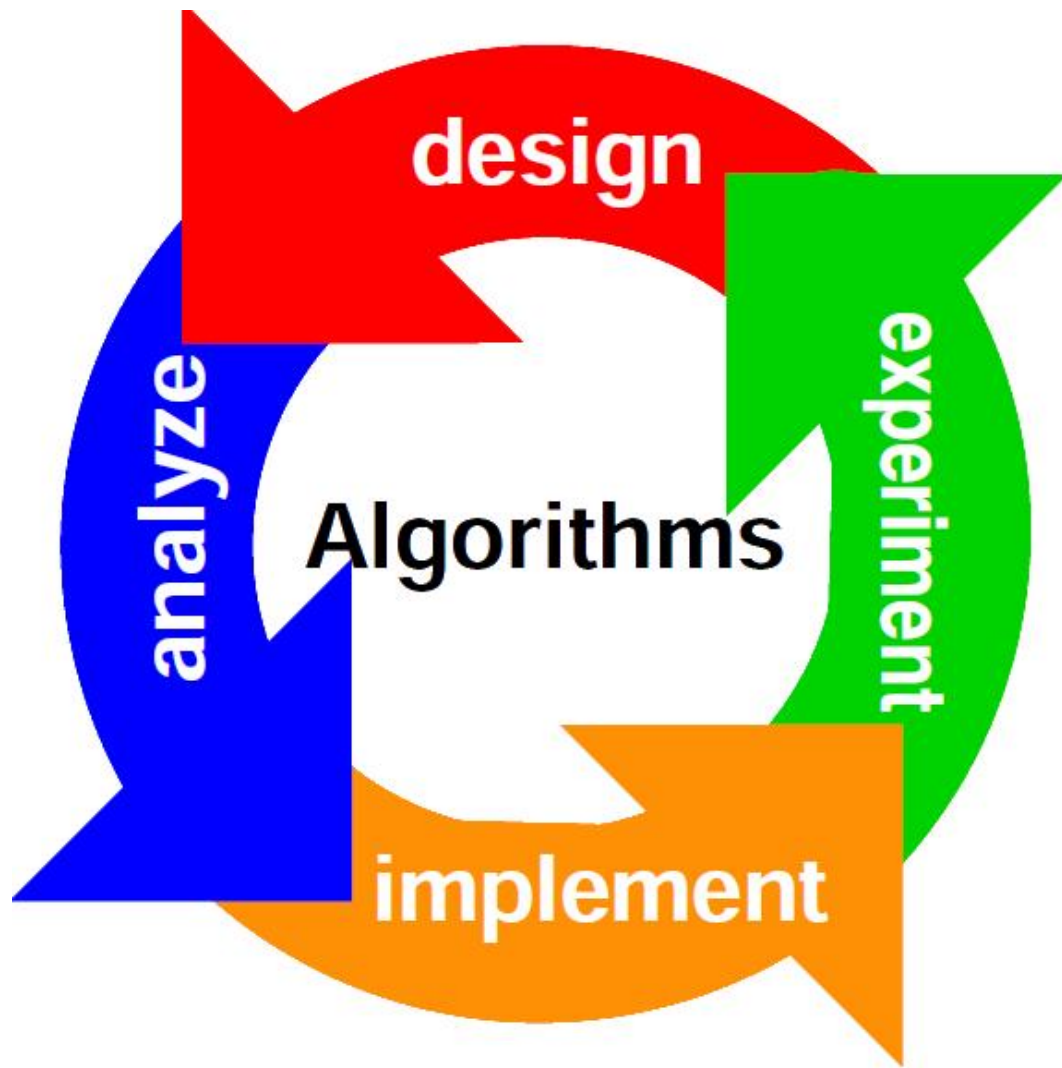


ALGORITMA DAN PEMROGRAMAN 2



TEKNIK PERANGKAT LUNAK
FT - UNPAM

2. PENCARIAN (SEARCHING)lanjutan

IV. Metode Pencarian Bagidua/Biner (Binary Search)

Metode pencarian bagidua lebih efisien dibandingkan metode pencarian beruntun. Metode ini memerlukan data yang sudah terurut. Dalam proses pencarian diperlukan dua buah indeks array, yaitu indeks terkecil (indeks kiri) dan indeks terbesar (indeks kanan).

Misalkan indeks kiri adalah i dan indeks kanan adalah j . Data sudah terurut menurun.

Pada mulanya indek kiri i diinisialisasi dengan 1 dan j diinisialisasi dengan n .

81	76	21	18	16	13	10	7
$i = 0$	1	2	3	4	5	6	$7 = j$

Langkah-langkah dalam metode ini adalah:

Langkah 1:

Bagi dua elemen larik pada elemen tengah. Elemen tengah adalah elemen dengan indeks $k = (i + j) \text{ div } 2$

Langkah 2:

Periksa apakah $L[k] = x$?

