**Kode Percobaan & Latihan dalam Bentuk WORD**

Praktikum Algoritma dan Struktur Data

****

**Oleh:**

Samudero Dirgantoro / 5223600016

**Program Studi D4 Teknologi Game**

**Departemen Teknologi Multimedia Kreatif**

**Politeknik Elektronika Negeri Surabaya**

**2023/2024**

1. Percobaan:

//Samudero Dirgantoro 5223600016

#include <iostream>

#include <cstdlib>

#include <ctime>

#define MAX 10

int Data[MAX];

//pertukaran bilangan

void Tukar(int\* a, int\* b) {

int temp = \*a;

\*a = \*b;

\*b = temp;

}

//pengurutan langsung

void langsung() {

int i, j, x;

for (i = 1; i < MAX; i++) {

x = Data[i];

j = i - 1;

while (j >= 0 && x < Data[j]) {

Data[j + 1] = Data[j];

j--;

}

Data[j + 1] = x;

}

}

//pengurutan biner

void biner() {

int i, j, l, r, m, x;

for (i = 1; i < MAX; i++) {

x = Data[i];

l = 0;

r = i - 1;

while (l <= r) {

m = (l + r) / 2;

if (x < Data[m])

r = m - 1;

else

l = m + 1;

}

for (j = i - 1; j >= l; j--)

Data[j + 1] = Data[j];

Data[l] = x;

}

}

//pengurutan seleksi

void seleksi() {

int i, j, k;

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

k = i;

for (j = i + 1; j < MAX; j++) {

if (Data[k] > Data[j])

k = j;

}

Tukar(&Data[i], &Data[k]);

}

}

//mencetak data

void TampilkanData(const char\* label) {

std::cout << label << "\n";

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

std::cout << "Data ke " << i << " : " << Data[i] << "\n";

}

}

int main() {

int pilihan;

//Generate bilangan acak

srand(time(0));

std::cout << "Membangkitkan bilangan acak\n";

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

Data[i] = rand() % 1000 + 1;

}

// Meminta pengguna memilih metode pengurutan

std::cout << "Pilih metode pengurutan:\n";

std::cout << "1. Penyisipan Langsung\n";

std::cout << "2. Penyisipan Biner\n";

std::cout << "3. Metode Seleksi\n";

std::cout << "Masukkan pilihan Anda: \n";

std::cin >> pilihan;

//Menu

switch (pilihan) {

case 1:

langsung();

TampilkanData("Penyisipan Langsung:");

break;

case 2:

biner();

TampilkanData("Penyisipan Biner:");

break;

case 3:

seleksi();

TampilkanData("Metode Seleksi:");

break;

default:

std::cout << "error\n";

break;

}

}

1. Latihan:

#include <iostream>

#include <string>

using namespace std;

struct Pegawai {

int NIP;

string Nama;

string Alamat;

char Golongan;

};

// Fungsi menampilkan data pegawai

void tampilkanPegawai(Pegawai arr[], int n) {

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

cout << "NIP: " << arr[i].NIP << ", Nama: " << arr[i].Nama << ", Alamat: " << arr[i].Alamat << ", Golongan: " << arr[i].Golongan << "\n";

}

cout << "\n";

}

// Fungsi langsung (straight insertion) berdasarkan NIP

void langsungNIP(Pegawai arr[], int n, bool urutanNaik) {

int i, j, perbandingan = 0, pergeseran = 0;

Pegawai key;

for (i = 1; i < n; i++) {

key = arr[i];

j = i - 1;

if (urutanNaik) {

while (j >= 0 && arr[j].NIP > key.NIP) {

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

j = j - 1;

perbandingan++;

pergeseran++;

}

} else {

while (j >= 0 && arr[j].NIP < key.NIP) {

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

j = j - 1;

perbandingan++;

pergeseran++;

}

}

arr[j + 1] = key;

pergeseran++;

}

cout << "Pengurutan berdasarkan NIP\n";

cout << "Perbandingan: " << perbandingan << ", Pergeseran: " << pergeseran << "\n";

tampilkanPegawai(arr, n);

}

// Fungsi langsung (straight insertion) berdasarkan Nama

void langsungNAMA(Pegawai arr[], int n, bool urutanNaik) {

int i, j, perbandingan = 0, pergeseran = 0;

Pegawai key;

for (i = 1; i < n; i++) {

key = arr[i];

j = i - 1;

if (urutanNaik) {

while (j >= 0 && arr[j].Nama > key.Nama) {

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

j = j - 1;

perbandingan++;

pergeseran++;

}

} else {

while (j >= 0 && arr[j].Nama < key.Nama) {

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

j = j - 1;

perbandingan++;

pergeseran++;

}

}

arr[j + 1] = key;

pergeseran++;

}

cout << "Pengurutan berdasarkan Nama\n";

cout << "Perbandingan: " << perbandingan << ", Pergeseran: " << pergeseran << "\n";

tampilkanPegawai(arr, n);

}

// Fungsi biner (binary insertion) berdasarkan NIP

void binerNIP(Pegawai arr[], int n, bool urutanNaik) {

int i, j, perbandingan = 0, pergeseran = 0;

Pegawai key;

int low, high, mid;

for (i = 1; i < n; i++) {

key = arr[i];

low = 0;

high = i - 1;

while (low <= high) {

mid = (low + high) / 2;

if (urutanNaik) {

if (key.NIP < arr[mid].NIP) {

high = mid - 1;

