**LAPORAN PRAKTIKUM**

**S 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 2023/2024**

**LEMBAR PENGESAHAN**

No. Praktikum : 11 /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 : 27 November 2023

Tanggal Penyerahan : 4 Desember 2023

|  |
| --- |
| Buketrata, 4 November 2023 |
| Dosen Pembimbing, |
| Indrawati, SST.MT |
| Nip. 19740815 200112 2 001 |

**1.1. PENDAHULUAN**

## 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.

## 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 Praktikum

## 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**

## 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");

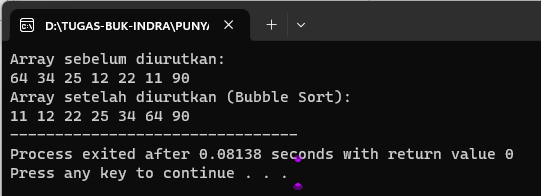
for (int i = 0; i < n; i++) {

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

}

return 0;

}

* **Output:**
* **Penjelasan:**

1. Fungsi Bubble Sort:

* Fungsi ini didefinisikan sebagai void bubbleSort(int arr[], int n).
* Menggunakan dua loop bersarang: loop pertama mengontrol iterasi melalui array, dan loop kedua membandingkan dan menukar elemen-elemen.
* Pada setiap iterasi loop pertama, elemen terbesar "muncul" ke posisi akhir array.
* Tujuan akhir dari fungsi ini adalah mengurutkan array yang diberikan.

1. Fungsi main:

* Fungsi utama (main) inisialisasi array dengan nilai tertentu, dan menghitung ukuran array (n).
* Mencetak array sebelum pengurutan.
* Memanggil fungsi bubbleSort untuk mengurutkan array.
* Mencetak array setelah pengurutan menggunakan algoritma Bubble Sort.
* Tujuan utama dari main adalah mendemonstrasikan cara menggunakan dan melihat hasil dari algoritma Bubble Sort.

Program ini secara keseluruhan memberikan gambaran visual tentang cara Bubble Sort mengurutkan array dengan menukar elemen-elemen yang tidak berurutan. Hasilnya, array akan terurut dari elemen terkecil hingga elemen terbesar setelah proses pengurutan selesai.

## 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");

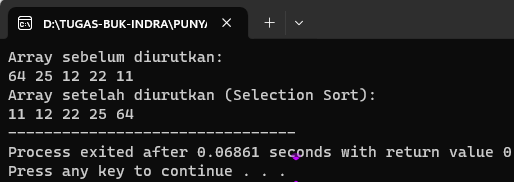
for (int i = 0; i < n; i++) {

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

}

return 0;

}

* **Output:**
* **Penjelasan:**

1. Header File:

Program dimulai dengan memasukkan header file <stdio.h>, yang menyediakan fungsi input-output standar seperti printf yang digunakan untuk menampilkan output di layar.

1. Fungsi Selection Sort:

Program mendefinisikan fungsi selectionSort yang menerima array dan ukuran array sebagai parameter. Fungsi ini mengurutkan array menggunakan algoritma Selection Sort. Proses pengurutan dilakukan dengan iterasi melalui array dan pada setiap iterasi mencari elemen terkecil di bagian sisa array dan menukarnya dengan elemen pertama pada iterasi.

1. Fungsi Utama (main):

* Membuat array awal yang akan diurutkan.
* Menampilkan array awal sebelum diurutkan.
* Memanggil fungsi selectionSort untuk mengurutkan array.
* Menampilkan array setelah diurutkan menggunakan algoritma Selection Sort.

1. Algoritma Selection Sort:

* Iterasi pertama memilih elemen terkecil dan menukarnya dengan elemen pertama.
* Iterasi kedua memilih elemen terkecil dari elemen kedua hingga akhir array dan menukarnya dengan elemen kedua.
* Proses ini berlanjut hingga seluruh array diurutkan.

