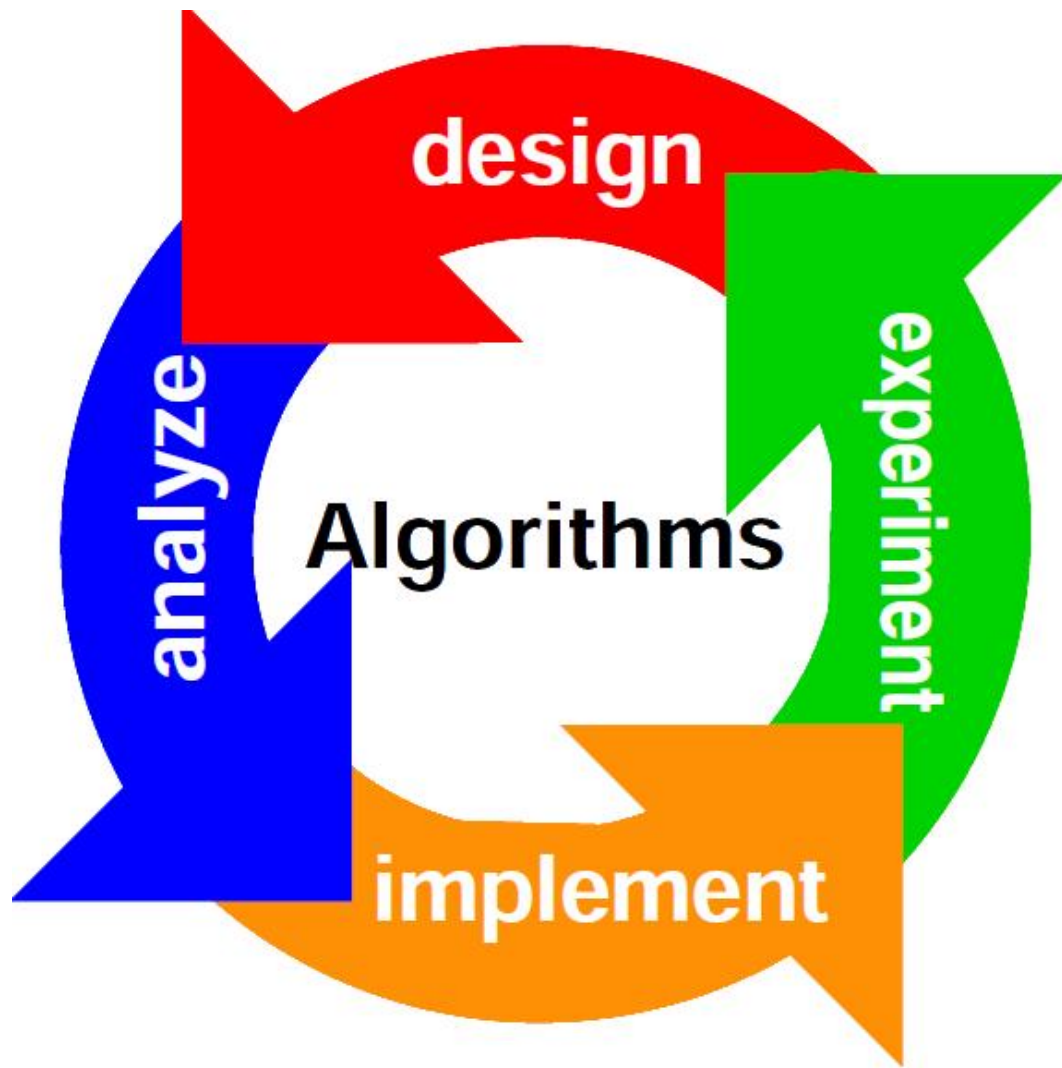


ALGORITMA DAN PEMROGRAMAN 2



TEKNIK PERANGKAT LUNAK
FT - UNPAM

2. PENCARIAN (SEARCHING)

Pencarian (searching) merupakan proses yang fundamental dalam pengolahan data. Proses pencarian adalah menemukan nilai (data) tertentu didalam sekumpulan data yang bertipe sama. Data dapat disimpan secara temporer dalam memori utama atau disimpan secara permanen dalam memori sekunder. Dalam memori utama data disimpan dalam bentuk array(larik) sedangkan dalam memori sekunder dalam bentuk file(arsip). Pencarian elemen dalam larik disebut juga pencarian internal, sedangkan pencarian data yang disimpan dalam memori sekunder disebut juga pencarian eksternal.

