**LAPORAN PRAKTIKUM 2**

**ANALISIS ALGORITMA**

****

Disusun oleh :

Hafidh Akhdan Najib

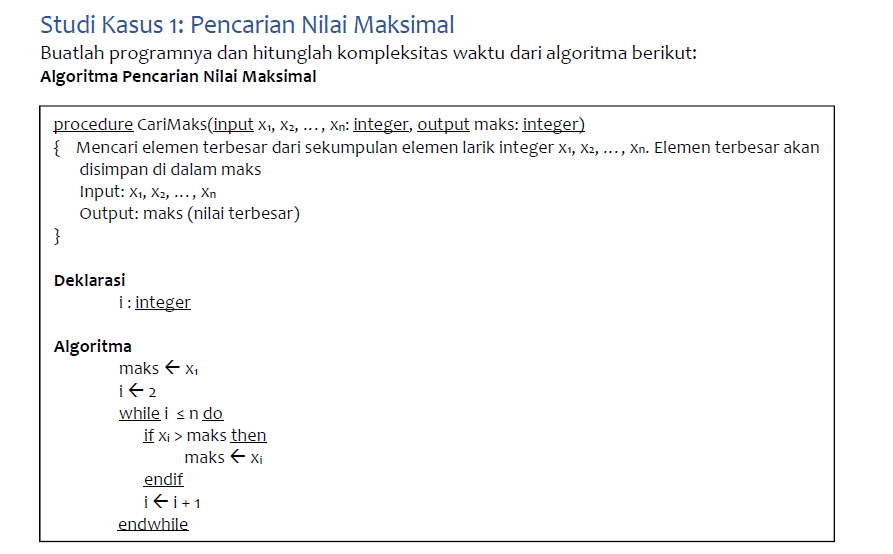
140810180061

**PROGRAM STUDI S-1 INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**2020**



Jawab :  
  
/\*

Nama : Hafidh Akhdan N

NPM : 140810180061

Kelas : A

Program : Studi Kasus 1 "Pencarian nilai maksimal"

\*/

#include<iostream>

using namespace std;

int main(){

int x[5]={31,45,100,4,35};

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

int maks = x[0];

int i= 2;

while (i<= n){

if(x[i] > maks){

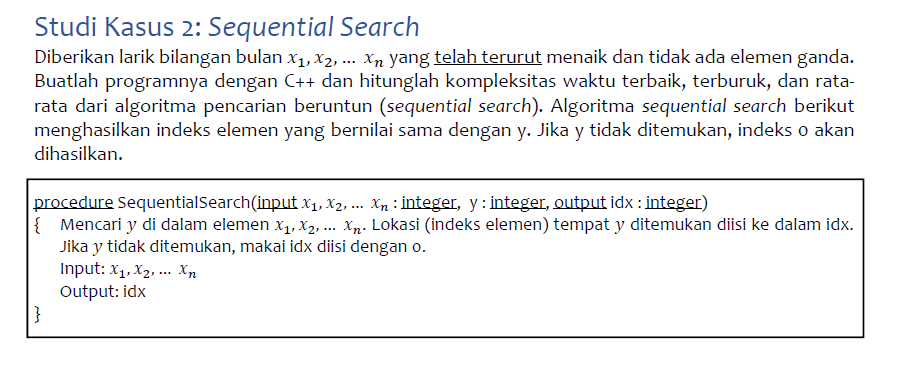
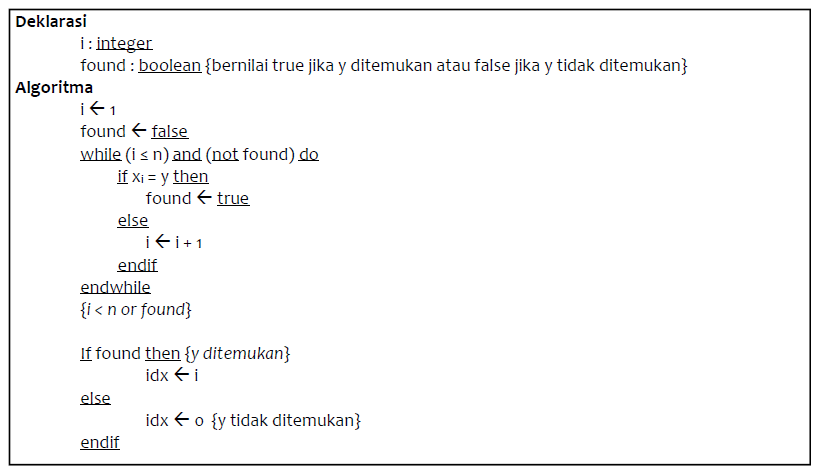
maks = x[i];

}

i=i+1;

}

cout<<"Nilai maksimum adalah : "<<maks;

}  
  
Kompleksitas waktu  
T(n) = 2+2+(n-1)+(n-2)+2(n-2)+2n  
 = 6n-3  
  
  
  
