



# *PERTEMUAN 11*

# **TEHNIK SEARCHING**



# TEHNIK SEARCHING

## Tehnik Pencarian :

1. Tehnik Pencarian Tunggal :
  - a. Tehnik Sequential Search / Linier Search
  - b. Tehnik Binary Search
  
2. Tehnik Pencarian Nilai MAXMIN :
  - a. Tehnik StraitMAXMIN
  - b. Tehnik D and C



## 1. Tehnik Pencarian Tunggal :

### *a. Linear/Sequential Search ( Untuk data yg belum terurut / yg sudah terurut )*

Pencarian yg dimulai dari record-1 diteruskan ke record selanjutnya yaitu record-2, ke-3,...., sampai diperoleh isi record sama dengan informasi yg dicari

Algoritma :

1. Tentukan  $I = 1$
2. Ketika Nilai  $(I) \neq X$  Maka Tambahkan  $I = I + 1$
3. Ulangi langkah No. 2 sampai Nilai  $(I) = X$
4. Jika Nilai  $(I) = N+1$  Maka Cetak "Pencarian Gagal"  
selain itu Cetak " Pencarian Sukses "



## **b. Binary Search ( Untuk data yg sudah terurut )**

Digunakan mencari sebuah data pd himp.data-data yg tersusun secara urut, yaitu data yg telah diurutkan dr besar ke kecil/sebaliknya. Proses dilaksanakan pertama kali pd bgn tengah dr elemen himpunan, jk data yg dicari ternyata  $<$  elemen bagian atasnya, maka pencarian dilakukan dr bagian tengah ke bawah.



## Algoritma :

1. Low = 1 , High = N
2. Ketika Low  $\leq$  High Maka kerjakan langkah No .3,  
Jika tidak Maka kerjakan langkah No.7
3. Tentukan Nilai Tengah dengan rumus  
 $mid = ( Low + High ) \text{ Div } 2$
4. Jika  $X < \text{Nil. Tengah}$  Maka High = Mid -1
5. Jika  $X > \text{Nil. Tengah}$  Maka Low = Mid +1
6. Jika  $X = \text{Nil. Tengah}$  Maka Nil. Tengah = Nil. Yg dicari
7. Jika  $X > High$  Maka Pencarian GAGAL



## 2. Tehnik Pencarian MAXMIN

### Searcing dengan Tehnik STRAITMAXMIN

Menentukan / mencari elemen max & min.  
Pada Himpunan yg berbentuk array linear.  
Waktu tempuh/*time complexity* yg digunakan  
untuk menyelesaikan pencarian hingga  
mendapatkan solusi yg optimal terbagi atas  
***best case, average case*** dan ***worst case***.



Algoritma untuk mencari elemen MaxMin :

## PROCEDURE

### STRAITMAXMIN(A,n,i,max,min)

int i,n, A [n], max,min

max  $\leftarrow$  min  $\leftarrow$  A[0]

FOR i  $\leftarrow$  1 To n

    IF A[i] > max; max  $\leftarrow$  A[i];

        ELSE IF A[i] < min ; min  $\leftarrow$  A[i] ENDIF

    ENDIF

REPEAT

END STRAITMAXMIN



## BEST CASE

- Keadaan yg tercapai jika elemen pada himpunan A disusun secara *increasing* (menaik). Dengan perbandingan waktu  $n - 1$  kali satuan operasi.
- Contoh : Terdapat himp.A yg berisi 4 buah bilangan telah disusun secara *increasing* dengan  $A[0] = 2$ ,  $A[1] = 4$ ,  $A[2]=5$ ,  $A[3]=10$ . Tentukan / cari Bilangan Max&Min serta jumlah operasi perbandingan yg dilakukan.





- *Penyelesaian*

untuk masalah tersebut dapat digunakan procedure STRAITMAXMIN yg menghasilkan bilangan Min=2 & bilangan Max=10, Operasi perbandingan data mencari bilangan MaxMin dari himpunan tersebut  $(n-1) = 3$  kali operasi perbandingan.



## WORST CASE

- Terjadi jika elemen dalam himp. disusun secara *decreasing* (menurun). Dengan. Oprasi perbandingan sebanyak  **$2(n-1)$**  kali satuan operasi.
- **Contoh** : Mencari elemen MaxMin & jumlah oprasi perbandingan yg dilakukan terhadap himp.A yg disusun *decreasing*.  
 $A[0]=80, A[1]=21, A[2]=6, A[3]=-10$



## **Penyelesaian**

- untuk masalah tersebut dengan proses STRAITMAXMIN adalah elemen  $\text{max}=80$  & elemen  $\text{min}=-10$ , Operasi perbandingan untuk elemen Maxmin tersebut adalah  $2(4-1) = 6$  kali satuan operasi.