Persoalan Pencarian

Diberikan sebuah larik L yang sudah terdefinisi elemen-elemennya, x adalah elemen yang bertipe sama dengan elemen larik L. Carilah x didalam larik L.

Hasil dari proses pencarian dapat bermacam-macam :

- a) Pencarian hanya memeriksa keberadaan x. Keluaran yang diinginkan misalnya berupa pesan bahwa x ditemukan atau tidak ditemukan.

contoh :

write(x,'ditemukan') atau

write(x,'tidak ditemukan')

- b) Hasil pencarian adalah indeks elemen larik. Jika x ditemukan maka indek elemen larik tempat x berada diisikan kedalam idx. Jika x tidak terdapat didalam larik L maka idx diisi dengan nilai khusus misalnya -1.

contoh :

D	35	15	60	70	25	10	12
	0	1	2	3	4	5	6

Misalkan $x=70$, maka $idx=3$. Tetapi jika $x=100$, maka $idx=-1$

- c) Hasil pencarian adalah sebuah nilai boolean yang menyatakan status hasil pencarian. Jika x ditemukan maka sebuah variabel

bertipe boolean misalnya ketemu diisi dengan nilai true, jika sebaliknya maka ketemu diisi dengan false.

I. Pencarian Beruntun (sequential search)

Metode pencarian beruntun adalah proses membandingkan setiap elemen larik satu persatu secara beruntun, mulai dari elemen pertama sampai dengan elemen yang dicari ditemukan atau seluruh elemen sudah ditemukan.

I. a. Versi 1 (Pembandingan elemen dilakukan diawal pengulangan)

(1). Hasil pencarian : sebuah variabel boolean bernilai true bila x ditemukan atau false bila x tidak ditemukan.

Sudah ada array 1D yang dideklarasikan dengan int L[11] dan sudah ada isinya dengan ilustrasi sebagai berikut :

										n
0	1	2	3	4	5	6	7	8	9	10
12	17	10	5	15	25	11	7	25	16	19

Misal elemen yang akan dicari adalah x.

Setiap elemen larik L dibandingkan dengan x mulai dari elemen pertama L[0]. Aksi pembandingan dilakukan selama indek larik i belum melebihi n dan L[i] belum sama dengan n. Aksi pembandingan dihentikan jika L[i] = x atau i = n. Elemen terakhir L[n] diperiksa secara khusus.

<pre> <u>procedure</u> SeqSearch1(<u>input</u> L : LarikInt, <u>input</u> n : <u>integer</u>, <u>input</u> x : <u>integer</u>, <u>output</u> ketemu : <u>boolean</u>) DEKLARASI i : <u>integer</u> </pre>	<pre> #include<iostream.h> void SeqSearch1(int Data[], int n, int x, bool *ketemu); void main(void) { int Data[]={23,56,10,90,35,45,9,100,200,65}; </pre>
---	---

<p>DESKRIPSI</p> <pre> i ← 0 while (i < n) and (L[i] ≠ x) do i ← i + 1 endwhile if L[i] = x then ketemu ← true else ketemu ← false endif </pre>	<pre> int x,i,jmlDat=10;bool ketemu; cout<<"Elemen Array : "; for(i=0;i<jmlDat;i++)cout<<Data[i]<<" ";cout<<endl; cout<<"Masukan data yang akan dicari ?:";cin>>x; SeqSearch1(Data,jmlDat,x,&ketemu); if(ketemu!=false) cout<<"Data yang dicari ditemukan"<<endl; else cout<<"Data yang dicari tidak ada dalam array"<<endl; } void SeqSearch1(int Data[],int n,int x, bool *ketemu) { int i=0; while(i<n-1 && Data[i]!=x)i++; if(Data[i]==x)*ketemu=true; else *ketemu=false; } </pre>
--	---

- (2). Hasil Pencarian : indek elemen larik yang mengandung x
Setiap elemen larik L dibandingkan dengan x mulai dari elemen pertama L[0]. Aksi pembandingan dilakukan selama indek larik i belum melebihi n dan L[i] belum sama dengan n. Aksi pembandingan dihentikan jika L[i] = x atau i = n. Elemen terakhir L[n] diperiksa secara khusus. Keluaran yang dihasilkan oleh prosedur adalah variabel idx yang berisi indek larik tempat x ditemukan. Jika x tidak ditemukan, idx diisi dengan -1.

