**LAPORAN TUGAS 1**

**Analisis Algoritma**

Algoritma Pemangkatan, Linear Search, Binary Search

****

Fajar Adiyansyah Rahiq 140810160006

Fikri Ikhsan 140810160016

Muhammad Jordiyansyah 140810160040

KELAS B

S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2018/2019

**Tugas**

* Tulis algoritma dan program memangkatkan bilangan 2 pangkat n (integer) menggunakan algoritma iteratif dan rekursif
* Buat algortima dan program pencarian linear dan pencarian biner
* Analisis “running time” nya program-program tersebut dengan mencoba data yang berbeda dan bandingkan satu sama lain, tuliskan spesifikasi komputer yang digunakan, tuliskan kesimpulannya !!

1. Algoritma Pemangkatan

**Algoritma iteratif :** perulangan yang melakukan proses perulangan terhadap sekelompok intruksi. Perulangan dilakukan dalam batasan syarat tertentu. Ketika syarat tersebut tidak terpenuhi lagi maka perulangan akan terhenti.

**Algoritma rekursif :** salah satu metode didalam pemrograman yang mana dalam sebuah fungsi terdapat intruksi yang memanggil fungsi itu sendri, atau lebih sering disebut memanggil dirinya sendiri.

**Perbedaan :**

* Iteratif menggunakan FOR, WHILE, DO-WHILE sedangkan rekursif hanya menggunakan IF.
* Iteratif dapat berjalan pada program yang terdiri dari prosedur (tidak terdapat fungsi) sedangkan rekursif merupakan fungsi.

Algoritma Pemangkatan metode Iteratif

#include<iostream>

#include <time.h>

using namespace std;

int pangkat (int n)

{int hasil=1;

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

{

hasil=hasil\*2;

}

return hasil;

}

int main()

{

clock\_t start, end;

float time;

int j;

cout<<"PROGRAM MENGHITUNG PANGKAT"<<endl;

cout<<endl<<"Masukkan Pangkat : ";

cin>>j;

start = clock();

cout<<endl<<"Hasilnya = ";

cout<<pangkat(j)<<endl;

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s"<<endl;

}

int pangkat (int n)

{int hasil=1;

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

{

hasil=hasil\*2;

}

return hasil;

}

int main()

{

clock\_t start, end;

float time;

int j;

cout<<"PROGRAM MENGHITUNG PANGKAT"<<endl;

cout<<endl<<"Masukkan Pangkat : ";

cin>>j;

start = clock();

cout<<endl<<"Hasilnya = ";

cout<<pangkat(j)<<endl;

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s"<<endl;

}

Algoritma Pemangkatan metode Rekursif

#include <iostream>

#include <time.h>

using namespace std;

int Pangkat(int y){

if (y==0){

return 1;

}

else{

return (2\*Pangkat(y-1));

}

}

int main() {

clock\_t start, end;

float time;

int p;

cout<<"PROGRAM MENGHITUNG PANGKAT"<<endl;

cout<<"\nMasukkan pangkat : ";

cin>>p;

cout<<endl;

start = clock();

cout<<"Hasil = "<<Pangkat(p)<<endl;

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s"<<endl;

}

**Perbandingan Pemangkatan Iteratif dan Rekursif :**

cout<<"\nMasukkan pangkat : ";

cin>>p;

cout<<endl;

start = clock();

cout<<"Hasil = "<<Pangkat(p)<<endl;

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s"<<endl;

}

1. Algoritma Pencarian Linear dan Biner

Linear/Sequential Search

//Menyertakan library

#include <iostream>

#include <time.h>

//Menentukan banyak data

#define MAX 500000

using namespace std;

int main()

{

clock\_t start, end;

float time;

//Deklarasi variabel

int item;

int nomor[MAX];

bool found = false;

//Input Data Nomor

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

nomor[i] = i+1;

}

//Input key untuk pencarian

cout<<"Masukkan Nomor Pencarian : ";

cin >> item;

start = clock();

//Lakukan iterasi sebanyak jumlah data yang ada

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

{

//Logika ketika key sama dengan data pada array pada indeks i, maka data ditemukan

if(item==nomor[i])

