

Jobsheet 10  
Praktikum Struktur Data

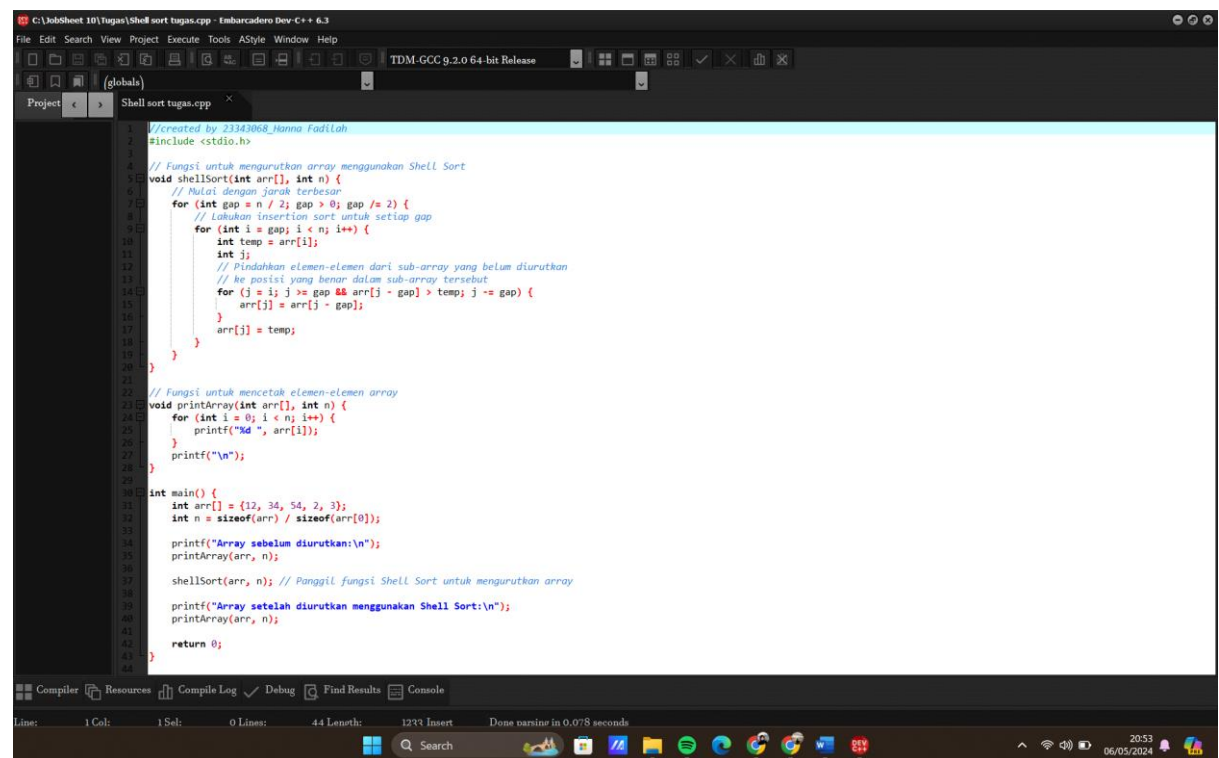


Dosen pengampu : Randi Proska Sandra, S.Pd, M.Sc  
Kode Kelas : 202323430158

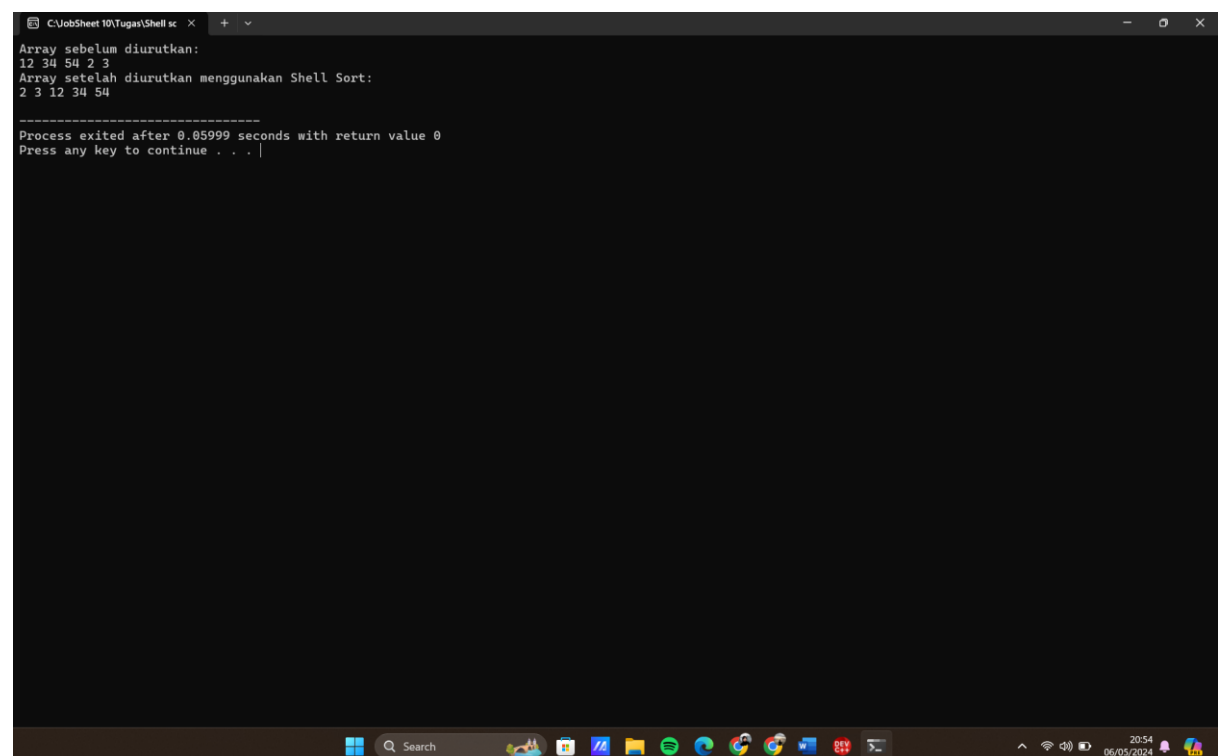
Disusun Oleh :  
Hanna Fadilah  
23343068

PROGRAM STUDI INFORMATIKA (NK)  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI PADANG  
2024

# 1. Shell sort



```
1 //created by 23342068_hanna Fadilah
2 #include <stdio.h>
3
4 // Fungsi untuk mengurutkan array menggunakan Shell Sort
5 void shellSort(int arr[], int n) {
6     // Mulai dengan jarak terbesar
7     for (int gap = n / 2; gap > 0; gap /= 2) {
8         // Lakukan insertion sort untuk setiap gap
9         for (int i = gap; i < n; i++) {
10             int temp = arr[i];
11             int j;
12             // Pindahkan elemen-elemen dari sub-array yang belum diurutkan
13             // ke posisi yang benar dalam sub-array tersebut
14             for (j = i; j >= gap && arr[j - gap] > temp; j -= gap) {
15                 arr[j] = arr[j - gap];
16             }
17             arr[j] = temp;
18         }
19     }
20 }
21
22 // Fungsi untuk mencetak elemen-elemen array
23 void printArray(int arr[], int n) {
24     for (int i = 0; i < n; i++) {
25         printf("%d ", arr[i]);
26     }
27     printf("\n");
28 }
29
30 int main() {
31     int arr[] = {12, 34, 54, 2, 3};
32     int n = sizeof(arr) / sizeof(arr[0]);
33
34     printf("Array sebelum diurutkan:\n");
35     printArray(arr, n);
36
37     shellSort(arr, n); // Panggil fungsi Shell Sort untuk mengurutkan array
38
39     printf("Array setelah diurutkan menggunakan Shell Sort:\n");
40     printArray(arr, n);
41
42     return 0;
43 }
```



