

TUGAS JOBSHEET 10



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>

// Fungsi untuk melakukan Shell Sort
void shellSort(int arr[], int n) {
    for (int gap = n/2; gap > 0; gap /= 2) {
        for (int i = gap; i < n; i++) {
            int temp = arr[i];
            int j;
            for (j = i; j >= gap && arr[j - gap] > temp; j -= gap) {
                arr[j] = arr[j - gap];
            }
            arr[j] = temp;
        }
    }
}

// Fungsi untuk membagi array dan mengembalikan indeks pivot
int partition(int arr[], int low, int high) {
    int pivot = arr[high];
    int i = (low - 1);

    for (int j = low; j <= high - 1; j++) {
        if (arr[j] < pivot) {
            i++;
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
    int temp = arr[i + 1];
    arr[i + 1] = arr[high];
    arr[high] = temp;
    return (i + 1);
}

// Fungsi untuk melakukan Quick Sort
void quickSort(int arr[], int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);

        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

// 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 shellSortArray[] = {12, 34, 54, 2, 3};
    int n = sizeof(shellSortArray)/sizeof(shellSortArray[0]);

    printf("Original array for Shell Sort: \n");
    printArray(shellSortArray, n);
}
```

```

    shellSort(shellSortArray, n);

    printf("Sorted array using Shell Sort: \n");
    printArray(shellSortArray, n);

    int quickSortArray[] = {10, 7, 8, 9, 1, 5};
    n = sizeof(quickSortArray)/sizeof(quickSortArray[0]);

    printf("Original array for Quick Sort: \n");
    printArray(quickSortArray, n);

    quickSort(quickSortArray, 0, n - 1);

    printf("Sorted array using Quick Sort: \n");
    printArray(quickSortArray, n);

    return 0;
}

```

Implementasi Shell Sort dan Quick Sort

1. Shell Sort

- Loop luar mengurangi nilai gap dari setengah panjang array hingga 1.
- Loop dalam menjalankan insertion sort pada sub-array yang dipisahkan oleh nilai gap.
- Elemen-elemen dibandingkan dan dipindahkan sesuai kebutuhan untuk memastikan sub-array diurutkan.

2. Quick Sort

- Fungsi partition mengatur elemen-elemen array sedemikian rupa sehingga elemen yang lebih kecil dari pivot berada di sebelah kiri dan yang lebih besar di sebelah kanan.
- Fungsi quickSort memanggil partition untuk membagi array dan kemudian memanggil dirinya sendiri secara rekursif pada kedua sub-array yang dihasilkan.