

LAPORAN PRAKTIKUM SORTING DALAM BAHASA C



Oleh:

Nama : L Hafidl Alkhair
NIM : 2023903430060
Kelas : TRKJ 1.C
Jurusan : Teknologi Informasi dan Komputer
Program Studi : Teknologi Rekayasa Komputer Jaringan
Dosen Pembimbing : Indrawati, SST. MT



**POLITEKNIK NEGERI LHOKSEUMAWE
TAHUN AJARAN 2023/2023**

LEMBAR PENGESAHAN

No. Praktikum : 10 /TIK/TRKJ-1C/ Data Structure And Algorithms Practice

Judul : Laporan Pratikum Sorting

Nama : L Hafidl Alkhair

NIM : 2023903430060

Kelas : TRKJ-1C

Jurusan : Teknologi Informasi Dan Komputer

Prodi : Teknologi Rekayasa Komputer Jaringan

Tanggal Praktikum : 20 November 2023

Tanggal Penyerahan : 27 November 2023

Buketrata, 27 November 2023
Dosen Pembimbing,

Indrawati, SST.MT
Nip. 19740815 200112 2 001

1.1. PENDAHULUAN

A. Tujuan

1. Mampu menguasai konsep penulisan program Sorting dalam bahasa C.
2. Mampu mengimplementasi program Sorting dalam bahasa C.
3. Mampu menyelesaikan masalah dengan memanfaatkan Sorting dalam bahasa C.

B. Alat dan Bahan:

- Komputer atau Laptop Dengan Kompiler C (GCC)
- Editor Teks (Misalnya Notepad ++, DEV – C++)
- Buku Panduan atau Materi Pratikum
- Kertas dan Pensil
- Ruang Pratikum

C. Dasar Teori

Dasar teori tentang sorting dalam bahasa C mencakup konsep bahwa Sorting merupakan sebuah proses yang sangat penting dalam dunia pemrograman dan ilmu komputer. Proses ini melibatkan penyusunan kembali suatu himpunan objek dengan menggunakan aturan tertentu. Tujuan dari sorting adalah untuk mengorganisir data secara terstruktur sehingga memudahkan akses dan pencarian informasi. Sorting juga dapat dianggap sebagai sebuah algoritma, yaitu langkah-langkah sistematis yang digunakan untuk meletakkan kumpulan elemen data ke dalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen. Dalam konteks sorting, terdapat dua macam urutan yang umum digunakan, yaitu urutan naik (ascending) dan urutan turun (descending). Urutan naik mengurutkan data dari nilai terkecil hingga nilai terbesar, sementara urutan turun mengurutkan data dari nilai terbesar hingga nilai terkecil. Pemilihan jenis urutan tergantung pada kebutuhan dan konteks penggunaan data tersebut.

Beberapa metode pengurutan yang umum digunakan dalam bahasa pemrograman C mencakup metode pengurutan gelembung (bubble sort) dan metode pengurutan pilih (selection sort). Metode pengurutan gelembung bekerja dengan cara membandingkan dan menukar elemen data secara berurutan hingga seluruh himpunan data terurut. Sedangkan metode pengurutan pilih memilih elemen dengan nilai terkecil atau terbesar secara berulang-ulang dan menukarnya dengan elemen di posisi tertentu hingga seluruh himpunan data terurut. Dengan kemajuan teknologi dan kebutuhan akan pengolahan data yang semakin kompleks, pemahaman yang mendalam terhadap algoritma sorting menjadi kunci dalam pengembangan perangkat lunak yang efisien dan responsif.

1.2. URAIAN PRAKTIKUM

1. Program Bubble Sort

```
#include <stdio.h>

void bubbleSort(int arr[], int n) {

    for (int i = 0; i < n - 1; i++) {

        for (int j = 0; j < n - i - 1; j++) {

            if (arr[j] > arr[j + 1]) {

                // Menukar elemen jika urutannya
                tidak benar

                int temp = arr[j];

                arr[j] = arr[j + 1];

                arr[j + 1] = temp;

            }

        }

    }

}
```

```

}

int main() {

    int arr[] = {64, 34, 25, 12, 22, 11, 90};

    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Array sebelum diurutkan:\n");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    bubbleSort(arr, n);

    printf("\nArray setelah diurutkan (Bubble Sort):\n");

    for (int i = 0; i < n; i++) {

        printf("%d ", arr[i]);

    }

    return 0;

}

```

- **Output:**

```

Array sebelum diurutkan:
64 34 25 12 22 11 90
Array setelah diurutkan (Bubble Sort):
11 12 22 25 34 64 90
-----

```

- **Penjelasan:**

1. Fungsi Bubble Sort:

- Fungsi ini menerima dua parameter, yaitu array arr yang akan diurutkan dan jumlah elemen dalam array (n).
- Menggunakan nested loop untuk membandingkan dan menukar

elemen-elemen array secara berpasangan.