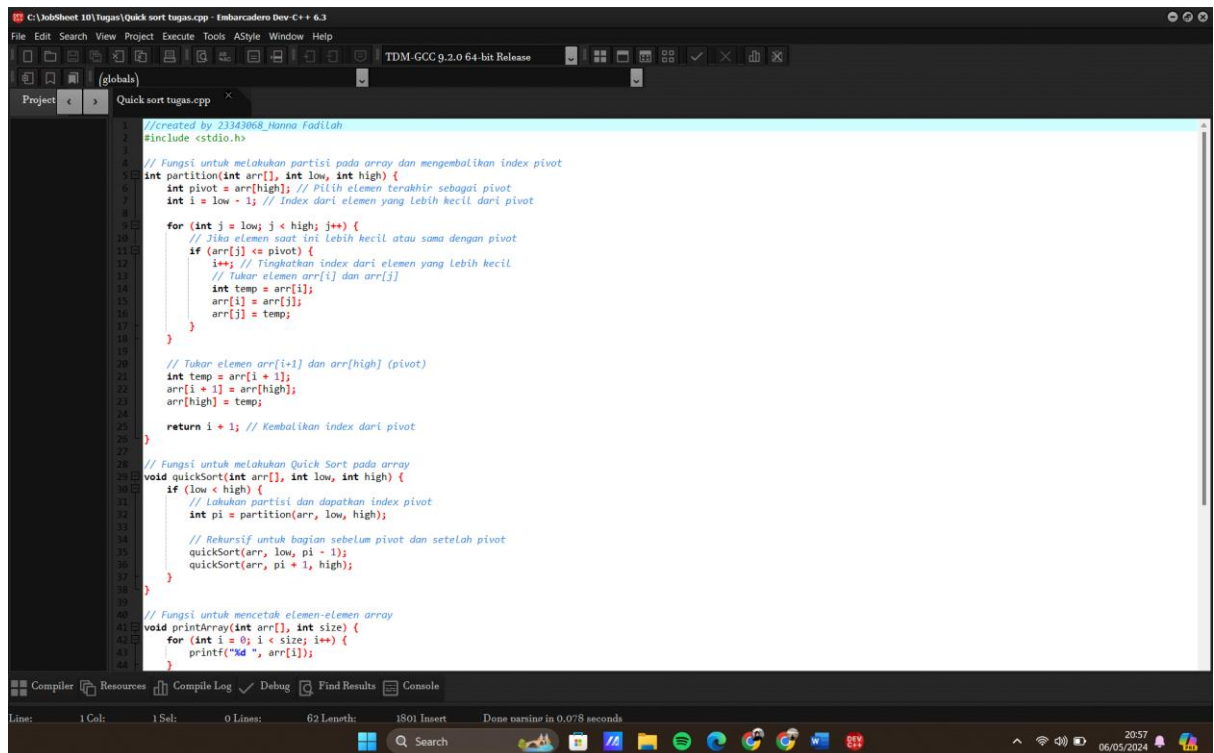
```
C:\JobSheet 10\Tugas\Shell sc
Array sebelum diurutkan:
12 34 54 2 3
Array setelah diurutkan menggunakan Shell Sort:
2 3 12 34 54

-----
Process exited after 0.05999 seconds with return value 0
Press any key to continue . . .
```

