

TUGAS ANALISIS ALGORITMA

ANALISIS KOMPLEKSITAS WAKTU



Disusun Oleh:
Muhammad Ahsan Nurrijal
140810160004

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN

2019

Program

Binary Search

```
#include<iostream>

#include <chrono>

using namespace std;

using namespace std::chrono;

int main()
{
    int n, i, arr[50], search, first, last, middle;

    cout<<"Enter total number of elements :";

    cin>>n;

    cout<<"Enter "<<n<<" number :";

    for (i=0; i<n; i++)
    {
        cin>>arr[i];
    }

    cout<<"Enter a number to find :";

    cin>>search;

    auto start = high_resolution_clock::now();

    first = 0;

    last = n-1;

    middle = (first+last)/2;
```

```

while (first <= last)
{
    if(arr[middle] < search)
    {
        first = middle + 1;

    }
    else if(arr[middle] == search)
    {
        cout<<search<<" found at location "<<middle+1<<"\n";
        break;
    }
    else
    {
        last = middle - 1;
    }
    middle = (first + last)/2;
}

if(first > last)
{
    cout<<"Not found! "<<search<<" is not present in the list.";
}

auto stop = high_resolution_clock::now();

auto duration = duration_cast<nanoseconds>(stop - start);

```

```

cout << "Time taken by function: "

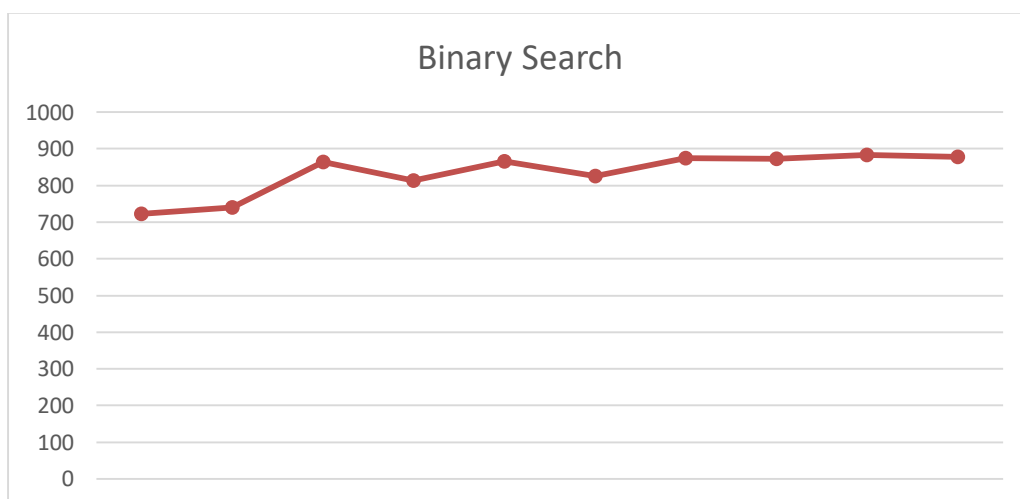
    << duration.count() << " nanoseconds" << endl;

return 0;

}

```

Binary Search	
Percobaan	Elapsed Time (ns)
1	723
2	740
3	864
4	813
5	865
6	825
7	875
8	873
9	883
10	877