} else {

low = mid + 1;

}

} else {

if (key.NIP > arr[mid].NIP) {

high = mid - 1;

} else {

low = mid + 1;

}

}

perbandingan++;

}

for (j = i - 1; j >= low; j--) {

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

pergeseran++;

}

arr[low] = key;

pergeseran++;

}

cout << "Pengurutan berdasarkan NIP\n";

cout << "Perbandingan: " << perbandingan << ", Pergeseran: " << pergeseran << "\n";

tampilkanPegawai(arr, n);

}

// Fungsi biner (binary insertion) berdasarkan Nama

void binerNAMA(Pegawai arr[], int n, bool urutanNaik) {

int i, j, perbandingan = 0, pergeseran = 0;

Pegawai key;

int low, high, mid;

for (i = 1; i < n; i++) {

key = arr[i];

low = 0;

high = i - 1;

while (low <= high) {

mid = (low + high) / 2;

if (urutanNaik) {

if (key.Nama < arr[mid].Nama) {

high = mid - 1;

} else {

low = mid + 1;

}

} else {

if (key.Nama > arr[mid].Nama) {

high = mid - 1;

} else {

low = mid + 1;

}

}

perbandingan++;

}

for (j = i - 1; j >= low; j--) {

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

pergeseran++;

}

arr[low] = key;

pergeseran++;

}

cout << "Pengurutan berdasarkan Nama\n";

cout << "Perbandingan: " << perbandingan << ", Pergeseran: " << pergeseran << "\n";

tampilkanPegawai(arr, n);

}

// Fungsi seleksi berdasarkan NIP

void seleksiNIP(Pegawai arr[], int n, bool urutanNaik) {

int i, j, minIndex, perbandingan = 0, pergeseran = 0;

Pegawai temp;

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

minIndex = i;

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

if (urutanNaik) {

if (arr[j].NIP < arr[minIndex].NIP) {

minIndex = j;

}

} else {

if (arr[j].NIP > arr[minIndex].NIP) {

minIndex = j;

}

}

perbandingan++;

}

temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

pergeseran++;

}

cout << "Pengurutan berdasarkan NIP\n";

cout << "Perbandingan: " << perbandingan << ", Pergeseran: " << pergeseran << "\n";

tampilkanPegawai(arr, n);

}

// Fungsi seleksi berdasarkan Nama

void seleksiNAMA(Pegawai arr[], int n, bool urutanNaik) {

int i, j, minIndex, perbandingan = 0, pergeseran = 0;

Pegawai temp;

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

minIndex = i;

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

if (urutanNaik) {

if (arr[j].Nama < arr[minIndex].Nama) {

minIndex = j;

}

} else {

if (arr[j].Nama > arr[minIndex].Nama) {

minIndex = j;

}

}

perbandingan++;

}

temp = arr[minIndex];

arr[minIndex] = arr[i];

arr[i] = temp;

pergeseran++;

}

cout << "Pengurutan berdasarkan Nama\n";

cout << "Perbandingan: " << perbandingan << ", Pergeseran: " << pergeseran << "\n";

tampilkanPegawai(arr, n);

}

int main() {

// Contoh data pegawai

Pegawai pegawai[] = {

{101, "Dimas", "Jakarta", 'A'},

{103, "Anwar", "Bandung", 'B'},

{102, "Budi", "Surabaya", 'C'},

{104, "Jojo", "Yogyakarta", 'A'}

};

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

// Pilihan metode pengurutan

int metode;

cout << "Pilih metode pengurutan:\n";

cout << "1. Straight Insertion\n";

cout << "2. Binary Insertion\n";

cout << "3. Selection Sort\nInput:";

cin >> metode;

// Pilihan urutan

int urutan;

cout << "Pilih urutan:\n";

cout << "1. Naik\n";

cout << "2. Turun\nInput: ";

cin >> urutan;

// Pilihan pengurutan berdasarkan

int berdasarkan;

cout << "Pilih pengurutan berdasarkan:\n";

cout << "1. NIP\n";

cout << "2. Nama\nInput: ";

cin >> berdasarkan;

// Mengurutkan data berdasarkan pilihan

switch (berdasarkan) {

case 1:

switch (metode) {

case 1:

if (urutan == 1) {

langsungNIP(pegawai, n, true);

} else {

langsungNIP(pegawai, n, false);

}

break;

case 2:

if (urutan == 1) {

binerNIP(pegawai, n, true);

} else {

binerNIP(pegawai, n, false);

}

break;

case 3:

if (urutan == 1) {

seleksiNIP(pegawai, n, true);

} else {

seleksiNIP(pegawai, n, false);

}

break;

default:

cout << "Pilihan tidak valid.";

}

break;

case 2:

switch (metode) {

case 1:

if (urutan == 1) {

langsungNAMA(pegawai, n, true);

} else {

langsungNAMA(pegawai, n, false);

}

break;

case 2:

if (urutan == 1) {

binerNAMA(pegawai, n, true);

} else {

binerNAMA(pegawai, n, false);

}

break;

case 3:

if (urutan == 1) {

seleksiNAMA(pegawai, n, true);

} else {

seleksiNAMA(pegawai, n, false);

}

break;

default:

cout << "Pilihan tidak valid.";

}

break;

default:

cout << "Pilihan tidak valid.";

}

}