{

//Tampilan ketika data ditemukan

found = true;

cout << "Data ditemukan pada lokasi " << i+1 << endl;

break;

}

}

//Logika jika data tidak ditemukan

if(found=false)

{

//Tampilan ketika data tidak ditemukan

cout<<"Data tidak ditemukan";

}

//Akhir clock time setelah proses

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s";

}

nomor[i] = i+1;

}

//Input key untuk pencarian

cout<<"Masukkan Nomor Pencarian : ";

cin >> item;

start = clock();

//Lakukan iterasi sebanyak jumlah data yang ada

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

{

//Logika ketika key sama dengan data pada array pada indeks i, maka data ditemukan

if(item==nomor[i])

{

//Tampilan ketika data ditemukan

found = true;

cout << "Data ditemukan pada lokasi " << i+1 << endl;

break;

}

}

//Logika jika data tidak ditemukan

if(found=false)

{

//Tampilan ketika data tidak ditemukan

cout<<"Data tidak ditemukan";

}

//Akhir clock time setelah proses

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s";

}

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s";

}

Binary Search

//Menyertakan library yang dibutuhkan

#include <iostream>

#include <time.h>

//Menentukan banyak data

#define MAX 500000

//namespace

using namespace std;

int main()

{

//Deklarasi clock

clock\_t start, end;

double time;

//Deklarasi variabel

int nomor[MAX];

int key, low = 0, high = MAX;

bool found = false;

//Input Data Nomor

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

nomor[i] = i+1;

}

//Input key untuk pencarian

cout<<"Masukkan Nomor Pencarian : ";

cin>>key;

//Start clock time setelah input

start = clock();

while(low<=high)

{

//Deklarasi nilai tengah

int mid = (low+high)/2;

//Logika jika key sama dengan data pada array pada indeks i, maka data ditemukan

if(key == nomor[mid])

{

//Tampilan ketika data ditemukan

cout<<"Data ditemukan pada lokasi "<<mid+1<<endl;

found = true;

break;

}

//jika key < data pada array maka high = mid - 1

else if(key<nomor[mid])

{

high=mid - 1;

}

//jika sebaliknya key > data pada array maka low = mid + 1

else

{

low = mid + 1;

}

if(found=false)

{

//Tampilan ketika data tidak ditemukan

cout<<"Data tidak ditemukan";

}

}

//Akhir clock time setelah proses

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s";

}

//Deklarasi variabel

int nomor[MAX];

int key, low = 0, high = MAX;

bool found = false;

//Input Data Nomor

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

nomor[i] = i+1;

}

//Input key untuk pencarian

cout<<"Masukkan Nomor Pencarian : ";

cin>>key;

//Start clock time setelah input

start = clock();

while(low<=high)

{

//Deklarasi nilai tengah

int mid = (low+high)/2;

//Logika jika key sama dengan data pada array pada indeks i, maka data ditemukan

if(key == nomor[mid])

{

//Tampilan ketika data ditemukan

cout<<"Data ditemukan pada lokasi "<<mid+1<<endl;

found = true;

break;

}

//jika key < data pada array maka high = mid - 1

else if(key<nomor[mid])

{

high=mid - 1;

}

//jika sebaliknya key > data pada array maka low = mid + 1

else

{

low = mid + 1;

}

if(found=false)

{

//Tampilan ketika data tidak ditemukan

cout<<"Data tidak ditemukan";

}

}

//Akhir clock time setelah proses

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s";

}

//jika key < data pada array maka high = mid - 1

else if(key<nomor[mid])

{

high=mid - 1;

}

//jika sebaliknya key > data pada array maka low = mid + 1

else

{

low = mid + 1;

}

if(found=false)

{

//Tampilan ketika data tidak ditemukan

cout<<"Data tidak ditemukan";

}

}

//Akhir clock time setelah proses

end = clock();

//Perhitungan waktu dari clock time yang ada

time = (float) (end - start)\*1000000000000000.0;

cout<<"Process Time : "<<time<<" s";

}