  
Jawab:  
/\*

Nama : Hafidh Akhdan N

NPM : 140810180061

Kelas : A

Program : Studi Kasus 2 "Sequental Search"

\*/

#include<iostream>

using namespace std;

int main(){

int x[5] = {1,2,3,4,5};

int y = 4;

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

int i = 1;

int idx;

bool found = false;

while(i<=n && !found){

if(x[i] == y){

found = true;

}else

i = i+1;

}

if(found == true){

idx = i;

}else

idx = 0;

cout<<"Hasil yang dicari index elemen : "<<idx;

}

Kompleksitas Waktu  
  
Kasus terbaik (Best case) : ini terjadi bila x1 = y

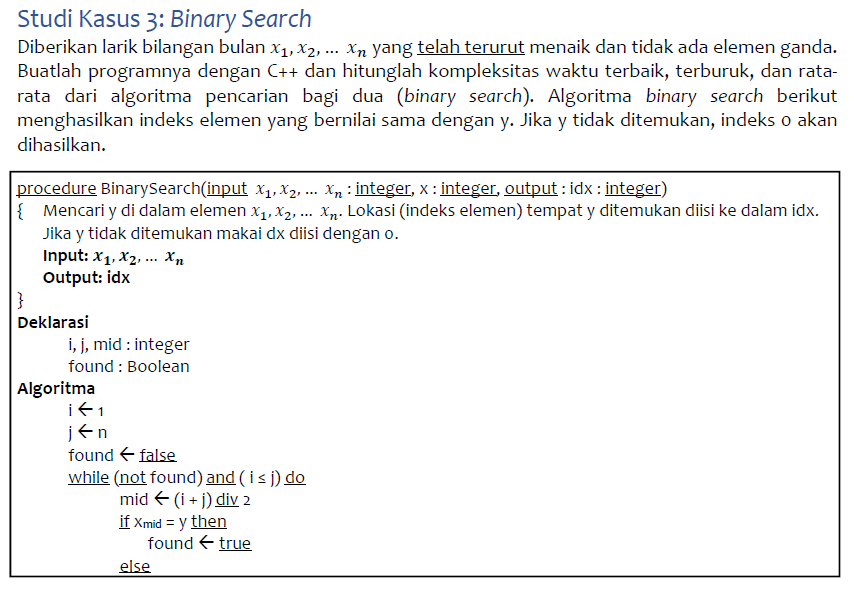
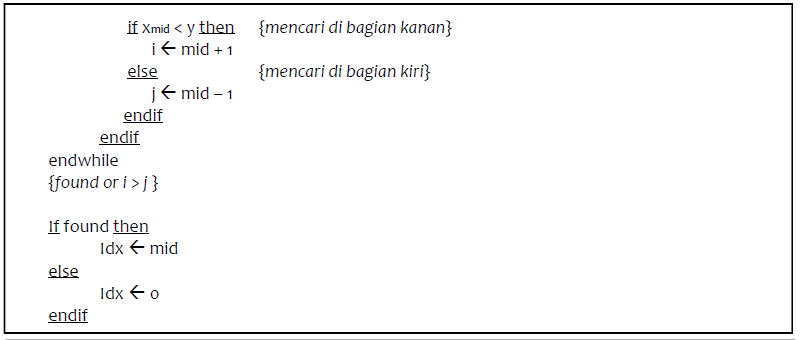
Tmin(n) = 1

Kasus terburuk (Worst case): bila xn = y atau y tidak ditemukan.

Tmax(n) = n

Kasus rata-rata (Average case): Jika *y* ditemukan pada posisi ke-*j*, maka operasi perbandingan (ak = y) akan dieksekusi sebanyak *j*kali.

*T*avg(*n*) = (1+2+3+..+n)/n = (1/2n(1+n))/n = (n+1)/2

  
  
Jawab :

/\*

Nama : Hafidh Akhdan N

NPM : 140810180061

Kelas : A

Program : Studi Kasus 3 "Binary Search"

\*/

#include<iostream>

using namespace std;

int main(){

int x[5]={1,2,3,4,5};

int idx;

int y = 5;

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

int i, j, mid;

bool found;

i = 1;

j = n;

found = false;

while(!found && i<= j){

mid = (i + j)/2;

if (x[mid] == y){

found = true;

}

else if(x[mid] < y){

i = mid+1;

}

else{

j = mid - 1;

}

}

if(found == true){

idx=mid;

}else

idx= 0;

cout<<"Hasil yang dicari indeks elemen : "<<idx;