<pre> procedure SeqSearch2(input L : LarikInt, input n : integer, </pre>	<pre> #include<iostream.h> void SeqSearch2(int Data[], int n, int x, int *idx); void main(void) </pre>
--	--

<u>input</u> x : <u>integer</u> , <u>output</u> <u>idx</u> : <u>integer</u> DEKLARASI i : <u>integer</u> DESKRIPSI i ← 0 <u>while</u> (i < n) <u>and</u> (L[i] ≠ x) <u>do</u> i ← i + 1 <u>endwhile</u> <u>if</u> L[i] = x <u>then</u> idx ← i <u>else</u> idx ← -1 <u>endif</u>	<pre> { int Data[]={23,56,10,90,35,45,9,100,200,65}; int idx,x,i,jmlDat=10; cout<<"Elemen Array : "; for(i=0;i<jmlDat;i++)cout<<Data[i]<<" ";cout<<endl; cout<<"Masukan data yang akan dicari ?:"<<cin>>x; SeqSearch2(Data,jmlDat,x,&idx); if(idx!=-1) cout<<"Data yang dicari berada pada indek "<<idx<<endl; else cout<<"Data yang dicari tidak ada dalam array"<<endl; } void SeqSearch2(int Data[],int n,int x, int *idx) { int i=0; while(i<n-1 && Data[i]!=x)i++; if(Data[i]==x)*idx=i; else *idx=-1; } </pre>
--	--

I. b. Versi 2 (Pembandingan elemen dilakukan didalam pengulangan)

(1). Hasil pencarian : sebuah variabel boolean bernilai true bila x ditemukan atau false bila x tidak ditemukan.

<u>procedure</u> SeqSearch3(<u>input</u> L : LarikInt, <u>input</u> n : <u>integer</u> , <u>input</u> x : <u>integer</u> , <u>output</u> ketemu : <u>boolean</u>) DEKLARASI i : <u>integer</u> DESKRIPSI i ← 0 ketemu ← false <u>while</u> (i <= n) <u>and</u> (<u>not</u>	<pre> #include<iostream.h> void SeqSearch3(int Data[], int n, int x, bool *ketemu); void main(void) { int Data[]={23,56,10,90,35,45,9,100,200,65}; int x,i,jmlDat=10;bool ketemu; cout<<"Elemen Array : "; for(i=0;i<jmlDat;i++)cout<<Data[i]<<" ";cout<<endl; cout<<"Masukan data yang akan dicari ?:"<<cin>>x; SeqSearch3(Data,jmlDat,x,&ketemu); if(ketemu==true) </pre>
--	--

ketemu) <u>do</u> <u>if</u> L[i] = x <u>then</u> ketemu ← <u>true</u> <u>else</u> i ← i + 1 <u>endif</u> <u>endwhile</u>	cout<<"Data yang dicari ditemukan"<<endl; else cout<<"Data yang dicari tidak ada dalam array"<<endl; } void SeqSearch3(int Data[],int n,int x, bool *ketemu) { int i=0;*ketemu=false; while(i<n && !(*ketemu)) { if(Data[i]==x) *ketemu=true; else i++; } }
--	--

Pada versi ini variabel boolean ketemu diinisialisasi dengan nilai false. Setiap elemen L dibandingkan dengan x mulai dari elemen pertama. Jika L[i] sama dengan x, variabel ketemu diisi nilai true dan pengulangan dihentikan. Jika L[i] tidak sama dengan x, perbandingan dilanjutkan untuk elemen berikutnya. Setiap elemen diperiksa termasuk elemen terakhir.

