**LAPORAN**

**PRAKTIKUM ANALISIS ALGORITMA**

**TUGAS 04**



**Disusun Oleh:**

Dimas Satria Prakoso (140810170007)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**SUMEDANG**

**2019**

**LAPORAN**

**PRAKTIKUM ANALISIS ALGORITMA**

**TUGAS 04**

**Studi Kasus**

1. Merge Sort

Program:

/\*

Nama Program    : Merge Sort

Nama            : Dimas Satria Prakoso

NPM             : 140810170007

Tanggal Buat    : 20 Maret 2019

Praktikum : Analisis Algoritma

Tugas : QUIZ 1

.cpp tested with CodeBlocks 16.01

\*/

#include<iostream>

#include<conio.h>

#include<stdlib.h>

#define MAX\_SIZE 10

using namespace std;

void merge\_sort(int, int);

void merge\_array(int, int, int, int);

int arr\_sort[MAX\_SIZE];

int main() {

int i;

cout << "\nEnter " << MAX\_SIZE << " Elements for Sorting : " << endl;

for (i = 0; i < MAX\_SIZE; i++)

cin >> arr\_sort[i];

cout << "\nYour Array Data :";

for (i = 0; i < MAX\_SIZE; i++) {

cout << "\t" << arr\_sort[i];

}

merge\_sort(0, MAX\_SIZE - 1);

cout << "\n\nSorted Data :";

for (i = 0; i < MAX\_SIZE; i++) {

cout << "\t" << arr\_sort[i];

}

getch();

}

void merge\_sort(int i, int j) {

int m;

if (i < j) {

m = (i + j) / 2;

merge\_sort(i, m);

merge\_sort(m + 1, j);

merge\_array(i, m, m + 1, j);

}

}

void merge\_array(int a, int b, int c, int d) {

int t[50];

int i = a, j = c, k = 0;

while (i <= b && j <= d) {

if (arr\_sort[i] < arr\_sort[j])

t[k++] = arr\_sort[i++];

else

t[k++] = arr\_sort[j++];

}

while (i <= b)

t[k++] = arr\_sort[i++];

while (j <= d)

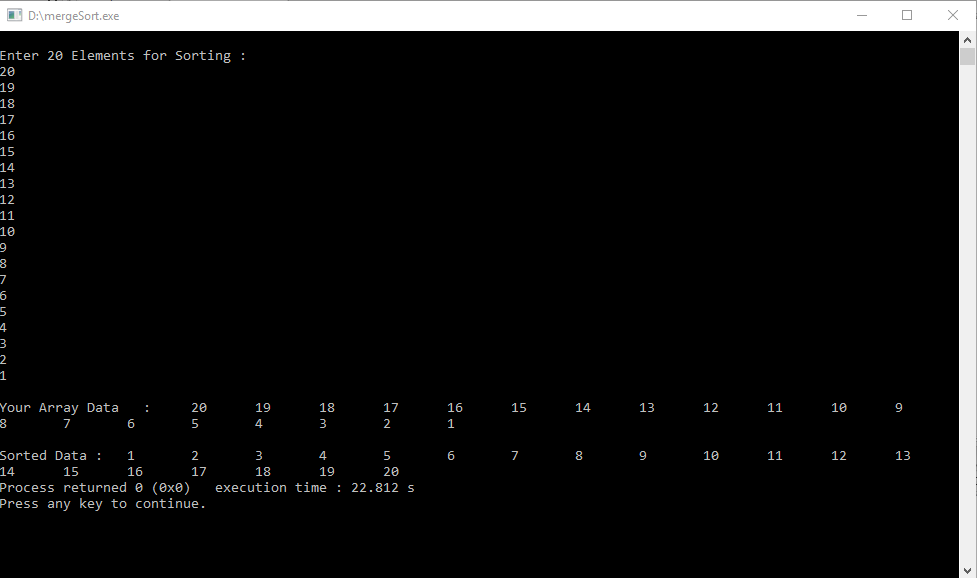
t[k++] = arr\_sort[j++];

for (i = a, j = 0; i <= d; i++, j++)