## AVERAGE CASE

- Jika pencarian elemen MaxMin dilakukan pada elemen dalam himpunan yg tersusun secara acak (tidak decreasing/tidak increasing). Jumlah oprasi. Perbandingan yg dilakukan adalah rata-rata waktu tempuh *best case* & *worst case*, yaitu  $\frac{1}{2} [ (n-1) + 2(n-1) ] = ( \frac{3n}{2} - 1 )$  kali.
- Contoh, Pada himpunan A yg berisi { 5,-4, 9,7 } dilakukan pencarian elemen max & min dengan menggunakan proses STRAITMAXMIN. Berapa elemen maxmin yg didapatkan & jumlah oprasi perbandingan yg dilakukan.



- Penyelesaiannya :

Elemen max=9, & elemen min=-4.  
Jumlah operasi perbandingan adalah  $(3 \cdot 4/2 - 1) = 5$  kali satuan operasi.



## **Searching dengan Tehnik DANDC**

- Dengan Prinsip Dasar Metode Devide & akan dapat dipecahkan suatu permasalahan proses Searching elemen Max&Min dengan teknik DANC
- Contoh : Tentukan elemen MaxMin suatu array A yg terdiri 9 bil. :

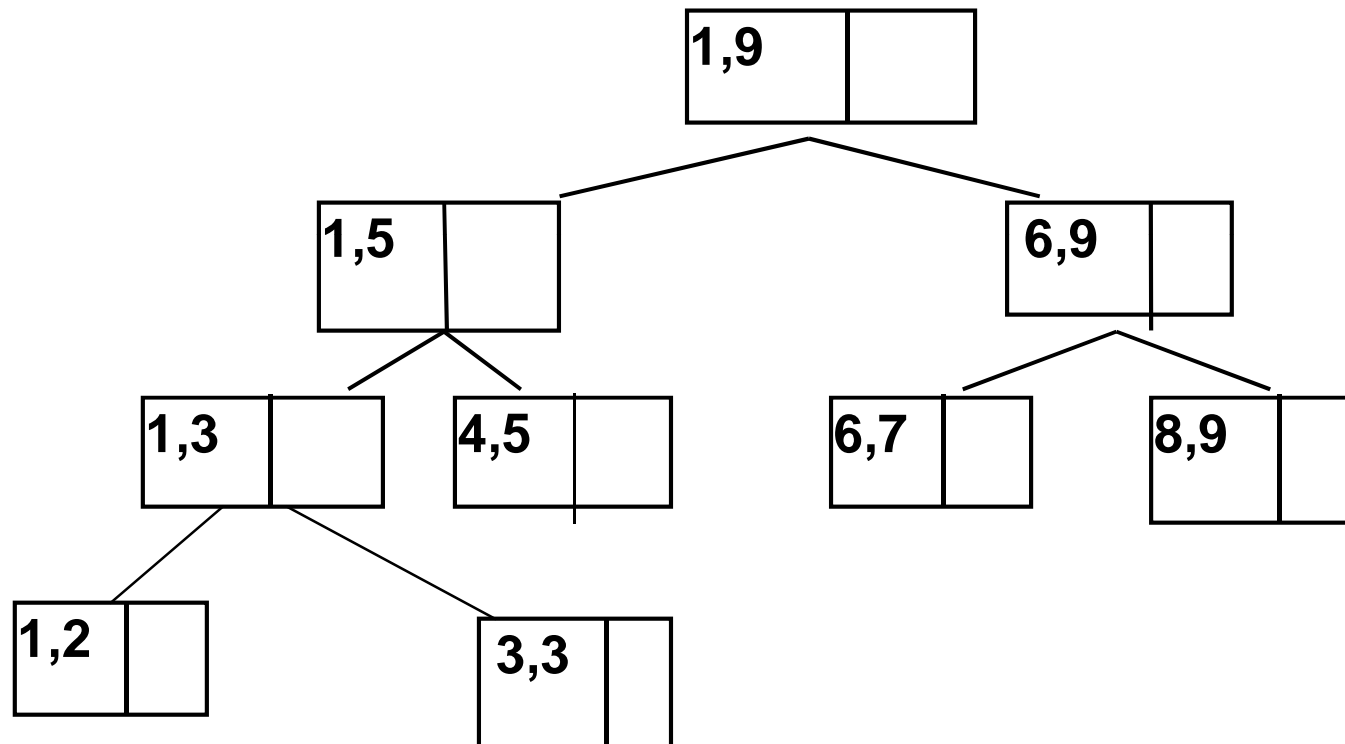
$$A[1] = 22, \quad A[4] = -8, \quad A[7] = 17$$

$$A[2] = 13, \quad A[5] = 15, \quad A[8] = 31$$

$$A[3] = -5, \quad A[6] = 60, \quad A[9] = 47$$



# Penyelesaian :





Lalu *Proses tree call* dr setiap elemen yg ditunjuk pada bagan tree tersebut diatas. Dengan cara, membalik terlebih dahulu posisi tree dr bawah ke atas. Lalu mengisinya dengan elemen-elemennya sesuai dengan bagan tree. Perhatikan bagan *tree call* ini :

