TUGAS 2

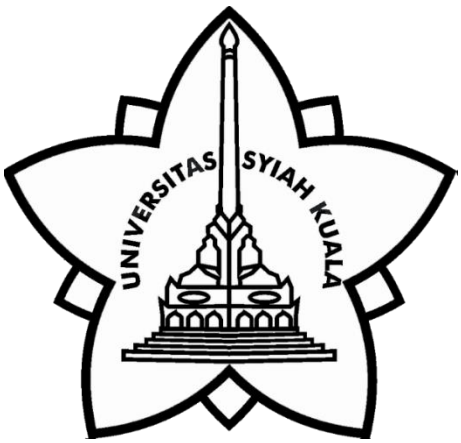
Disusun untuk memenuhi

Tugas Mata Kuliah Struktur Data dan Algoritma

Oleh:

Nurul Uzratun Nashriyyah

2208107010030



JURUSAN INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS SYIAH KUALA

2024

Kode:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#define MAX 1000000

void bubbleSort(int arr[], int n)
{
    int i, j, temp;
    int swapped;
    for (i = 0; i < n - 1; i++)
    {
        swapped = 0;
        for (j = 0; j < n - i - 1; j++)
        {
            if (arr[j] > arr[j + 1])
            {
                temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
                swapped = 1;
            }
        }
        // Jika tidak ada elemen yang ditukar dalam loop, maka array sudah
diurutkan
        if (swapped == 0)
            break;
    }
}

void selectionSort(int arr[], int n)
{
    int i, j, min_idx, temp;
    for (i = 0; i < n - 1; i++)
    {               // Ulangi (size - 1) kali
        min_idx = i; // Setel elemen yang belum diurutkan pertama sebagai minimum
        for (j = i + 1; j < n; j++)
        { // Untuk setiap elemen yang belum diurutkan
            if (arr[j] < arr[min_idx])
            {               // Jika elemen < minimum saat ini
                min_idx = j; // Setel elemen sebagai minimum baru
            }
        }
        // Tukar minimum dengan posisi yang belum diurutkan pertama
        temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
}
```

```c
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {                // Untuk setiap elemen yang belum diurutkan
        key = arr[i]; // 'Ekstrak' elemen X
        j = i - 1;
        // Jika elemen saat ini > X, pindahkan elemen yang sudah diurutkan ke
kanan sebanyak 1
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        // Sisipkan X di sini
        arr[j + 1] = key;
    }
}
void saveToFile(char *filename, int arr[], int n, char *title)
{
    FILE *fp;
    fp = fopen(filename, "a");
    if (fp == NULL)
    {
        printf("Gagal membuka file.\n");
        exit(1);
    }
    fprintf(fp, "%s\n", title);
    for (int i = 0; i < n; i++)
    {
        fprintf(fp, "%d\n", arr[i]);
    }
    fprintf(fp, "\n");
    fclose(fp);
}

int main()
{
    int arr[MAX], temp[MAX], n = MAX, i;
    clock_t start, end;
    double cpu_time_used;
    char filename[20] = "numbers.txt";
    char title[50];

    printf("| %-15s | %-16s | %-20s |\n", "Jenis Algoritma", "Jumlah Bilangan",
"Waktu Eksekusi (ms)");
    printf("|-----------------|------------------|----------------------|\n");

    // Menghasilkan bilangan acak
    for (i = 0; i < n; i++)
```

```
    {
        arr[i] = rand();
    }

    // Menyimpan bilangan yang belum terurut
    saveToFile(filename, arr, n, "Unsorted Numbers");

    for (int n = 100000; n <= MAX; n += 100000)
    {
        // Copy array untuk digunakan dalam setiap algoritma pengurutan
        memcpy(temp, arr, n * sizeof(int));

        // Mengurutkan bilangan dengan bubble sort dan mencatat waktu eksekusi
        start = clock();
        bubbleSort(temp, n);
        end = clock();
        cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
        printf("| %-15s | %15dk | %20.2f |\n", "Bubble Sort", n / 1000,
cpu_time_used * 1000);

        // Menyimpan bilangan yang sudah terurut dengan bubble sort
        sprintf(title, "Sorted Numbers (Bubble Sort) - %d Numbers", n);
        saveToFile(filename, temp, n, title);
    }
    for (int n = 100000; n <= MAX; n += 100000)
    {
        // Copy array untuk digunakan dalam setiap algoritma pengurutan
        memcpy(temp, arr, n * sizeof(int));

