#include <iostream>

#include <cstring>

using namespace std;

struct Pegawai {

string NIP;

string Nama;

string Alamat;

char Golongan;

};

const int MAX\_PEGAWAI = 5;

Pegawai dataPegawai[MAX\_PEGAWAI] = {

{"D003", "Kholid", "Surabaya", 'A'},

{"D005", "Safrodin", "Surabaya", 'A'},

{"D001", "Artiani", "Surabaya", 'A'},

{"D004", "Rizky", "Surabaya", 'A'},

{"D002", "Fahim", "Surabaya", 'A'}

};

void TukarPegawai(Pegawai& a, Pegawai& b) {

Pegawai temp = a;

a = b;

b = temp;

}

void BubbleSortPegawai(bool isAscending, bool sortByNIP) {

int comparisons = 0, swaps = 0;

for (int i = 1; i < MAX\_PEGAWAI; i++) {

for (int j = MAX\_PEGAWAI - 1; j >= i; j--) {

comparisons++;

bool shouldSwap = false;

if (sortByNIP) {

shouldSwap = isAscending ? dataPegawai[j - 1].NIP > dataPegawai[j].NIP

: dataPegawai[j - 1].NIP < dataPegawai[j].NIP;

}

else {

shouldSwap = isAscending ? dataPegawai[j - 1].Nama > dataPegawai[j].Nama

: dataPegawai[j - 1].Nama < dataPegawai[j].Nama;

}

if (shouldSwap) {

TukarPegawai(dataPegawai[j - 1], dataPegawai[j]);

swaps++;

}

}

}

cout << "Jumlah Perbandingan: " << comparisons << endl;

cout << "Jumlah Pergeseran: " << swaps << endl;

}

void ShellSortPegawai(bool isAscending, bool sortByNIP) {

int comparisons = 0, swaps = 0;

int jarak, i, j;

bool sudah;

jarak = MAX\_PEGAWAI;

while (jarak > 1) {

jarak = jarak / 2;

sudah = false;

while (!sudah) {

sudah = true;

for (j = 0; j < MAX\_PEGAWAI - jarak; j++) {

i = j + jarak;

comparisons++;

bool shouldSwap = false;

if (sortByNIP) {

shouldSwap = isAscending ? dataPegawai[j].NIP > dataPegawai[i].NIP

: dataPegawai[j].NIP < dataPegawai[i].NIP;

}

else {

shouldSwap = isAscending ? dataPegawai[j].Nama > dataPegawai[i].Nama

: dataPegawai[j].Nama < dataPegawai[i].Nama;

}

if (shouldSwap) {

TukarPegawai(dataPegawai[j], dataPegawai[i]);

sudah = false;

swaps++;

}

}

}

}

cout << "Jumlah Perbandingan: " << comparisons << endl;

cout << "Jumlah Pergeseran: " << swaps << endl;

}

int main() {

int pilihan, urutanPilihan, kriteriaPilihan;

cout << "Data Pegawai Awal:" << endl;

for (int i = 0; i < MAX\_PEGAWAI; i++) {

cout << "NIP: " << dataPegawai[i].NIP << ", Nama: " << dataPegawai[i].Nama

<< ", Alamat: " << dataPegawai[i].Alamat << ", Golongan: " << dataPegawai[i].Golongan << endl;

}

cout << "\nPilih metode pengurutan:" << endl;

cout << "1. Bubble Sort" << endl;

cout << "2. Shell Sort" << endl;

cout << "Pilihan Anda: ";

cin >> pilihan;

cout << "\nPilih urutan pengurutan:" << endl;

cout << "1. Urut Naik" << endl;

cout << "2. Urut Turun" << endl;

cout << "Pilihan Anda: ";

cin >> urutanPilihan;

cout << "\nPilih kriteria pengurutan:" << endl;

cout << "1. Berdasarkan NIP" << endl;

cout << "2. Berdasarkan Nama" << endl;

cout << "Pilihan Anda: ";

cin >> kriteriaPilihan;

bool isAscending = urutanPilihan == 1;

bool sortByNIP = kriteriaPilihan == 1;

if (pilihan == 1) {

BubbleSortPegawai(isAscending, sortByNIP);

cout << "\nData Pegawai Setelah Diurutkan dengan Bubble Sort:" << endl;

}

else if (pilihan == 2) {

ShellSortPegawai(isAscending, sortByNIP);

cout << "\nData Pegawai Setelah Diurutkan dengan Shell Sort:" << endl;

}

else {

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

return 0;

}

for (int i = 0; i < MAX\_PEGAWAI; i++) {

cout << "NIP: " << dataPegawai[i].NIP << ", Nama: " << dataPegawai[i].Nama

<< ", Alamat: " << dataPegawai[i].Alamat << ", Golongan: " << dataPegawai[i].Golongan << endl;

}

return 0;

}

Kesimpulan pada percobaan kali ini adalah **Shell Sort** lebih disukai untuk data yang besar dan membutuhkan kecepatan pengurutan yang lebih tinggi, sedangkan Bubble sort lebih cocok untuk data yang kecil dan membutuhkan stabilitas.