TUGAS JOBSHEET 9

Disusun Oleh:

Qadrian Zakhri

23343081

Dosen pengampu :

Randi Proska Sandra, M.Sc

PROGRAM STUDI INFORMATIKA (S1)

DEPARTEMEN ELEKTRONIKA

FAKULTAS TEKNIK

UNIVERSITAS NEGERI PADANG

Source Code

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

// Fungsi untuk menukar
dua elemen
void swap(int* a, int*
b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Implementasi
Selection Sort
void selectionSort(int
arr[], int n) {
    for (int i = 0; i <
n - 1; i++) {
        int minIdx = i;
        for (int j = i
+ 1; j < n; j++) {
            if (arr[j]
< arr[minIdx]) {
                minIdx
= j;
            }
        }

swap(&arr[minIdx],
&arr[i]);
    }
}

// Fungsi untuk
```

```
menggabungkan dua
subarray
void merge(int arr[],
int l, int m, int r) {
    int n1 = m - l + 1;
    int n2 = r - m;

    int L[n1], R[n2];

    for (int i = 0; i <
n1; i++)
        L[i] = arr[l +
i];
    for (int j = 0; j <
n2; j++)
        R[j] = arr[m +
1 + j];

    int i = 0, j = 0, k
= l;
    while (i < n1 && j
< n2) {
        if (L[i] <=
R[j]) {
            arr[k] =
L[i];
            i++;
        } else {
            arr[k] =
R[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
```

```
        arr[k] = L[i];
        i++;
        k++;
    }

    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

// Implementasi Merge
Sort
void mergeSort(int
arr[], int l, int r) {
    if (l < r) {
        int m = l + (r
- l) / 2;

        mergeSort(arr,
l, m);
        mergeSort(arr,
m + 1, r);

        merge(arr, l,
m, r);
    }
}

// Fungsi untuk
mencetak array
void printArray(int
arr[], int size) {
    for (int i = 0; i <
size; i++)
```

```
        printf("%d ",
arr[i]);
    printf("\n");
}

int main() {
    int
selectionSortArray[] =
{64, 25, 12, 22, 11};
    int n =
sizeof(selectionSortArr
ay) /
sizeof(selectionSortArr
ay[0]);

    printf("Original
array for Selection
Sort: \n");

printArray(selectionSor
tArray, n);


selectionSort(selection
SortArray, n);

    printf("Sorted
array using Selection
Sort: \n");

printArray(selectionSor
tArray, n);

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

```
    n =
sizeof(mergeSortArray)
/
sizeof(mergeSortArray[0
]);

    printf("Original
array for Merge Sort:
\n");

printArray(mergeSortArr
ay, n);



mergeSort(mergeSortArra
y, 0, n - 1);

    printf("Sorted
array using Merge Sort:
\n");

printArray(mergeSortArr
ay, n);

    return 0;
}
```

Implementasi Selection Sort dan Merge Sort

1. Selection Sort

- Loop luar berjalan dari elemen pertama hingga elemen kedua terakhir.
- Loop dalam mencari elemen minimum dalam sub-array yang belum diurutkan.
- Elemen minimum ditukar dengan elemen pertama sub-array tersebut.

2. Merge Sort

- Fungsi mergeSort membagi array menjadi dua sub-array dan memanggil dirinya sendiri secara rekursif untuk masing-masing sub-array.

- Fungsi merge menggabungkan dua sub-array yang terurut menjadi satu array terurut dengan membandingkan elemen-elemen dari kedua sub-array.