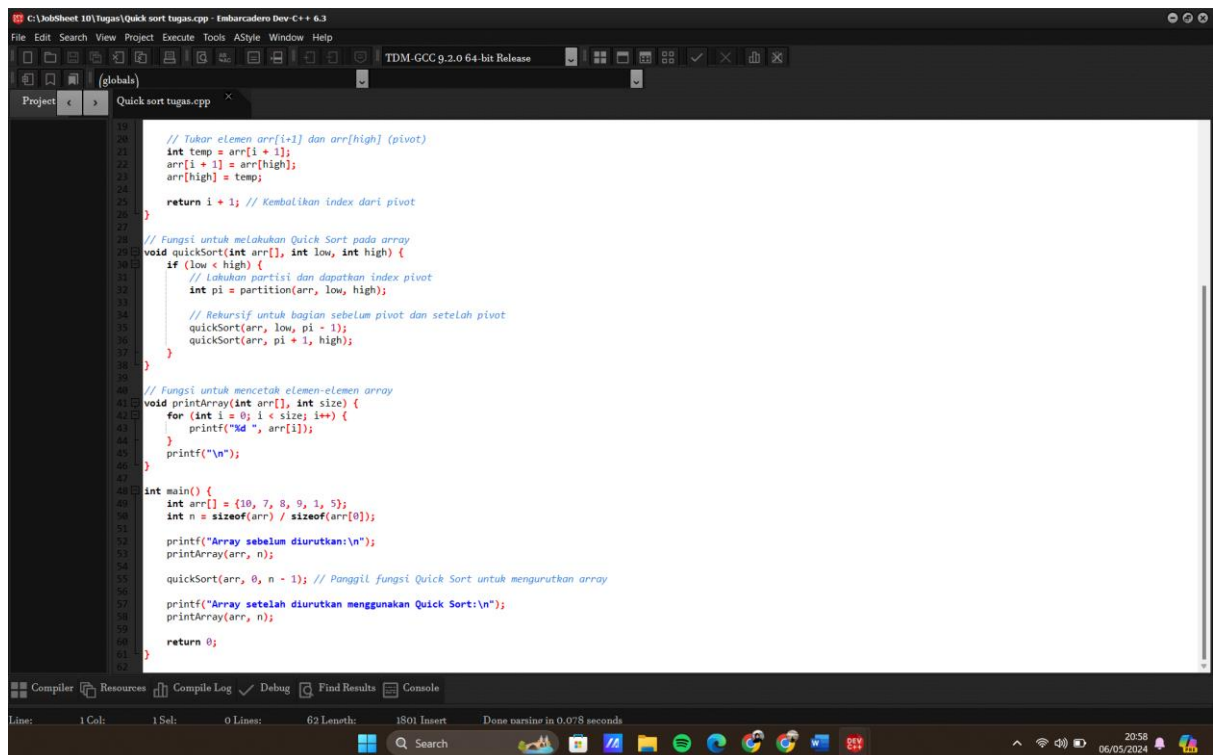
## Penjelasan

- Algoritma Shell Sort adalah variasi dari Insertion Sort yang mengurutkan elemen-elemen dengan membandingkan elemen yang terletak jauh satu sama lain.
- Prinsip kerjanya adalah dengan menggunakan jarak yang lebih besar antara elemen yang akan dibandingkan pada awalnya, dan kemudian mengurangnya secara bertahap hingga mencapai jarak 1.
- Pada setiap iterasi, Shell Sort menggunakan Insertion Sort pada sub-array dengan jarak tertentu. Hal ini memungkinkan elemen-elemen yang jauh dari posisi akhirnya untuk bergerak ke posisi yang benar dengan lebih cepat, sehingga mempercepat proses pengurutan secara keseluruhan.

