TUGAS 1 KELAS

RUNNING TIME

Disusun sebagai salah satu tugas

mata kuliah Analisis Algoritma



Shofiyyah Nadhiroh – 14018160057

Patricia Joanne – 140810160065

Dikumpulkan tanggal

6 Maret 2019

PROGRAM STUDI S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2019

**Soal**

1. Algoritma dan program memangkatkan bilangan 2 pangkat n (integer) menggunakan algoritma iteratif dan rekursif.
2. Algoritma dan program pencarian linear dan biner.
3. Analisis running timenya dengan mengganti data/spesifikasi hardwarenya.

**Jawaban**

1. 2n dengan algoritma iteratif

Algoritma:

Deklarasikan a sebagai angka 2 dan b sebagai pangkat n.

Hasil = 1, ulangi hasil \* a sebanyak b kali.

Program: <http://cpp.sh/7sctq>

#include<iostream>

#include<chrono>

using namespace std;

using namespace std::chrono;

int pangkat (int a, int b){

int hasil=1;

for (int i=1;i<=b;i++){

hasil=hasil\*a;

}

return hasil;

}

int main(){

high\_resolution\_clock::time\_point t1 = high\_resolution\_clock ::now();

//masukkan input manual

int a=2, b=30;

cout<<"Menghitung 2 Pangkat n Iteratif"<<endl;

cout<<endl<<"Pangkat: "<<b;

cout<<endl<<"Hasil: ";

cout<<pangkat(a,b)<<endl;

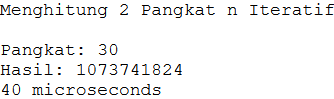
high\_resolution\_clock::time\_point t2 = high\_resolution\_clock ::now();

auto duration = duration\_cast<microseconds>( t2 - t1 ) .count();

cout<<duration <<" microseconds"<<endl;

}

Output:



1. 2n dengan algoritma rekursif

Algoritma:

Deklarasikan a sebagai angka 2 dan b sebagai pangkat n.

Hasil = 1 jika b = 0, selain itu ulangi a \* fungsi pangkat.

Program: <http://cpp.sh/3phul>

#include <iostream>

#include <cstdlib>

#include <chrono>

using namespace std;

using namespace std::chrono;

int pangkat(int a, int b){

if (b==0){

return (1);

}

else {

return (a\*pangkat (a, b-1));

}

}

int main(int argc, char \*argv[]) {

high\_resolution\_clock::time\_point t1 = high\_resolution\_clock ::now();

//masukkan input manual

int a=2, b=30;

cout<<"Menghitung 2 Pangkat n Rekursif"<<endl;

cout<<endl<<"Pangkat: "<<b;

cout<<endl<<"Hasil: ";

cout<<pangkat(a,b)<<endl;

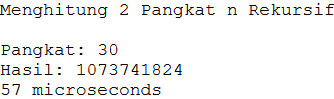
high\_resolution\_clock::time\_point t2 = high\_resolution\_clock ::now();

auto duration = duration\_cast<microseconds>( t2 - t1 ) .count();

cout<<duration <<" microseconds"<<endl;

}

Output:



1. Pencarian linear

Algoritma:

Deklarasikan isi array, jumlah isi array, dan angka yang ingin dicari.

Telusuri isi array dengan mengecek satu persatu urut arraynya hingga urut terakhir untuk menemukan angka yang ingin dicari.

Jika angka yang ingin dicari = isi array tersebut, maka jawabnya ada.

Program: <http://cpp.sh/2xzk3>

#include <iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

int main(){

high\_resolution\_clock::time\_point t1 = high\_resolution\_clock ::now();

int arr[] = { 21, 12, 34, 56, 67, 80, 3, 100, 96, 49 };

int size = sizeof(arr) / sizeof(arr[0]); //array 10 angka

int n = 50; //masukkan input manual

cout<<"Melakukan pencarian linear"<<endl;

cout<<"Apakah angka "<<n<<" ada?"<<endl;

bool ketemu = false;

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

if (n == arr[i])

ketemu = true;

}

cout<<"Jawab: "<<(ketemu == true ? "Ada!":"Tidak ada!")<<endl;

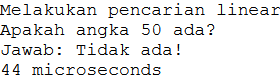
high\_resolution\_clock::time\_point t2 = high\_resolution\_clock ::now();

auto duration = duration\_cast<microseconds>( t2 - t1 ) .count();

cout<<duration <<" microseconds" <<endl;

}

Output:



1. Pencarian biner

Algoritma:

Deklarasikan isi array, jumlah isi array, dan angka yang ingin dicari.

Telusuri isi array dengan mengecek dari urut tengah ke urut pertama dan urut terakhir hingga ditemukan angka yang ingin dicari.

Jika angka yang ingin dicari = isi array tersebut, maka jawabnya ada.

Program: <http://cpp.sh/9qbc>

#include <iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

int main(){

high\_resolution\_clock::time\_point t1 = high\_resolution\_clock ::now();

int arr[] = { 21, 12, 34, 56, 67, 80, 3, 100, 96, 49 };

int size = sizeof(arr) / sizeof(arr[0]); //array 10 angka

int n = 50; //masukkan input manual

cout<<"Melakukan pencarian biner"<<endl;

cout<<"Apakah angka "<<n<<" ada?"<<endl;

bool ketemu = false;

int left = 0, right = size-1;

while (left <= right){

int mid = left + (right - left) / 2;

if (arr[mid] == n)

ketemu = true;

if (arr[mid] < n)

left = mid + 1;

else

right = mid - 1;

}

cout<<"Jawab: "<<(ketemu == true ? "Ada!":"Tidak ada!")<<endl;

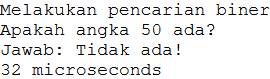
high\_resolution\_clock::time\_point t2 = high\_resolution\_clock ::now();

auto duration = duration\_cast<microseconds>( t2 - t1 ) .count();

cout<<duration <<" microseconds" <<endl;

}

Output:



**Analisis**

Running time dalam program-program di atas didapatkan dari library chrono pada C++. Running time selalu berubah-ubah bahkan untuk program dengan code dan data yang sama sekalipun. Running time dipengaruhi oleh kerja CPU dan kita tidak pernah tahu apa saja proses yang sedang dikerjakan oleh CPU kita. Untuk spesifikasi dari PC yang digunakan untuk menjalankan program-program di atas adalah sebagai berikut:

