**Laporan Hasil Praktikum**

****Praktikum 06: Insertion dan Selection Sort

**Oleh :**

Tiyyasha Ananda Mufti Hanifah (5223600011)

**Program Studi D4 Teknologi Game**

**Departemen Teknologi Multimedia Kreatif**

**Politeknik Elektronika Negeri Surabaya**

**2024**

1. **Dasar Teori**

Algoritma pengurutan bertugas mengatur ulang elemen dalam suatu kumpulan data berdasarkan urutan tertentu. Algoritma ini memainkan peran penting dalam berbagai aplikasi, termasuk pengolahan data, pemrograman, dan analisis statistik. Program pengurutan data pegawai umumnya menggunakan algoritma Insertion Sort, Selection Sort, atau algoritma pengurutan bawaan bahasa pemrograman (seperti sort dalam C++).

1. **Alat dan Bahan**

Alat yang digunakan pada percobaan menggambar kali ini adalah :

1. Laptop/PC
2. C++ compiler
3. Internet
4. **Analisa**
5. Buatlah project baru untuk Latihan dan implementasikan pengurutan data Pegawai pada tugas pendahuluan dengan ketentuan :.
6. Metode pengurutan dapat dipilih.
7. Pengurutan dapat dipilih secara urut naik atau turun.
8. Pengurutan dapat dipilih berdasarkan NIP dan NAMA.
9. Gunakan struktur data array.

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

// Define Employee structure

struct Employee {

string NIP;

string NAMA;

};

// Function to compare employees based on NIP (ascending order)

bool compareNIPAscending(const Employee& e1, const Employee& e2) {

return e1.NIP < e2.NIP;

}

// Function to compare employees based on NIP (descending order)

bool compareNIPDescending(const Employee& e1, const Employee& e2) {

return e1.NIP > e2.NIP;

}

// Function to compare employees based on NAMA (ascending order)

bool compareNAMAAscending(const Employee& e1, const Employee& e2) {

return e1.NAMA < e2.NAMA;

}

// Function to compare employees based on NAMA (descending order)

bool compareNAMADescending(const Employee& e1, const Employee& e2) {

return e1.NAMA > e2.NAMA;

}

int main() {

// Initialize employee data

vector<Employee> employees = {

{"12345", "Budi"},

{"67890", "Andi"},

{"45678", "Cici"},

{"23456", "Dedi"},

{"89012", "Eni"},

};

// Choose sorting method, order, and comparison function

int choiceMethod;

int choiceOrder;

bool (\*compareFunction)(const Employee&, const Employee&);

cout << "Pilih metode pengurutan (1 - Insertion Sort, 2 - Selection Sort): ";

cin >> choiceMethod;

cout << "Pilih urutan pengurutan (1 - Ascending, 2 - Descending): ";

cin >> choiceOrder;

cout << "Pilih kriteria pengurutan (1 - NIP, 2 - NAMA): ";

cin >> choiceMethod;

if (choiceMethod == 1) {

compareFunction = compareNIPAscending;

} else if (choiceMethod == 2) {

compareFunction = compareNIPDescending;

} else if (choiceMethod == 3) {

compareFunction = compareNAMAAscending;

} else if (choiceMethod == 4) {

compareFunction = compareNAMADescending;

} else {

cout << "Pilihan tidak valid!" << endl;

return 0;

}

// Sort the employee data using the chosen method, order, and comparison function

if (choiceMethod == 1) {

insertion\_sort(employees.begin(), employees.end(), compareFunction);

} else if (choiceMethod == 2) {

selection\_sort(employees.begin(), employees.end(), compareFunction);

} else {

sort(employees.begin(), employees.end(), compareFunction);

}

// Print the sorted employee data

cout << "\nData Pegawai yang Telah Diurutkan:" << endl;

for (const Employee& employee : employees) {

cout << "NIP: " << employee.NIP << ", NAMA: " << employee.NAMA << endl;

}

return 0;

}

1. Analisa program:
2. Program ini menggunakan struktur `Employee` untuk mewakili catatan karyawan, merangkum atribut NIP dan NAMA mereka. Dan program ini menggunakan algoritma pengurutan (Insertion Sort, Selection Sort, atau fungsi `sort` standar) untuk mengatur data karyawan berdasarkan kriteria yang dipilih.
3. Program ini meminta pengguna untuk memilih metode penyortiran, urutan, dan kriteria perbandingan, memberikan fleksibilitas dalam manipulasi data. Ini memungkinkan skenario dan preferensi penyortiran yang berbeda.
4. Program mendefinisikan fungsi perbandingan untuk membandingkan employee berdasarkan NIP atau NAMA, baik dalam urutan menaik atau menurun.
5. Fungsi-fungsi ini memungkinkan program untuk mengurutkan data berdasarkan kriteria dan urutan yang ditentukan.
6. **Kesimpulan**

Program ini berfungsi sebagai dasar yang kuat untuk memahami dan menerapkan algoritma pengurutan pada tugas pengelolaan data dunia nyata. Ini menunjukkan kemampuan untuk memanipulasi dan mengatur data berdasarkan kriteria tertentu, meningkatkan aksesibilitas data dan kemampuan analisis.