

Jobsheet 9
Praktikum Struktur Data

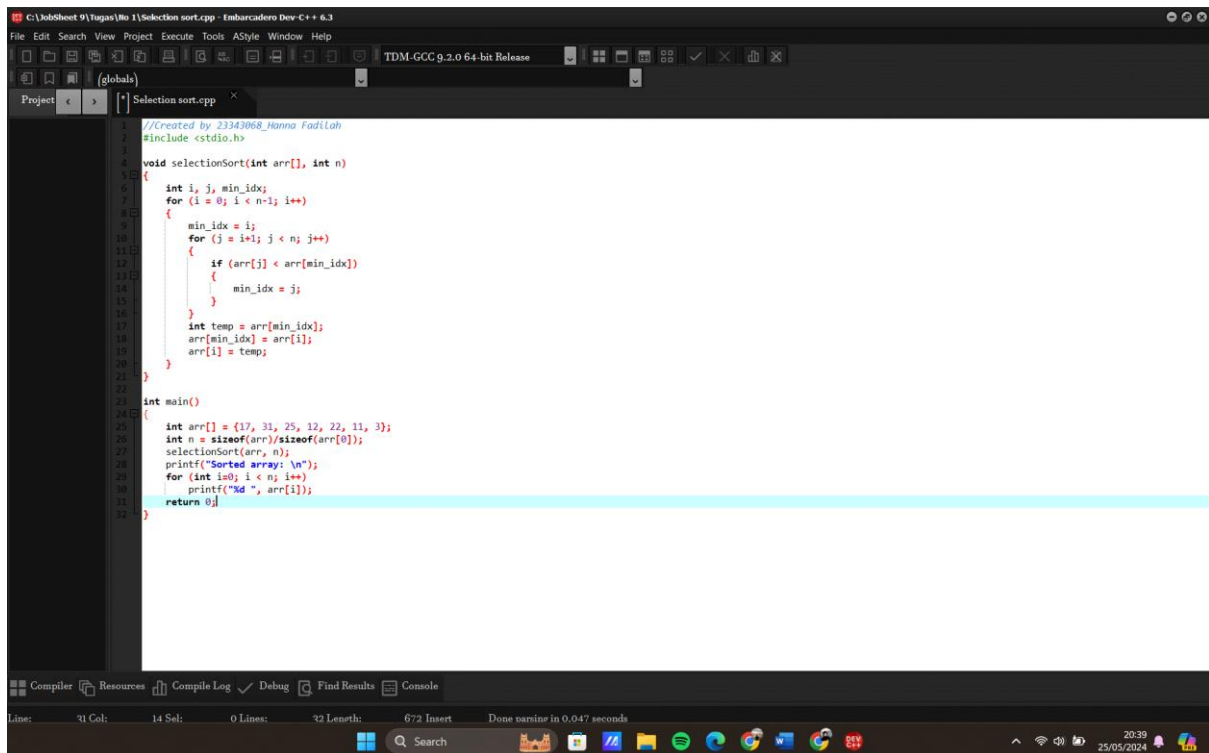


Dosen pengampu : Randi Proska Sandra, S.Pd, M.Sc
Kode Kelas : 202323430158

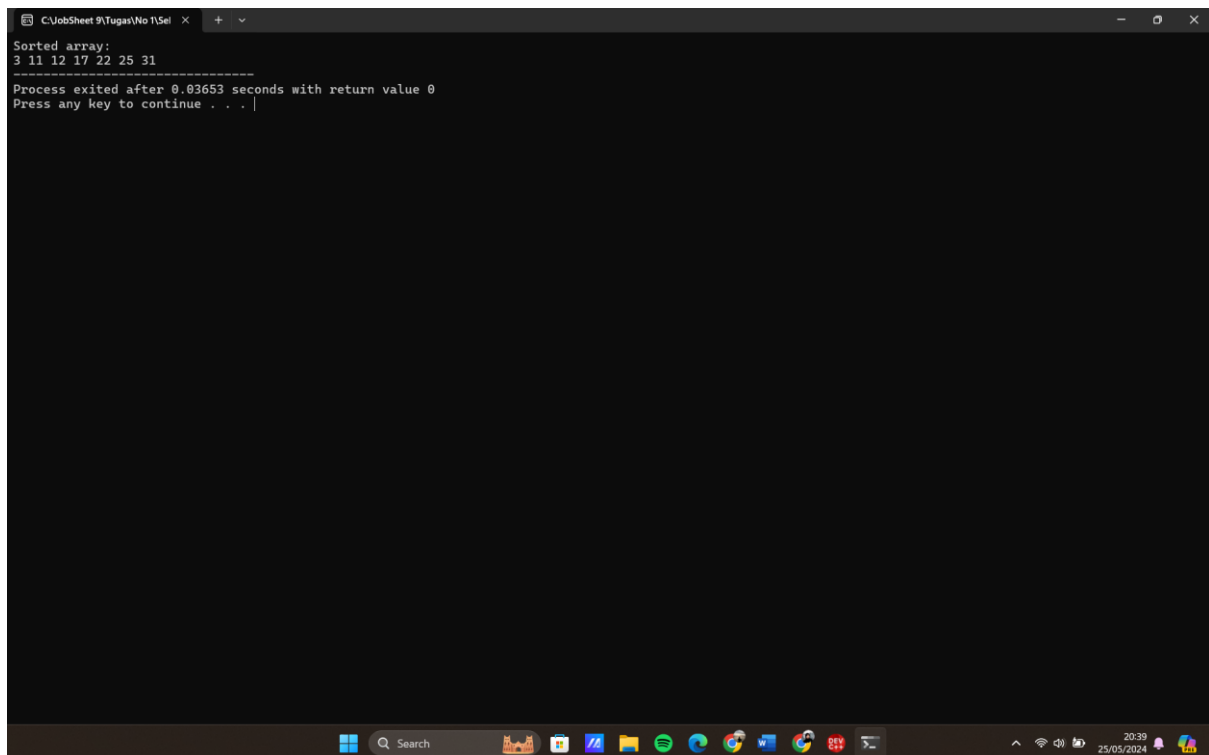
Disusun Oleh :
Hanna Fadilah
23343068

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

1. Selection Sort



```
1 //Created by 23343068_nanna Fadilah
2 #include <stdio.h>
3
4 void selectionSort(int arr[], int n)
5 {
6     int i, j, min_idx;
7     for (i = 0; i < n-1; i++)
8     {
9         min_idx = i;
10        for (j = i+1; j < n; j++)
11        {
12            if (arr[j] < arr[min_idx])
13            {