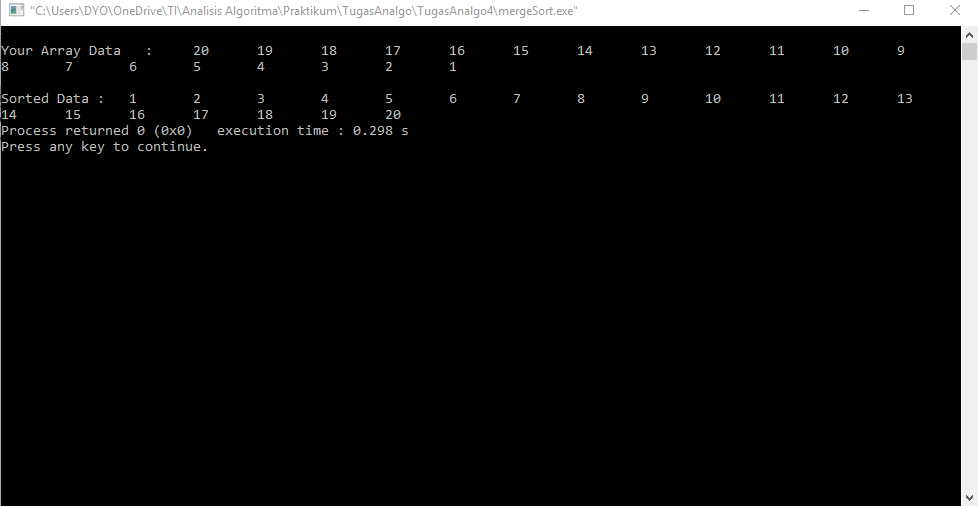
arr\_sort[i] = t[j];

}

Running time (N=20) dengan user input: 22,812



Running time tanpa user input (n=20) = 0.298s



1. Selection sort

Tentukan T(n)!

Menentukan T(n):

Oleh karena itu:

*Karena*

Program

#include <stdio.h>

swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void selectionSort(int arr[], int n)

{

int i, j, min\_idx;

for (i = 0; i < n - 1; i++)

{

min\_idx = i;

for (j = i + 1; j < n; j++)

if (arr[j] < arr[min\_idx])

min\_idx = j;

swap(&arr[min\_idx], &arr[i]);

}

}

void printArray(int arr[], int size)

{

int i;

for (i = 0; i < size; i++)

printf("%d ", arr[i]);

printf("\n");

}

int main()

{

int arr[] = {25, 12, 22, 64, 11};

int n = sizeof(arr) / sizeof(arr[0]);

printf("Inputed Array: ");

printArray(arr, n);

selectionSort(arr, n);

printf("Sorted Array: ");

printArray(arr, n);

return 0;

}

1. Insertion Sort

Tentukan T(n)!

Menentukan T(n) :

T(n) =

Program:

#include <math.h>

#include <stdio.h>

void insertionSort(int arr[], int n)

{

    int i, key, j;

    for (i = 1; i < n; i++) {

        key = arr[i];

        j = i - 1;

        while (j >= 0 && arr[j] > key) {

            arr[j + 1] = arr[j];

            j = j - 1;

        }

        arr[j + 1] = key;

    }

}

void printArray(int arr[], int n)

{

    int i;

    for (i = 0; i < n; i++)

        printf("%d ", arr[i]);

    printf("\n");

}

int main()

{

    int arr[] = { 4, 7, 5, 16, 9, 23};

    int n = sizeof(arr) / sizeof(arr[0]);

printf("Inputed Array: ");

printArray(arr, n);

    insertionSort(arr, n);

printf("Sorted Array: ");

    printArray(arr, n);

    return 0;

}

1. Bubble sort

Tentukan T(n)!

Menentukan T(n):

Program:

#include <stdio.h>

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void bubbleSort(int arr[], int n)

{

int i, j;

for (i = 0; i < n - 1; i++)

for (j = 0; j < n - i - 1; j++)

if (arr[j] > arr[j + 1])

swap(&arr[j], &arr[j + 1]);

}

void printArray(int arr[], int size)

{

int i;

for (i = 0; i < size; i++)

printf("%d ", arr[i]);

printf("\n");

}

int main()

{

int arr[] = {9, 16, 3, 73, 53,24, 31, 41};

int n = sizeof(arr) / sizeof(arr[0]);

printf("Inputed Array: ");

printArray(arr, n);

bubbleSort(arr, n);

printf("Sorted Array: ");

printArray(arr, n);

return 0;

}