        // Mengurutkan bilangan dengan selection sort dan mencatat waktu eksekusi
        start = clock();
        selectionSort(temp, n);
        end = clock();
        cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
        printf("| %-15s | %15dk | %20.2f |\n", "Selection Sort", n / 1000,
cpu_time_used * 1000);

        // Menyimpan bilangan yang sudah terurut dengan selection sort
        sprintf(title, "Sorted Numbers (Selection Sort) - %d Numbers", n);
        saveToFile(filename, temp, n, title);
    }
    for (int n = 100000; n <= MAX; n += 100000)
    {
        // Copy array untuk digunakan dalam setiap algoritma pengurutan
        memcpy(temp, arr, n * sizeof(int));

        // Mengurutkan bilangan dengan insertion sort dan mencatat waktu eksekusi
        start = clock();
        insertionSort(temp, n);
        end = clock();
        cpu_time_used = ((double)(end - start)) / CLOCKS_PER_SEC;
```

```
        printf("| %-15s | %15dk | %20.2f |\n", "Insertion Sort", n / 1000,
cpu_time_used * 1000);

        // Menyimpan bilangan yang sudah terurut dengan insertion sort
        sprintf(title, "Sorted Numbers (Insertion Sort) - %d Numbers", n);
        saveToFile(filename, temp, n, title);
    }
    return 0;
}
```

Output max 1.000.000 dan kenaikan jumlah bilangan 100.000:

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2>
```

Output max 900.000 dan kenaikan jumlah bilangan 100.000

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2>
```

Output max 800.000 dan kenaikan jumlah bilangan 100.000

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2>
```

Output max 700.000 dan kenaikan jumlah bilangan 100.000

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2>
```

Output max 600.000 dan kenaikan jumlah bilangan 100.000

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2>
```

Output 500.000 dan kenaikan jumlah bilangan 100.000

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2>
```

Output 400.000 dan kenaikan jumlah bilangan 100.000

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2>
```

Output 300.000 dan kenaikan jumlah bilangan 100.000

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
```

Output 260.000 dan kenaikan jumlah bilangan 20.000

```
PS D:\sem4\StrukturData\tugas2> gcc 2208107010030_Simple_Sorting.c -o Simple_Sorting
PS D:\sem4\StrukturData\tugas2> ./Simple_Sorting
PS D:\sem4\StrukturData\tugas2>
```

Output 250.000 dan jenaikan jumlah bilangan 20.000

Jenis Algoritma	Jumlah Bilangan	Waktu Eksekusi (ms)
Bubble Sort	20k	1027.00
Bubble Sort	40k	4567.00
Bubble Sort	60k	10519.00
Bubble Sort	80k	18873.00
Bubble Sort	100k	29346.00
Bubble Sort	120k	43210.00
Bubble Sort	140k	58578.00
Bubble Sort	160k	77398.00
Bubble Sort	180k	99574.00
Bubble Sort	200k	120523.00
Bubble Sort	220k	145962.00
Bubble Sort	240k	175297.00
Selection Sort	20k	513.00
Selection Sort	40k	1972.00
Selection Sort	60k	4578.00
Selection Sort	80k	8194.00
Selection Sort	100k	12663.00
Selection Sort	120k	18007.00
Selection Sort	140k	25225.00
Selection Sort	160k	35231.00
Selection Sort	180k	40952.00
Selection Sort	200k	52690.00
Selection Sort	220k	63710.00
Selection Sort	240k	73200.00
Insertion Sort	20k	288.00
Insertion Sort	40k	989.00
Insertion Sort	60k	2327.00
Insertion Sort	80k	4149.00
Insertion Sort	100k	6323.00
Insertion Sort	120k	9993.00
Insertion Sort	140k	13005.00
Insertion Sort	160k	16363.00
Insertion Sort	180k	20689.00
Insertion Sort	200k	25989.00
Insertion Sort	220k	31873.00
Insertion Sort	240k	38955.00

Spesifikasi laptop:

Nama perangkat	desktop
Prosesor	11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz 3.11 GHz
RAM terinstal	8,00 GB (7,70 GB dapat digunakan)
ID perangkat	A7B166E4-337D-4F82-9D6E-7F76327BFEE2
ID Produk	00356-24549-34948-AAOEM
Jenis sistem	Sistem operasi 64-bit, prosesor berbasis x64
Pena dan sentuhan	Tidak ada input pena atau sentuhan yang tersedia untuk tampilan ini