- Jumlah iterasi luar (i) mengatur sejauh mana elemen array telah terurut, sedangkan jumlah iterasi dalam (j) digunakan untuk membandingkan dan menukar elemen-elemen yang berdekatan.

2. Fungsi main:

- Mendeklarasikan array arr yang berisi bilangan bulat.
- Menghitung jumlah elemen dalam array menggunakan sizeof dan membaginya dengan ukuran satu elemen.
- Menampilkan array sebelum diurutkan.
- Memanggil fungsi bubbleSort untuk mengurutkan array.
- Menampilkan array setelah diurutkan menggunakan algoritma Bubble Sort.

2. Selection Sort:

```
#include <stdio.h>

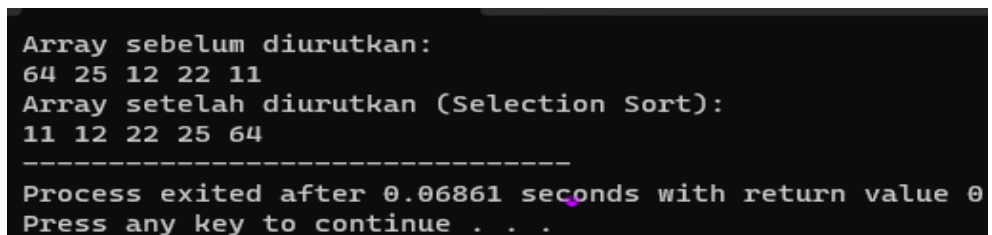
void selectionSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < n; j++) {
            if (arr[j] < arr[minIndex]) {
                minIndex = j;
            }
        }
        // Menukar elemen terkecil dengan elemen
        pertama pada iterasi saat ini
        int temp = arr[minIndex];
        arr[minIndex] = arr[i];
        arr[i] = temp;
    }
}
```

```

int main() {
    int arr[] = {64, 25, 12, 22, 11};
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("Array sebelum diurutkan:\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    selectionSort(arr, n);
    printf("\nArray setelah diurutkan (Selection Sort):\n");
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}

```

- **Output:**



```

Array sebelum diurutkan:
64 25 12 22 11
Array setelah diurutkan (Selection Sort):
11 12 22 25 64
-----
Process exited after 0.06861 seconds with return value 0
Press any key to continue . . .

```

- **Penjelasan:**

1. Fungsi selectionSort:

- Fungsi ini menerima dua parameter, yaitu array arr yang akan diurutkan dan jumlah elemen dalam array (n).
- Menggunakan dua loop bersarang untuk mencari nilai terkecil di antara elemen-elemen yang belum terurut.
- Pada setiap iterasi luar, mencari indeks elemen terkecil pada array yang belum terurut dan menukarnya dengan elemen pertama pada

iterasi saat ini.

2. Fungsi main:

- Mendeklarasikan array arr yang berisi bilangan bulat.
- Menghitung jumlah elemen dalam array menggunakan sizeof dan membaginya dengan ukuran satu elemen.
- Menampilkan array sebelum diurutkan.
- Memanggil fungsi selectionSort untuk mengurutkan array.
- Menampilkan array setelah diurutkan menggunakan algoritma Selection Sort.

3. Insertion Sort:

```
#include <stdio.h>

void insertionSort(int arr[], int n) {
    for (int i = 1; i < n; i++) {
        int key = arr[i];
        int j = i - 1;

        // Memindahkan elemen-elemen yang lebih besar
        // dari key ke posisi lebih belakang
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

int main() {
    int arr[] = {12, 11, 13, 5, 6};
    int n = sizeof(arr) / sizeof(arr[0]);
```



```

printf("Array sebelum diurutkan:\n");

for (int i = 0; i < n; i++) {
    printf("%d ", arr[i]);
}

insertionSort(arr, n);

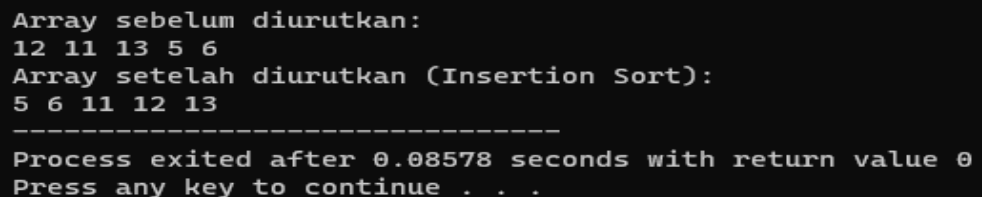
printf("\nArray setelah diurutkan (Insertion Sort):\n");

for (int i = 0; i < n; i++) {
    printf("%d ", arr[i]);
}

return 0;
}