## 2. Quick sort



```
1 //created by 23143068_nanna Fadilah
2 #include <stdio.h>
3
4 // Fungsi untuk melakukan partisi pada array dan mengembalikan index pivot
5 int partition(int arr[], int low, int high) {
6     int pivot = arr[high]; // Pilih elemen terakhir sebagai pivot
7     int i = low - 1; // Index dari elemen yang lebih kecil dari pivot
8
9     for (int j = low; j < high; j++) {
10         // Jika elemen saat ini lebih kecil atau sama dengan pivot
11         if (arr[j] <= pivot) {
12             i++; // Tingkatkan index dari elemen yang lebih kecil
13             // Tukar elemen arr[i] dan arr[j]
14             int temp = arr[i];
15             arr[i] = arr[j];
16             arr[j] = temp;
17         }
18     }
19
20     // Tukar elemen arr[i+1] dan arr[high] (pivot)
21     int temp = arr[i + 1];
22     arr[i + 1] = arr[high];
23     arr[high] = temp;
24
25     return i + 1; // Kembalikan index dari pivot
26 }
27
28 // Fungsi untuk melakukan Quick Sort pada array
29 void quickSort(int arr[], int low, int high) {
30     if (low < high) {
31         // Lakukan partisi dan dapatkan index pivot
32         int pi = partition(arr, low, high);
33
34         // Rekursif untuk bagian sebelum pivot dan setelah pivot
35         quickSort(arr, low, pi - 1);
36         quickSort(arr, pi + 1, high);
37     }
38 }
39
40 // Fungsi untuk mencetak elemen-elemen array
41 void printArray(int arr[], int size) {
42     for (int i = 0; i < size; i++) {
43         printf("%d ", arr[i]);
44     }
45 }
```



```
19 // Tukar elemen arr[i+1] dan arr[high] (pivot)
20 int temp = arr[i + 1];
21 arr[i + 1] = arr[high];
22 arr[high] = temp;
23
24 return i + 1; // Kembalikan index dari pivot
25 }
26
27 // Fungsi untuk melakukan Quick Sort pada array
28 void quickSort(int arr[], int low, int high) {
29     if (low < high) {
30         // Lakukan partisi dan dapatkan index pivot
31         int pi = partition(arr, low, high);
32
33         // Rekursif untuk bagian sebelum pivot dan setelah pivot
34         quickSort(arr, low, pi - 1);
35         quickSort(arr, pi + 1, high);
36     }
37 }
38
39 // fungsi untuk mencetak elemen-elemen array
40 void printArray(int arr[], int size) {
41     for (int i = 0; i < size; i++) {
42         printf("%d ", arr[i]);
43     }
44     printf("\n");
45 }
46
47 int main() {
48     int arr[] = {10, 7, 8, 9, 1, 5};
49     int n = sizeof(arr) / sizeof(arr[0]);
50
51     printf("Array sebelum diurutkan:\n");
52     printArray(arr, n);
53
54     quickSort(arr, 0, n - 1); // Panggil fungsi Quick Sort untuk mengurutkan array
55
56     printf("Array setelah diurutkan menggunakan Quick Sort:\n");
57     printArray(arr, n);
58
59     return 0;
60 }
61
62 }
```

```
C:\JobSheet 10\Tugas\Quicksort >
Array sebelum diurutkan:
10 7 8 9 1 5
Array setelah diurutkan menggunakan Quick Sort:
1 5 7 8 9 10

-----
Process exited after 0.05578 seconds with return value 0
Press any key to continue . . .
```

## Penjelasan

- Quick Sort adalah algoritma pengurutan yang cepat dan efisien.
- Algoritma ini memilih sebuah elemen sebagai pivot, kemudian mempartisi array menjadi dua bagian - satu bagian dengan elemen yang lebih kecil dari pivot, dan yang lainnya dengan elemen yang lebih besar.
- Proses partisi ini diulangi secara rekursif untuk setiap bagian hingga seluruh array terurut.
- Pemilihan pivot yang baik dapat mempengaruhi performa algoritma. Dalam contoh ini, pivot dipilih sebagai elemen terakhir dari array.
- Melalui proses partisi, elemen-elemen akan terus tertukar sehingga elemen-elemen yang lebih kecil dari pivot berada di sebelah kiri pivot, dan elemen-elemen yang lebih besar berada di sebelah kanan pivot.