}

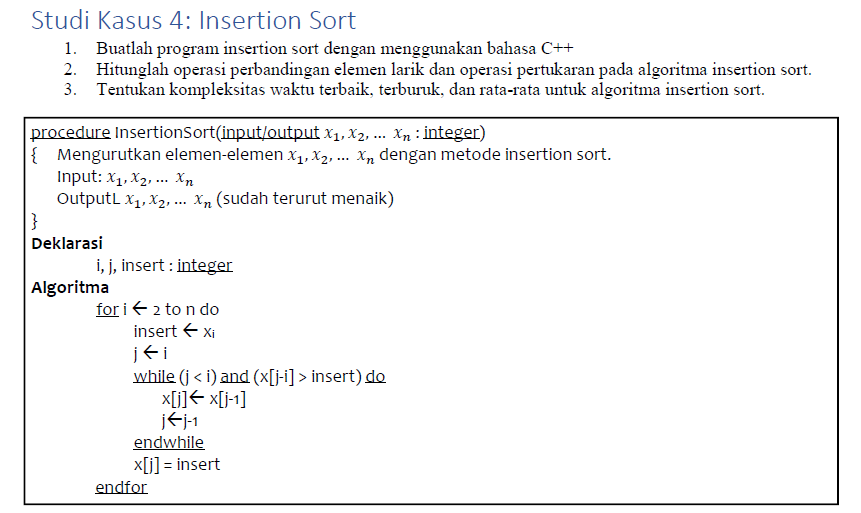
Kompleksitas Waktu

Kasus terbaik (best case) : Jika ditemukan pada arr[mid] atau indeks di tengah

Tmin(n) = 1

Kasus terburuk (worst case) : Jika tidak ditemukan sama sekali

Tmax (n) = 2log n

  
Jawab :

/\*

Nama : Hafidh Akhdan N

NPM : 140810180061

Kelas : A

Program : Studi Kasus 4 "Insertion Sort"

\*/

#include<iostream>

using namespace std;

int main(){

int x[5]={2,3,1,5,4};

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

int i , j, insert;

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

insert= x[i];

j = i - 1;

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

x[j+1] = x[j];

j = j - 1;

}

x[j+1] = insert;

}

for(j = 0; j < n ; j++ ){

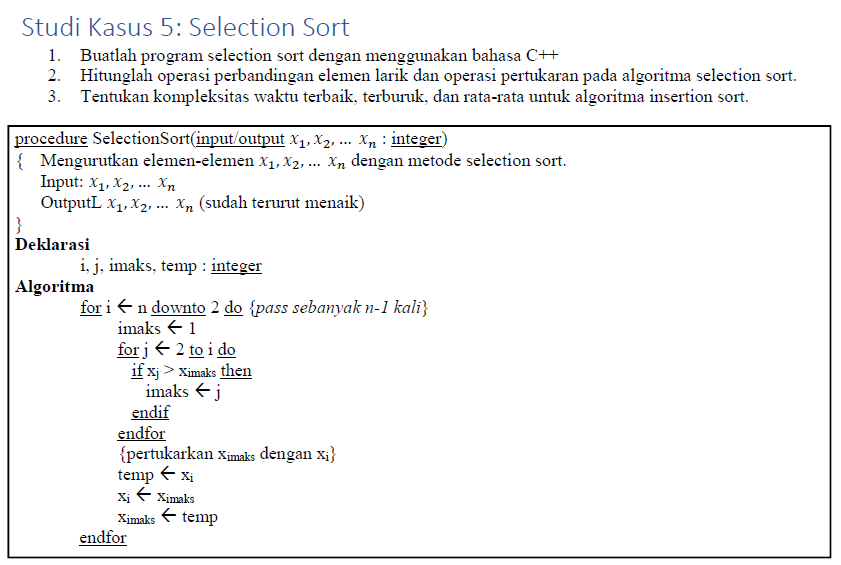
cout<<x[j]<<" ";

}

}

Kompleksitas waktu

Kasus terbaik (best case) : Jika array sudah terurut sehingga loop while tidak dijalankan  
Kasus rata-rata (average case) : Jika sebagian elemen array sudah terurut  
Kasus terburuk (worst case) : Jika array harus diurutkan sebanyak n kali = O (n2).



Jawab :

/\*

Nama : Hafidh Akhdan N

NPM : 140810180061

Kelas : A

Program : Studi Kasus 5 "Selection Sort"

\*/

#include<iostream>

using namespace std;

int main(){

int x[5] = {1,7,3,9,5};

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

int i, j, imaks, temp;

for ( i=2 ; i<n; i++){

imaks = 1;

for( j=2; j<i; j++){

if ( x[j] > x[imaks]){

imaks = j;

}

}

temp = x[i];

x[i] = x[imaks];

x[imaks] = temp;

}

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

cout<<x[i]<<" ";

}

}

Kompleksitas waktu

1. Jumlah operasi perbandingan elemen  
   Untuk setiap loop ke-i,

i = 1 🡪 jumlah perbandingan = n-1

i = 2 🡪 jumlah perbandingan = n-2

i = k 🡪 jumlah perbandingan = n-k

i = n-1 🡪 jumlah perbandingan = 1

Sehingga T(n) = (n-1) + (n-2) + … + 1 = n(n-1)/2

1. Jumlah operasi pertukaran

Untuk setiap loop ke-1 sampai n-1 terjadi satu kali pertukaran elemen sehingga T(n) = n-1.