**Tugas 2**

**Analisis Algoritma**



Disusun oleh :

Baby Cattleya Gustina Permatagama

(140810160048)

S-1 Teknik Informatika   
Fakultas Matematika & Ilmu Pengetahuan Alam   
Universitas Padjadjaran  
Jalan Raya Bandung - Sumedang Km. 21 Jatinangor 45363

1. Program *Searching Linear*

/\*

Nama Program : Program Menghitung Pangkat dengan looping while

Oleh : Baby Cattleya

Dibuat : Minggu, 24 Maret 2019

\*/

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <chrono>

using namespace std;

int linearSearch(const int [], int, int);

const int SIZE = 100;

int main() {

int array[SIZE];

unsigned seed = time(0);

srand(seed);

cout << "Array Angka Random: " <<endl;

for(int i = 0; i<100;i++)

{

array[i]=rand()%100+1;

cout << array[i]<<" ";

}

int results;

int input;

cout << endl << endl <<"Masukkan Angka yang Dicari: ";

cin >> input;

auto start = chrono::steady\_clock::now();

results = linearSearch(array, SIZE, input);

if (results == -1){

cout << "Angka Tidak Ditemukan di Array\n";

}

else {

cout << "Angka Ditemukan di indeks ke " << results;

cout << " pada array.\n" <<endl;

}

auto end = chrono::steady\_clock::now();

auto diff = end - start;

cout << "Running Time: "<<chrono::duration <double, milli> (diff).count() << " ms" << endl;

return 0;

}

int linearSearch(const int array[], int size, int value){

int lokasi = 0;

bool found = false;

while (!found && lokasi < size) {

if (array[lokasi] == value){

found = true;

}

else {

lokasi=lokasi+1;

}

}

return lokasi;

}

1. Buat Program *Binary Search*

/\*

Nama Program : Program Binary Seacrh

Oleh : Baby Cattleya

Dibuat : Minggu, 24 Maret 2019

\*/

#include <iostream>

#include <chrono>

using namespace std;

main () {

int n, i, search, first, last, middle;

const int SIZE = 100;

int arr[SIZE];

unsigned seed = time(0);

srand(seed);

cout << "Array Angka Random: " <<endl;

for(int i = 0; i<100;i++) //to get random numbers inside the array.

{

arr[i]=rand()%100+1;

cout << arr[i]<<" ";

}

cout<<endl<<"Masukkan angka yang akan dicari :";

cin>>search;

auto start = chrono::steady\_clock::now();

int posisi;

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

posisi=i;

for (int j=i+1;j<SIZE;j++) {

if (arr[posisi]>arr[j]) {

posisi=j;

}

}

swap(arr[i], arr[posisi]);

}

cout << endl << "Array Angka Sorted: " <<endl;

for(int i = 0; i<100;i++) //to get random numbers inside the array.

{

cout << arr[i]<<" ";

}

first = 0;

last = SIZE-1;

middle = (first+last)/2;

while (first <= last)

{

if(arr[middle] < search)

{

first = middle + 1;

}

else if(arr[middle] == search)

{

cout<< endl <<"Angka "<< search<<" ditemukan"<<endl;

break;

}

else

{

last = middle - 1;

}

middle = (first + last)/2;

}

if(first > last)

{

cout<<endl<<"Error! "<<search<<" tidak ditemukan dalam Array" <<endl;

}

auto end = chrono::steady\_clock::now();

auto diff = end - start;

cout << "waktu program :" <<chrono::duration <double, milli> (diff).count() << " ms" << endl;

}

1. Program Maksimum Minimum Sort
2. Quick Sort (pake pointer linked list)
3. Analisis kompleksitas dan perbandingan waktu running time tiap metode dan jumlah data

Catatan :

masing masing dicoba diberi data minimal 100 dan maksimum 1000 elemen array

Running program

Hitung waktu running, tentukan kompleksitas

Bikin Program