14                min_idx = j;
15            }
16        }
17        int temp = arr[min_idx];
18        arr[min_idx] = arr[i];
19        arr[i] = temp;
20    }
21 }
22
23 int main()
24 {
25     int arr[] = {17, 31, 25, 12, 22, 11, 3};
26     int n = sizeof(arr)/sizeof(arr[0]);
27     selectionSort(arr, n);
28     printf("Sorted array: \n");
29     for (int i=0; i < n; i++)
30         printf("%d ", arr[i]);
31     return 0;
32 }
```

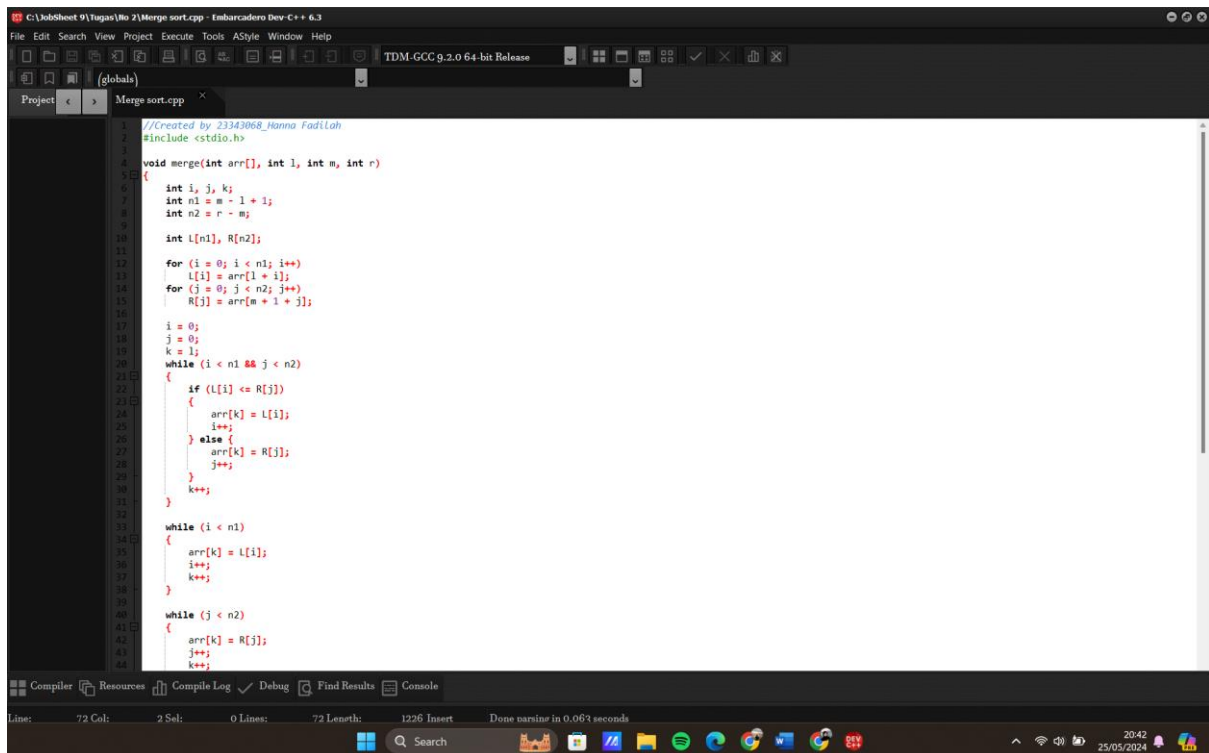


```
Sorted array:
3 11 12 17 22 25 31
-----
Process exited after 0.03653 seconds with return value 0
Press any key to continue . . .
```

