

TUGAS JOBSHEET 9

Dosen pengampu :

Randi Proska Sandra, M.Sc



Disusun Oleh:

Rendi Aigo Brandon

NIM : 23343082

PROGRAM STUDI
INFORMATIKA(NK) DEPARTEMEN
ELEKTRONIKA FAKULTAS TEKNIK
UNIVERSITAS NEGERI PADANG

SOURCE CODE

```
#include <stdio.h>

// Deklarasi fungsi Selection Sort
void selectionSort(int arr[], int n) {
    int i, j, min_idx;
    // Iterasi melalui seluruh array
    for (i = 0; i < n-1; i++) {
        min_idx = i;
        // Cari elemen terkecil dari array yang belum terurut
        for (j = i+1; j < n; j++) {
            if (arr[j] < arr[min_idx]) {
                min_idx = j;
            }
        }
        // Tukar elemen terkecil dengan elemen pada indeks i
        int temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
}

// Deklarasi fungsi Merge Sort
void merge(int arr[], int left, int mid, int right) {
    int i, j, k;
    int n1 = mid - left + 1;
    int n2 = right - mid;

    // Buat array sementara untuk menyimpan bagian kiri dan kanan
    int L[n1], R[n2];

    // Salin elemen ke dalam array sementara
    for (i = 0; i < n1; i++)
        L[i] = arr[left + i];
```

```

for (j = 0; j < n2; j++)
    R[j] = arr[mid + 1 + j];

// Gabungkan kembali elemen-elemen yang terurut
i = 0;
j = 0;
k = left;
while (i < n1 && j < n2) {
    if (L[i] <= R[j]) {
        arr[k] = L[i];
        i++;
    } else {
        arr[k] = R[j];
        j++;
    }
    k++;
}

// Salin sisa elemen dari bagian kiri
while (i < n1) {
    arr[k] = L[i];
    i++;
    k++;
}

// Salin sisa elemen dari bagian kanan
while (j < n2) {
    arr[k] = R[j];
    j++;
    k++;
}
}

void mergeSort(int arr[], int left, int right) {

```

```

    if (left < right) {
        int mid = left + (right - left) / 2;

        // Panggil rekursif Merge Sort untuk kedua bagian
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);

        // Gabungkan bagian-bagian yang terurut
        merge(arr, left, mid, right);
    }
}

int main() {
    // Array untuk Selection Sort
    int arr1[] = {64, 25, 12, 22, 11};
    int n1 = sizeof(arr1)/sizeof(arr1[0]);

    printf("Array sebelum diurutkan (Selection Sort):\n");
    for (int i=0; i < n1; i++) {
        printf("%d ", arr1[i]);
    }
    printf("\n");

    // Panggil fungsi Selection Sort
    selectionSort(arr1, n1);

    printf("Array setelah diurutkan (Selection Sort):\n");
    for (int i=0; i < n1; i++) {
        printf("%d ", arr1[i]);
    }
    printf("\n");

    // Array untuk Merge Sort
    int arr2[] = {64, 25, 12, 22, 11};

```

```

int n2 = sizeof(arr2)/sizeof(arr2[0]);

printf("Array sebelum diurutkan (Merge Sort):\n");
for (int i=0; i < n2; i++) {
    printf("%d ", arr2[i]);
}
printf("\n");

// Panggil fungsi Merge Sort
mergeSort(arr2, 0, n2-1);

printf("Array setelah diurutkan (Merge Sort):\n");
for (int i=0; i < n2; i++) {
    printf("%d ", arr2[i]);
}
printf("\n");

return 0;
}

```

PENJELASAN

- Program ini terdiri dari dua fungsi sorting utama, yaitu **selectionSort** untuk Selection Sort dan **mergeSort** untuk Merge Sort.
- Di dalam fungsi **selectionSort**, iterasi dilakukan melalui seluruh array untuk mencari elemen terkecil pada setiap iterasi dan menukar elemen tersebut dengan elemen pada indeks yang sedang diperiksa.
- Di dalam fungsi **mergeSort**, array dibagi menjadi dua bagian, lalu masing-masing bagian diurutkan secara rekursif. Setelah itu, kedua bagian yang terurut digabungkan kembali menjadi satu array terurut.
- Di dalam fungsi **main**, kita membuat dua array yang akan diurutkan menggunakan Selection Sort dan Merge Sort, lalu mencetak array sebelum dan sesudah diurutkan.