Jika $L[k] = x$, pencarian selesai

Jika $L[k] < x$, pencarian dilakukan pada larik bagian kiri, $j = k-1$

Jika $L[k] > x$, pencarian dilakukan pada larik bagian kanan, $i = k + 1$

Langkah 3 :

Ulangi langkah 1 hingga x ditemukan atau $i > j$ (yaitu ukuran array sudah 0)

Contoh 1:

misal elemen yang dicari $x = 18$		Langkah 2 : Bandingkan $L[3] = x?$, ya (x ditemukan,
-----------------------------------	--	--

Langkah 1: $i = 0, j = 7$ $k = (0 + 7)$ <u>div</u> $2 = 3$									proses pencarian selesai)
	81	76	21	18	16	13	10	7	
	0	1	2	3	4	5	6	7	
	kiri				kanan				

Contoh 2: Misal elemen yang dicari $x = 16$

<p>Langkah 1:</p> <p>$i = 0$ dan $j = 7$</p> <p>$k = (0 + 7) \text{ div } 2 = 3$</p> <table><tr><td colspan="4"></td><td colspan="4"></td></tr><tr><td>81</td><td>76</td><td>21</td><td>18</td><td>16</td><td>13</td><td>10</td><td>7</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr></table> <p>kiri kanan</p>									81	76	21	18	16	13	10	7	0	1	2	3	4	5	6	7	<p>Langkah 2:</p> <p>$L[k] = x ?$, tidak</p> <p>$L[k] < x ?$, tidak</p> <p>$L[k] > x ?$, ya, berarti pencarian dilakukan pada array kanan dengan indeks $i = k + 1$</p> <table><tr><td>16</td><td>13</td><td>10</td><td>7</td></tr><tr><td>$i =$</td><td>5</td><td>6</td><td>7 =</td></tr><tr><td>4</td><td></td><td></td><td>j</td></tr></table>	16	13	10	7	$i =$	5	6	7 =	4			j
81	76	21	18	16	13	10	7																														
0	1	2	3	4	5	6	7																														
16	13	10	7																																		
$i =$	5	6	7 =																																		
4			j																																		
<p>Langkah 1':</p> <p>$i = 4, j = 7$</p> <p>$k = (4 + 7) \text{ div } 2 = 5$</p> <table><tr><td colspan="2"></td><td colspan="2"></td></tr><tr><td>16</td><td>13</td><td>10</td><td>7</td></tr><tr><td>4</td><td>5</td><td>6</td><td>7</td></tr></table> <p>kiri kanan</p>					16	13	10	7	4	5	6	7	<p>Langkah 2' :</p> <p>$L[k] = x ?$, tidak</p> <p>$L[k] < x ?$, tidak</p> <p>$L[k] > x ?$, ya, maka pencarian dilakukan pada array bagian kiri</p> <p>$j = k - 1 = 5 - 1 = 4$</p> <table><tr><td>16</td></tr><tr><td>4</td></tr></table>	16	4																						
16	13	10	7																																		
4	5	6	7																																		
16																																					
4																																					
<p>Langkah 1'':</p> <p>$i = 4, j = 4,$</p> <p>$k = (4 + 4) \text{ div } 2 = 4$</p> <table><tr><td>16</td></tr><tr><td>4</td></tr></table>	16	4	<p>Langkah 2'':</p> <p>$L[k] = x ?$, ya (x ditemukan, pencarian dihentikan)</p> <p>Jadi nilai 16 berada pada index k, yaitu $L[4]$</p>																																		
16																																					
4																																					

<pre> <u>procedure</u> BinarySearch(<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,j,k : <u>integer</u> ketemu : <u>boolean</u> DESKRIPSI i ← 0 j ← n ketemu ← <u>false</u> <u>while</u> (<u>not</u> ketemu) <u>and</u> (i ≤ n) <u>do</u> k ← (i + j) <u>div</u> 2 <u>if</u> (L[k] = x) <u>then</u> <u>ketemu</u> ← <u>true</u> <u>else</u> <u>if</u> (L[k] > x) <u>then</u> i ← k + 1 <u>else</u> j ← k - 1 <u>endif</u> <u>endif</u> <u>endwhile</u> <u>if</u> ketemu <u>then</u> idx ← k <u>else</u> idx ← -1 <u>endif</u> </pre>	<pre> #include<iostream.h> #define jmlDat 8 void BinSearch(int Data[], int n, int x, int *idx); void main(void) { int Data[jmlDat]={81,76,21,18,16,13,10,7},x,idx,i; cout<<"Elemen Array : "; for(i=0;i<jmlDat;i++)cout<<Data[i]<<" ";cout<<endl; cout<<"Masukan data yang akan dicari ?:";cin>>x; BinSearch(Data,jmlDat,x,&idx); if(idx!=-1)cout<<"Data yang dicari berada pada indeks : "<<idx<<endl; else cout<<"Data yang dicari tidak ada dalam array"<<endl; } void BinSearch(int Data[],int n,int x, int *idx) { bool ketemu = false; int top = n-1,bottom = 0,mid; while(bottom<=top && !ketemu) { mid=(top+bottom)/2; if(Data[mid]==x)ketemu=true; else if(Data[mid]<x)top=mid-1; else bottom=mid+1; } if(ketemu) *idx=mid; else *idx=-1; } </pre>
--	--

Tugas :

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