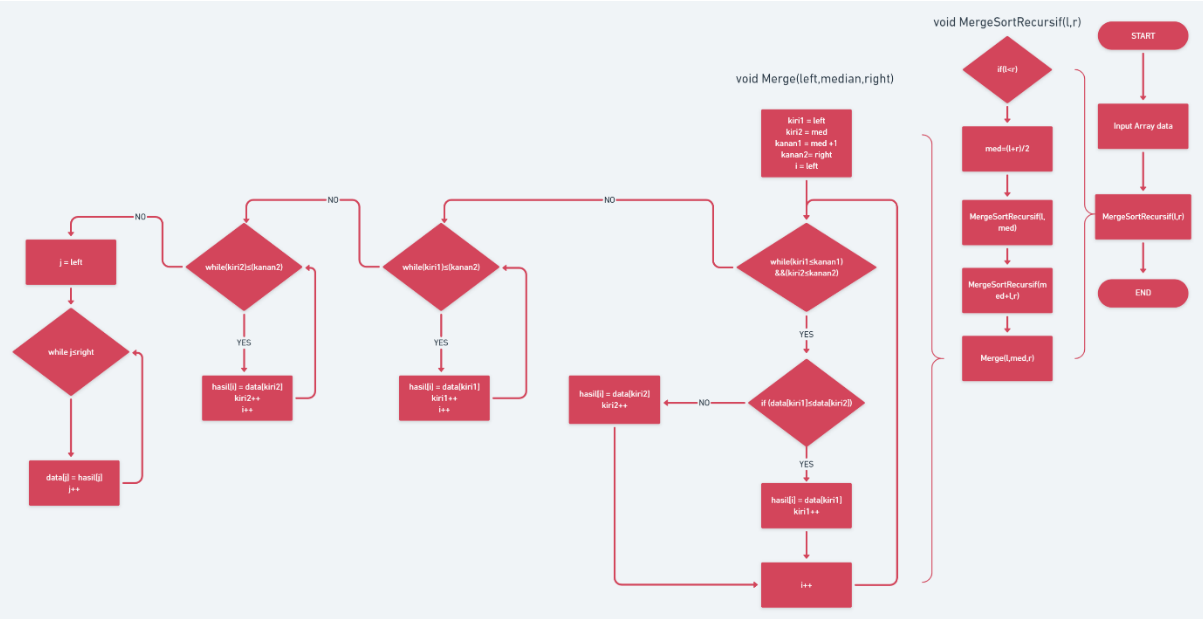
Agiftsany Azhar

D3-Sistem Informasi

LATIHAN MERGER SORT

Soal 1

Tuliskanlah algoritma MERGER SORT dengan menggunakan flow chart



Soal 2

Tuliskanlah algoritma MERGER SORT dengan menggunakan pseudo code

```
void MergeSortRekursif(l,r)
1. jika (l<r) maka kerjakan baris 2 – 5
2. med = (l+r)/2
3. MergeSortRekursif(l,med)
4. MergeSortRekursif(med+l,r)
5. Merge(l,med,r)

void Merge(left, median, right)
1. kiri1 ← left
2. kanan1 ← median
3. Kiri2 ← median+1
4. Kanan2 ← right
5. i ← left
6. selama (kiri1<=kanan1) && (kiri2<=kanan2) kerjakan baris 7 – 13
```

```
7.  if (Data[kiri1] <= Data[kiri2]) kerjakan baris 8 – 9
8.  hasil[i] = Data[kiri1]
9.  kiri1++
10. jika tidak kerjakan baris 11 – 12
11. hasil[i] = Data[kiri2]
12. kiri2++
13. i++
14. selama (kiri1 <= kanan1) kerjakan baris 15 – 17
15. hasil[i] = Data[kiri1]
16. kiri1++
17. i++

18. selama (kiri2 <= kanan2) kerjakan baris 19 – 21
19. hasil[i] = Data[kiri2]
20. i++
21. kiri2++

22. j ← left
23. selama (j <= rigt) kerjakan baris 24 – 25
24. Data[j] = hasil[j]
25. j++
```