Penjelasan

Selection sort adalah algoritma pengurutan yang sederhana dan mudah dipahami, tetapi tidak efisien untuk array besar karena kompleksitas waktunya. Algoritma ini lebih baik digunakan pada array kecil atau sebagai pengenalan konsep pengurutan dasar.

2. Merge Sort



```
//Created by 23343068_nanna Fadilah
#include <stdio.h>

void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

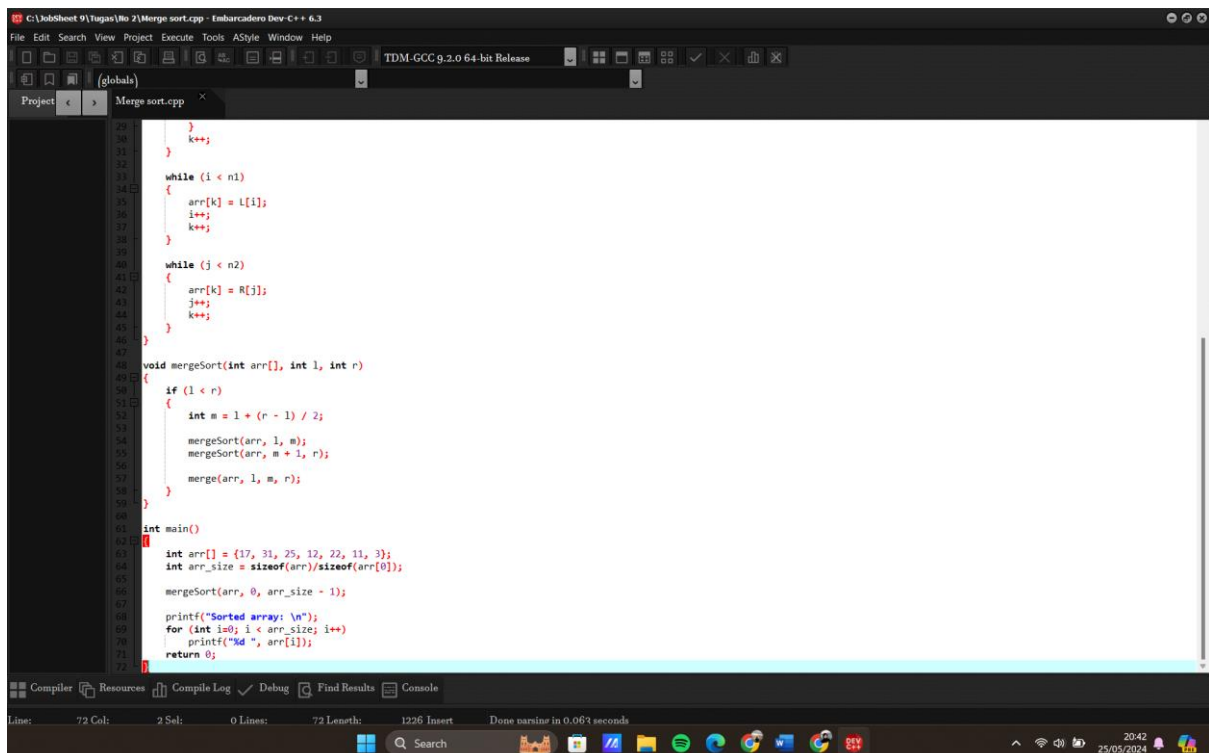
    int L[n1], R[n2];

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

    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++;
    }
}
```



```

    }
    k++;
}

while (i < n1)
{
    arr[k] = L[i];
    i++;
    k++;
}

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

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);
    }
}

int main()
{
    int arr[] = {17, 31, 25, 12, 22, 11, 3};
    int arr_size = sizeof(arr)/sizeof(arr[0]);

    mergeSort(arr, 0, arr_size - 1);

    printf("Sorted array: \n");
    for (int i=0; i < arr_size; i++)
        printf("%d ", arr[i]);

    return 0;
}
```

Penjelasan

Merge sort adalah algoritma pengurutan yang efisien dan stabil dengan kompleksitas waktu rata-rata. Algoritma ini cocok untuk digunakan pada array besar dan berbagai jenis data. Merge sort membutuhkan memori tambahan untuk menyimpan subarray selama proses penggabungan.