Linear Search

```
#include<iostream>
```

```
#include <chrono>
```

```
using namespace std;
```

```
using namespace std::chrono;
```

```
int main()
```

```
{
```

```
    int arr[10], i, num, n, c=0, pos;
```

```
    cout<<"Enter the array size : ";
```

```
    cin>>n;
```

```
    cout<<"Enter Array Elements : ";
```

```
    for(i=0; i<n; i++)
```

```
    {
```

```
        cin>>arr[i];
```

```
    }
```

```
    auto start = high_resolution_clock::now();
```

```
    cout<<"Enter the number to be search : ";
```

```
    cin>>num;
```

```
    for(i=0; i<n; i++)
```

```
    {
```

```
        if(arr[i]==num)
```

```

        {

            c=1;

            pos=i+1;

            break;

        }

    }

    if(c==0)

    {

        cout<<"Number not found..!!";

    }

    else

    {

        cout<<num<<" found at position "<<pos;

    }

    auto stop = high_resolution_clock::now();

    auto duration = duration_cast<nanoseconds>(stop - start);

    cout << "Time taken by function: "

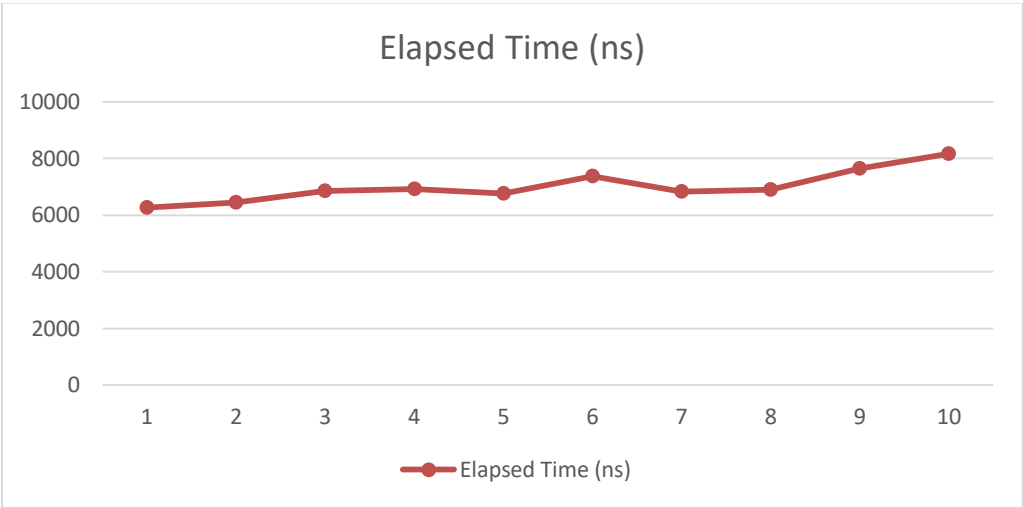
        << duration.count() << " nanoseconds" << endl;

    return 0;

}

```

Linear Search	
Percobaan	Elapsed Time (ns)
1	6251
2	6453
3	6848
4	6908
5	6757
6	7360
7	6837
8	6899
9	7636
10	8158



Pangkat Iterasi

```
#include <iostream>
```

```
#include <chrono>
```

```
using namespace std;
```

```
using namespace std::chrono;
```

```
main(){
```

```
    int pangkat, hasil, n;
```

```
    cout<<"Masukan nilai utama : "<<endl;
```

```
    cin>>n;
```

```
    cout<<"Masukan pangkat : "<<endl;
```

```
    cin>>pangkat;
```

```
    hasil = n;
```

```
    auto start = high_resolution_clock::now();
```

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

```
        hasil*=n;
```

```
    }
```

```
    cout<<"\nHasil = "<< hasil<<endl;
```



```

    auto stop = high_resolution_clock::now();

    auto duration = duration_cast<nanoseconds>(stop - start);

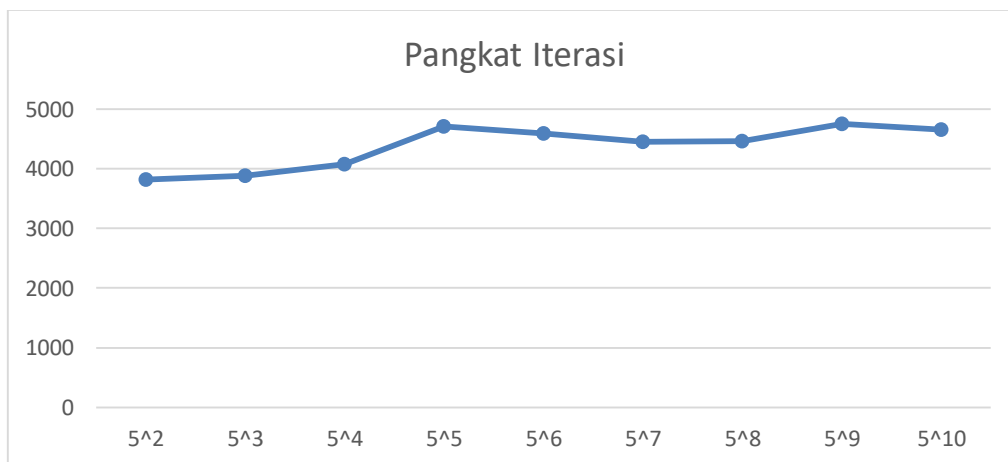
    cout << "Time taken by function: "

    << duration.count() << " nanoseconds" << endl;

}