Soal 3

Implementasikan algoritma MERGER SORT dalam bentuk function di dalam class array

```
package tugas13_152011513020;


public class Tugas13_152011513020 {


    public static void main(String[] args) {


//      BubbleSort(5,10);
//      SelectionSort(5,10);
//      InsertionSort(5,10);
        MergeSort(200000,1000000);
    }


    public static void BubbleSort(int data, int nilai){
        //bubelsrot
        Array bS;
        bS=new Array(data); //banyak data

        bS.input(nilai); //batas nilai data

        bS.bubbleSort();

//      bS.print();
    }

    public static void SelectionSort(int data, int nilai){
        //seleksensrot
        Array sS;

        sS=new Array(data); //banyak data

        sS.input(nilai); //batas nilai data

        sS.selectionSort();

//      sS.print();
    }

    public static void InsertionSort(int data, int nilai){
        //insertsrot
        Array iS;

        iS=new Array(data); //banyak data

        iS.input(nilai); //batas nilai data
        iS.insertionSort();

//      iS.print();
    }

    public static void MergeSort(int data, int nilai){
        //insertsrot
        Array mS;

        mS=new Array(data); //banyak data

        mS.input(nilai); //batas nilai data

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

        long mulai = System.currentTimeMillis();

        mS.mergerSort(mS.getArray(),1,mS.getArray().length);

        long kelar = System.currentTimeMillis()-mulai;

        System.out.println("Waktu :"+kelar+" ms");
    }
}

package tugas13_152011513020;

import java.util.Random;

public class Array {
    int[] a;
```

```
int maxsize;
Random rand = new Random();

Array(int size){
    this.a = new int[size];
    this.maxsize = size;
}

int[] getArray(){
    return this.a;
}

void print(){
    for(int i=0; i<this.maxsize; i++){
        System.out.print(this.a[i]+" ");
    }
    System.out.println("");
}

void swap(int i, int j){
    int z = this.a[i];
          this.a[i] = this.a[j];
    this.a[j] = z;
}

void input(int n){
    for (int i=0;i<this.maxsize;i++){
        this.a[i]=rand.nextInt(n)+1;
    }
}
```

```
void bubbleSort(){
    long swap = 0;
    for (int i=2;i<this.maxsize;i++){
        for (int j=0;j<=this.maxsize-i;j++){
            if(this.a[j]>this.a[j+1]){
                swap(j+1,j);
                swap++;
            }
        }
    }

    System.out.println(swap);
}

void selectionSort(){
    long swap = 0;
    for(int i=this.maxsize-1; i>=1; i--){
        int max=i;
        for(int j=i-1; j>=0;j--){
            if(this.a[j]>this.a[max]){
                max=j;
            }
        }
        swap(i,max);
        swap++;
    }
    System.out.println(swap);
}

void insertionSort(){
    long swap=0;
    for (int i=0; i<this.maxsize; i++){
        int j=i-1;
        int tmp = this.a[i];
        while(j>=0 && this.a[j]>tmp){
            swap(j,j+1);
            swap++;
            j--;
        }
        this.a[j+1]=tmp;
    }
    System.out.println(swap);
}

void mergerSort(int[] a, int start, int end) {
    int n = end - start + 1;
    if (n <= 1)
            return;

    int middle = start + (n / 2) - 1;
    mergerSort(a, start, middle);
    mergerSort(a, middle + 1, end);
    merge(a, start, middle, end);
}

void merge(int[] a,int start, int middle, int end){
    int[] b = new int[end - start + 1];

    int i = start;
    int j = middle + 1;

    for (int k = 0; k < b.length; k++) {
            if (j > end) {
                b[k] = a[i - 1];
                i++;
            }
            else if (i > middle) {
                b[k] = a[j - 1];
                j++;
            }
            else if (a[i - 1] < a[j - 1]) {
                b[k] = a[i - 1];
                i++;
            }
            else {
                b[k] = a[j - 1];
                j++;
            }
        }

        for (int k = 0; k < b.length; k++) {
            a[start - 1 + k] = b[k];
        }
    }
}
```

Soal 4

Generate 200.000 data dan hitunglah waktu yang diperlukan untuk mengurutkan data tersebut dengan menggunakan program yang dihasilkan di soal 3

```
run:
Waktu :78 ms
BUILD SUCCESSFUL (total time: 0 seconds)
```