```

- **Output:**



```

Array sebelum diurutkan:
12 11 13 5 6
Array setelah diurutkan (Insertion Sort):
5 6 11 12 13
-----
Process exited after 0.08578 seconds with return value 0
Press any key to continue . . .

```

- **Penjelasan:**

1. Fungsi insertionSort:

- Fungsi ini menerima dua parameter, yaitu array arr yang akan diurutkan dan jumlah elemen dalam array (n).
- Menggunakan loop luar untuk menelusuri seluruh array, dimulai dari elemen kedua.
- Pada setiap iterasi luar, elemen saat ini (key) dibandingkan dengan elemen-elemen di bagian yang sudah terurut dari array, dan elemen-elemen yang lebih besar dipindahkan ke posisi yang lebih belakang.

- Proses ini diulang hingga seluruh array terurut.
2. Fungsi main:
 - Mendeklarasikan array `arr` yang berisi bilangan bulat.
 - Menghitung jumlah elemen dalam array menggunakan `sizeof` dan membaginya dengan ukuran satu elemen.
 - Menampilkan array sebelum diurutkan.
 - Memanggil fungsi `insertionSort` untuk mengurutkan array.
 - Menampilkan array setelah diurutkan menggunakan algoritma Insertion Sort.

1.3. PENUTUP

A. Kesimpulan

Dalam praktikum tadi, telah diajarkan implementasi tiga algoritma pengurutan sederhana dalam bahasa pemrograman C, yaitu Bubble Sort, Selection Sort, dan Insertion Sort. Berikut adalah kesimpulan dari praktikum tersebut:

1. Bubble Sort:
 - Algoritma sederhana yang bekerja dengan membandingkan dan menukar elemen berturut-turut yang tidak berurutan.
 - Kurang efisien untuk jumlah data besar karena memiliki kompleksitas waktu $O(n^2)$.
 - Mudah diimplementasikan dan dapat memberikan hasil yang benar untuk data yang sudah hampir terurut.
2. Selection Sort:
 - Algoritma sederhana yang secara berulang mencari elemen terkecil dari array yang belum terurut dan menukarnya dengan elemen pertama dari array yang belum terurut.
 - Lebih efisien dalam hal pertukaran dibandingkan Bubble Sort, tetapi tetap memiliki kompleksitas waktu $O(n^2)$.
 - Cocok untuk jumlah data kecil atau saat sedang mencari elemen terkecil.

3. Insertion Sort:

- Algoritma yang memasukkan elemen-elemen ke dalam bagian yang sudah terurut dari array.
- Efisien untuk data yang hampir terurut atau jumlah data kecil, karena memiliki kompleksitas waktu $O(n)$ dalam kasus terbaik.
- Lebih efisien dalam hal pertukaran dibandingkan Bubble Sort dan Selection Sort.

4. Kesimpulan Umum:

- Pemilihan algoritma pengurutan tergantung pada kebutuhan dan sifat data yang dihadapi.
- Algoritma dengan kompleksitas waktu $O(n^2)$ kurang efisien untuk data yang besar.
- Algoritma pengurutan sederhana seperti Bubble Sort, Selection Sort, dan Insertion Sort cocok untuk memahami dasar-dasar algoritma pengurutan.

Praktikum ini memberikan pemahaman praktis tentang implementasi algoritma pengurutan dalam bahasa C dan pentingnya memilih algoritma yang sesuai dengan karakteristik data yang diurutkan. Pemahaman ini dapat membantu dalam pengembangan dan pemilihan algoritma pengurutan yang efisien dalam proyek-proyek pemrograman.

DAFTAR PUSTAKA

- <https://id.scribd.com/doc/314161174/Laporan-Pemograman-Bahasa-C-Sorting>
https://www.academia.edu/29021842/LAPORAN_PRAKTIKUM_II_SORTING_PENGURUTAN_SORTING_PENGURUTAN
<https://www.sanfoundry.com/c-program-sort-array-ascending-order/>