



BINARY SEARCHING

[PENCARIAN BAGI DUA]

1. Introduction
2. Penelusuran
3. Algoritma



INTRODUCTION

- Merupakan metode pencarian yang diterapkan pada sekumpulan data yang sudah terurut, baik itu urut naik (*ascending*) atau urut turun (*descending*) .
- Contoh dalam kehidupan sehari-hari: Buku telepon, kamus, data mahasiswa, dll.



PENELUSURAN

- Prinsip: membagi data atas dua bagian
- Data yang tersimpan dalam larik dalam keadaan sudah terurut
- Asumsi bahwa larik sudah terurut secara menurun
- Diperlukan dua buah indeks larik untuk indeks min (ujung kiri) dan max (ujung kanan)



PENELUSURAN \rightarrow

- Misalkan indeks kiri i dan indeks kanan j
- Kondisi awal, $i=1$ dan $j=N$
- Langkah 1:
 - Bagi dua elemen larik pada elemen tengah.
 - Elemen tengah adalah elemen dengan indeks $k = (i+j) \text{ div } 2$
 - Elemen tengah, $L[k]$, membagi dua bagian, yaitu bagian kiri $L[i..k-1]$ dan bagian kanan $L[k+1..j]$



PENELUSURAN [\rightarrow]

- Langkah 2

- Periksa apakah $L[k] = X$
- Jika $L[k] = X$, pencarian dihentikan
- Jika $L[k] \neq X$, maka:
 - Jika $L[k] < X$, pencarian dilakukan pada larik bagian kiri
 - Jika $L[k] > X$, pencarian dilakukan pada larik bagian kanan

- Langkah 3

- Ulangi langkah 1 sampai X ditemukan atau $I > j$ (yaitu, ukuran larik sudah nol!)



PENELUSURAN [→]

Ilustrasi

Misalkan diberikan larik L dengan delapan buah elemen yang sudah terurut menurun seperti di bawah ini:

81	76	21	18	16	13	10	7
i=1	2	3	4	5	6	7	8=j

PENELUSURAN [→]

- Misalkan elemen yang dicari adalah $X=16$
- Langkah 1 :
 - $i=1$ dan $j=8$
 - Indeks elemen tengah $k=(1+8) \text{ div } 2 = 4$

81	76	21	18	16	13	10	7
$i=1$	2	3	4	5	6	7	$8=j$
kiri				kanan			

- Langkah 2:
 - $L[4] = 18$? Ya! (X ditemukan, pencarian dihentikan)

PENELUSURAN [→]

- Misalkan elemen yang dicari adalah $X=16$
- Langkah 1 :
 - $i=1$ dan $j=8$
 - Indeks elemen tengah $k=(1+8) \text{ div } 2 = 4$

81	76	21	18	16	13	10	7
i=1	2	3	4	5	6	7	8=j
kiri				kanan			

PENELUSURAN \rightarrow

- Langkah 2 :

- $L[4] = 16$? Tidak!
- $L[4] > 16$? Ya!
 - Pencarian dilakukan pada larik bagian kanan
 - $i=k+1=5$ dan $j=8$ (tetap)

16	13	10	7
$i=5$	6	7	$8=j$

- Langkah 1' :

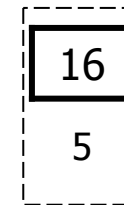
- $i=5$ dan $j=8$
- Indeks elemen tengah
 $k=(5+8) \text{ div } 2 = 6$

16	13	10	7
5	6	7	8
kiri `		kanan `	

PENELUSURAN \rightarrow

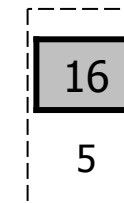
- Langkah 2' :

- $L[6] = 16$? Tidak!
- $L[6] > 16$? Tidak!
 - Lakukan pencarian pada larik bagian kiri
 - $i=5$ (tetap) dan $j=k-1=5$



- Langkah 1'' :

- $i=5$ dan $j=5$
- Indeks elemen tengah $k=(5+5) \div 2 = 5$



- Langkah 2'' :

- $L[5]=16$? Ya! (X ditemukan, pencarian dihentikan)

PENELUSURAN [→]

- Misalkan elemen yang dicari adalah $X=100$
- Langkah 1:
 - $i=1$ dan $j=8$
 - Indeks elemen tengah $k=(1+8) \text{ div } 2 = 4$

81	76	21	18	16	13	10	7
i=1	2	3	4	5	6	7	8=j
kiri				kanan			

PENELUSURAN \rightarrow

- Langkah 2:

- $L[4] = 100$? Tidak!
- $L[4] > 100$? Tidak!
 - Pencarian dilakukan pada larik bagian kiri
 - $i=1$ (tetap) dan $j=k-1 = 3$

81	76	21
$i=1$	2	$3=j$

- Langkah 1' :

- $i=1$ dan $j=3$
- Indeks elemen tengah
 $k=(1+3) \text{ div } 2 = 2$

81	76	21
1	2	3

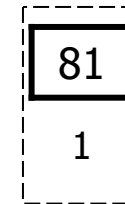
kiri `

kanan `

PENELUSURAN \rightarrow

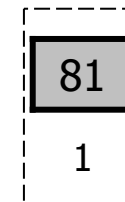
- Langkah 2' :

- $L[2] = 100$? Tidak!
- $L[2] > 100$? Tidak!
 - Lakukan pencarian pada larik bagian kiri
 - $i=1$ (tetap) dan $j=k-1=1$



- Langkah 1'' :

- $i=1$ dan $j=1$
- Indeks elemen tengah $k=(1+1) \text{ div } 2 = 1$



- Langkah 2'' :

- $L[1] = 100$? Tidak!
- $L[1] > 100$? Tidak!
 - Lakukan pencarian pada larik bagian kiri
 - $i=1$ (tetap) dan $j=k-1 = 0$
 - Karena $i > j$, tidak ada lagi bagian larik yang tersisa. X tidak ditemukan. Pencarian dihentikan



ALGORITMA

Procedure BagiDua1 (input L: larik, input N: integer, input X: integer, output IDX: integer)

KAMUS

i, j : integer

k : integer

ketemu : boolean

ALGORITMA

i \leftarrow 1

j \leftarrow N

ketemu \leftarrow false

while (not ketemu) and (i < = j) do

 k \leftarrow (i + j) div 2

if (L[k] = X) then

 ketemu \leftarrow true

else

if (L[k] > X) then

 i \leftarrow k + 1

else

 j \leftarrow k - 1

endif

endif

endwhile

if (ketemu) then

 IDX \leftarrow k

else

 IDX \leftarrow 0

endif



ALGORITMA [→]

Procedure BagiDua2 (input L: larik, input N: integer, input X: integer, output IDX: integer)

KAMUS

i, j : integer

k : integer

ketemu : boolean

ALGORITMA

i ← 1

j ← N

ketemu ← false

while (not ketemu) and (i < = j) do

 k ← (i + j) div 2

if (L[k] = X) then

 ketemu ← true

else

if (L[k] < X) then

 i ← k + 1

else

 j ← k - 1

endif

endif

endwhile

if (ketemu) then

 IDX ← k

else

 IDX ← 0

endif

Galih - IF - UNIKOM