1. Output:

Program menampilkan array sebelum diurutkan dan setelah diurutkan menggunakan algoritma Selection Sort.

## 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");

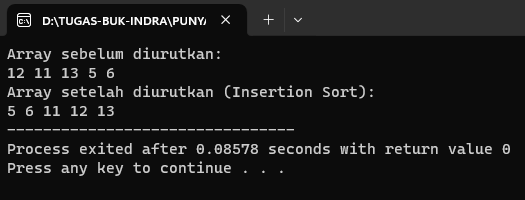
for (int i = 0; i < n; i++) {

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

}

return 0;

}

* **Output:**
* **Penjelasan:**

1. Inisialisasi Array:

Program dimulai dengan inisialisasi sebuah array arr yang berisi nilai-nilai integer, seperti 12, 11, 13, 5, dan 6. Variabel n dihitung untuk menyimpan jumlah elemen dalam array.

1. Menampilkan Array Sebelum Diurutkan:

Selanjutnya, program menampilkan isi array sebelum proses pengurutan. Ini dilakukan dengan menggunakan perulangan for untuk mencetak setiap elemen array.

1. Pengurutan Menggunakan Insertion Sort:

Program memanggil fungsi insertionSort untuk mengurutkan array. Algoritma Insertion Sort bekerja dengan membandingkan setiap elemen dengan elemen sebelumnya dan menyusunnya ke dalam urutan yang benar.

1. Implementasi Insertion Sort:

Fungsi insertionSort memiliki perulangan yang dimulai dari indeks kedua array. Setiap iterasi melibatkan pembandingan dan pertukaran elemen-elemen array untuk menyusunnya dalam urutan yang benar.

1. Menampilkan Array Setelah Diurutkan:

Setelah proses pengurutan selesai, program menampilkan isi array kembali. Kali ini, elemen-elemen array sudah diurutkan menggunakan algoritma Insertion Sort.

Dengan demikian, program ini memberikan contoh konkret tentang bagaimana algoritma Insertion Sort dapat diimplementasikan dalam bahasa pemrograman C untuk mengurutkan sebuah array. Prosesnya melibatkan perbandingan dan pertukaran elemen-elemen sehingga array diatur secara terurut.

# **1.3.** **PENUTUP**

## Kesimpulan

Dalam praktikum ini, kami tidak hanya mempelajari konsep dasar pemrograman C/C++ tetapi juga menerapkannya dalam konteks Sorting. Fokus utama praktikum adalah pada pemahaman dan penerapan konsep Sorting dalam bahasa pemrograman C/C++. Konsep Sorting sangat penting dalam mengorganisir dan menyusun data dalam suatu urutan tertentu.

Algoritma Sorting membantu mengurutkan elemen-elemen dalam suatu struktur data sehingga memudahkan pencarian dan pengelolaan data. Implementasi Sorting memungkinkan kita untuk menyusun data dengan cara tertentu, seperti urutan numerik atau alfabet, sehingga mempermudah analisis dan manipulasi data.

Contoh kode yang telah disajikan memberikan gambaran praktis tentang bagaimana menggunakan program untuk mengimplementasikan algoritma Sorting dalam program Anda. Dengan demikian, kami dapat lebih memahami bagaimana Sorting dapat diintegrasikan ke dalam solusi program kami.

Praktikum ini memberikan kesempatan bagi kami untuk mengasah keterampilan pemrograman kami dan meningkatkan kemampuan dalam memahami dan mengelola Sorting secara efisien. Penerapan konsep Sorting membantu kami memahami bagaimana mengatasi situasi di mana data harus diatur sesuai dengan urutan tertentu, meningkatkan efisiensi dan performa program.

Secara keseluruhan, praktikum ini tidak hanya menjadi landasan bagi pemahaman dasar dalam pemrograman C/C++, tetapi juga membuka wawasan tentang penggunaan Sorting dalam menyelesaikan masalah pemrograman yang lebih kompleks.

**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/>