```

Pangkat Iterasi	
Input	Elapsed Time (ns)
5 ²	3819
5 ³	3882
5 ⁴	4079
5 ⁵	4706
5 ⁶	4592
5 ⁷	4447
5 ⁸	4460
5 ⁹	4749
5 ¹⁰	4653



Pangkat Rekursif

```
#include <iostream>
```

```
#include <chrono>
```

```
using namespace std;
```

```
using namespace std::chrono;
```

```
int Pangkat(int x,int y){
```

```
    if (y==0){
```

```
        return 1;
```

```
    }
```

```
    else{
```

```
        return (x*Pangkat(x,y-1));
```

```
    }
```

```
}
```

```
int main(int argc, char** argv) {
```

```
    int b,p;
```

```
    cout<<"masukkan bilangan : "<<endl;
```

```
    cin>>b;
```

```
    cout<<"masukkan pangkat : "<<endl;
```

```
    cin>>p;
```

```
    cout<<endl;
```

```
    auto start = high_resolution_clock::now();
```

```

cout<<b<<"^"<<p<<"="<<Pangkat(b,p)<<endl;

auto stop = high_resolution_clock::now();

    auto duration = duration_cast<nanoseconds>(stop - start);

cout << "Time taken by function: "

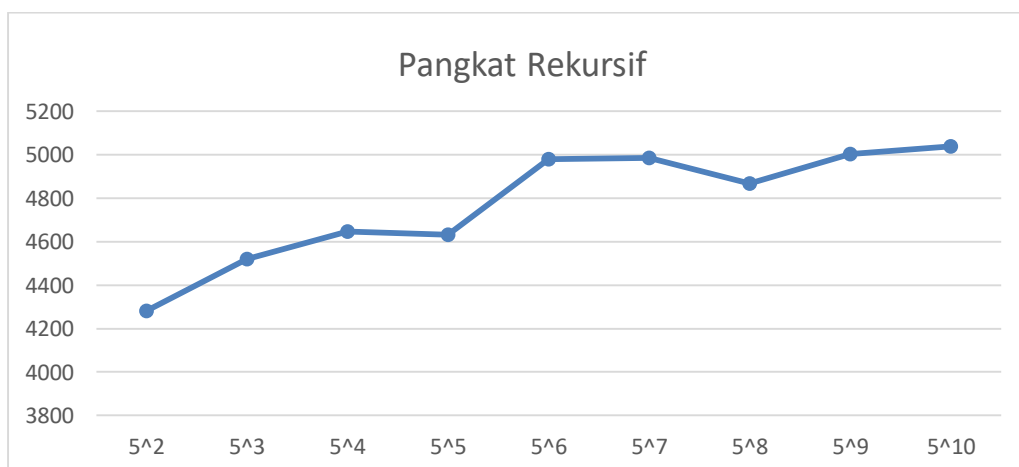
    << duration.count() << " nanoseconds" << endl;

return 0;

}

```

Pangkat Rekursif	
Input	Elapsed Time (ns)
5 ²	4281
5 ³	4519
5 ⁴	4647
5 ⁵	4631
5 ⁶	4978
5 ⁷	4985
5 ⁸	4867
5 ⁹	5003
5 ¹⁰	5037



Analisis

Dengan spesifikasi laptop 8GB RAM, Ryzen 5 dan Win 10 untuk menjalankan program-program diatas, terdapat grafik yang menunjukkan perbedaan waktu untuk menjalankan program tersebut dengan berbagai input data yang dimasukkan. Terlihat naik turun atau terus menanjak menunjukkan bahwa adanya perbedaan waktu dalam menjalankan program. Contohnya seperti hasil perpangkatan. Banyak program pencarian perpangkatan, tidak hanya satu saja. Namun, manakah program perpangkatan yang lebih efisien dalam menghasilkan outputnya. Jika dilihat dalam pencarian diatas, program pencarian pangkat melalui iterasi lebih stabil dan terlihat lebih cepat dan efisien ketimbang program pencarian pangkat rekursif.

Untuk yang kedua adalah program pencarian elemen menggunakan Binary Search atau Linear Search. Dapat dilihat bahwa terjadi ketimpangan waktu diantara keduanya dimana binary search terlihat lebih efisien dan lebih cepat dalam masalah waktu ketimbang program pencarian elemen menggunakan Linear Search.