(2). Hasil Pencarian : indek elemen larik yang mengandung x

<u>procedure</u> SeqSearch4(<u>input</u> L : LarikInt, <u>input</u> n : <u>integer</u> , <u>input</u> x : <u>integer</u> , <u>output</u> idx : <u>integer</u>) DEKLARASI i : <u>integer</u> ketemu : <u>boolean</u>	#include<iostream.h> void SeqSearch4(int Data[], int n, int x, int *idx); void main(void) { int Data[]={23,56,10,90,35,45,9,100,200,65}; int idx,x,i,jmlDat=10; cout<<"Elemen Array : "; for(i=0;i<jmlDat;i++)cout<<Data[i]<<" ";cout<<endl; cout<<"Masukan data yang akan dicari ?:"<<cin>>x; SeqSearch4(Data,jmlDat,x,&idx); if(idx!=-1) cout<<"Data yang dicari berada pada indek "<<idx<<endl; else
---	--

<p>DESKRIPSI</p> <pre> i ← 0 ketemu ← false while (i ≤ n) and (not ketemu) do if L[i] = x then ketemu ← true else i ← i + 1 endif endwhile if ketemu then idx ← i else idx ← -1 endif </pre>	<pre> cout<<"Data yang dicari tidak ada dalam array"<<endl; } void SeqSearch4(int Data[],int n,int x, int *idx) { int i=0;bool ketemu=false; while(i<n && !ketemu) { if(Data[i]==x)ketemu=true; elsei++; } if(ketemu) *idx=i; else *idx=-1; } </pre>
--	--

II. Kinerja Metode Pencarian Beruntun

Metode Pencarian Beruntun berjalan lambat. Waktu pencarian sebanding dengan jumlah elemen larik. Misal larik berukuran n elemen maka pada kasus dimana x tidak terdapat dalam larik atau x ditemukan pada elemen terakhir, maka harus dilakukan perbandingan sebanyak n kali. Jadi waktu pencarian dengan metode pencarian beruntun sebanding dengan n.

III. Metode Pencarian Beruntun pada larik terurut

Larik yang jumlah elemen-elemennya terurut dapat meningkatkan kinerja algoritma pencarian beruntun.

Contoh :

(a). Diberikan larik L tidak terurut :

13	16	14	21	76	15
----	----	----	----	----	----

untuk mencari 15, dibutuhkan perbandingan sebanyak 6 kali.

(b). Misalkan larik L diatas sudah diurut naik :

13	14	15	16	21	76
----	----	----	----	----	----

untuk mencari 15, dibutuhkan perbandingan hanya 3 kali

Prosedur berikut adalah algoritma pencarian beruntun pada larik yang terurut menaik, yang merupakan modifikasi dari algoritma sebelumnya dengan merubah $L[i] \neq x$ menjadi $L[i] < x$.

<pre> <u>procedure</u> SeqSearch(<u>input</u> L : LarikInt, <u>input</u> n : <u>integer</u>, <u>input</u> n : <u>integer</u>, <u>output</u> idx : <u>integer</u>) DEKLARASI i : <u>integer</u> DESKRIPSI i ← 0 <u>while</u>(i < n) <u>and</u> (L[i] < x) <u>do</u> i ← i + 1 <u>endwhile</u> <u>if</u> L[i] = x <u>then</u> idx ← i <u>else</u> idx ← -1 <u>endif</u> </pre>	<pre> #include<iostream.h> void SeqSearch(int Data[], int n, int x, int *idx); void main(void) { int Data[]={23,26,30,50,55,65,69,78,80,90}; int idx,x,i,jmlDat=10; cout<<"Elemen Array : "; for(i=0;i<jmlDat;i++)cout<<Data[i]<<" ";cout<<endl; cout<<"Masukan data yang akan dicari ?:";cin>>x; SeqSearch(Data,jmlDat,x,&idx); if(idx!=-1) cout<<"Data yang dicari berada pada indek "<<idx<<endl; else cout<<"Data yang dicari tidak ada dalam array"<<endl; } void SeqSearch(int Data[],int n,int x, int *idx) { int i=0; while(i<n-1 && Data[i]<x)i++; if(Data[i]==x) *idx=i; else *idx=-1; } </pre>
--	--

Tugas :

1. Tulislah algoritma dan program C++ untuk pencarian sequensial versi 1. Pencarian dilakukan dalam sebuah fungsi yang mengembalikan indeks array tempat data yang dicari berada.
2. Tulislah algoritma dan program C++ untuk pencarian sequensial versi 2. Pencarian dilakukan dalam sebuah fungsi yang mengembalikan indeks array tempat data yang dicari berada.
3. Tulislah algoritma dan program C++ untuk pencarian sequensial dengan data yang sudah terurut menurun.
4. Tulislah algoritma dan program C++ untuk pencarian metode bagidua dengan array terurut naik.