

PERTEMUAN 11

TEKNIK SEARCHING

TEKNIK SEARCHING

1. Pengertian Teknik Searching

Teknik dalam memilih dan menyeleksi sebuah elemen dari beberapa elemen yang ada.

2. Teknik Pencarian (Searching)

A. Teknik Pencarian Tunggal

1. Teknik Sequential Search/Linier Search
2. Teknik Binary Search

B. Teknik Pencarian Nilai MAXMIN

1. Teknik StraitMAXMIN
2. Teknik D and C

Teknik Pencarian Tunggal

a. Linear/Sequential Search (Untuk data yang belum terurut/yang sudah terurut)

Pencarian yang dimulai dari record-1 diteruskan ke record selanjutnya yaitu record-2, ke-3,..., sampai diperoleh isi record sama dengan informasi yang dicari (Nilai X).

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 "

Teknik Sequential/Linier Search (Lanjutan)

Contoh 1:

Data $A = \{ 10, 4, 9, 1, 15, 7 \}$

Dicari **15**

Langkah pencariannya:

Langkah 1: $A[1] = 10$

Langkah 2: $10 \neq 15$, maka $A[2] = 4$

Langkah 3: ulangi langkah 2

Langkah 2: $4 \neq 15$, maka $A[3] = 9$

Langkah 2: $9 \neq 15$, maka $A[4] = 1$

Langkah 2: $1 \neq 15$, maka $A[5] = 15$

Langkah 2: $15 = 15$

Langkah 4: “Pencarian Sukses”

Teknik Sequential/Linier Search (Lanjutan)

Contoh 2:

Diberikan Larik(L) dengan $n=6$ elemen

Data dari Larik L adalah sebagai berikut:

13	16	14	21	76	15
1	2	3	4	5	6

Misalkan data yang dicari adalah $x = 15$

- data $X = 15$ akan dibandingkan dengan ke $l=1$ sampai $l=6$

Langkah Pencariannya:

Langkah 1:

1. $i=1 \rightarrow \text{data} = 13, X=15$
2. Bandingkan $13 <> 15$, tambahkan $I = I + 1$

Langkah 2:

1. $I = 2 \rightarrow \text{data} = 16, X=15$
2. Bandingkan $16 <> 15$, tambahkan $I = I + 1$

Langkah 3:

1. $i=14 \rightarrow \text{data} = 13, X=15$
2. Bandingkan $14 <> 15$, tambahkan $I = I + 1$

Teknik Sequential/Linier Search (Lanjutan)

Teknik Sequential/Linier Search

Langkah 4:

1. $i=4 \rightarrow \text{data} = 21, X=21$
2. Bandingkan $21 <> 15$, tambahkan $I = I + 1$

Langkah 5:

1. $i=5 \rightarrow \text{data} = 76, X=15$
2. Bandingkan $76 <> 15$, tambahkan $I = I + 1$

Langkah 6:

1. $i=1 \rightarrow \text{data} = 15, X=15$
2. Bandingkan $15 = 15$, maka data berhasil ditemukan.
3. Indeks yang dicari ditampilkan yaitu 6

Contoh 3

Apabila ditemukan kondisi:

Nilai $(i) = N + 1$, maka pencarian tidak ditemukan atau **gagal**.

Dikarenakan jumlah elemen adalah N , $N + 1$ artinya data yang dicari bukan merupakan elemen data dari N .

Teknik Pencarian Tunggal (Lanjutan)

b. Binary Search (Untuk data yang sudah terurut)

Digunakan mencari sebuah data pada himpunan data-data yang tersusun secara urut, yaitu data yang telah diurutkan dari besar ke kecil/sebaliknya. Proses dilaksanakan pertama kali pada bagian tengah dari elemen himpunan, jika data yang dicari ternyata $<$ elemen bagian atasnya, maka pencarian dilakukan dari bagian tengah ke bawah

Binary Search (Lanjutan)

Algoritma **Binary Search**

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
(Low + High) Div 2
4. Jika $X <$ Nilai Tengah, Maka High = Mid -1, Ulangi langkah 1
5. Jika $X >$ Nilai Tengah, Maka Low = Mid +1, Ulangi langkah 1
6. Jika $X =$ Nilai Tengah, Maka Nilai Tengah = Nilai yang dicari
7. Jika $X >$ High Maka Pencarian GAGAL

Binary Search (Lanjutan)

Contoh:

Data A = { 1, 3, 9, 11, 15, 22, 29, 31, 48 }

Dicari **3**

Langkah Pencariannya:

Langkah 1: Low = 1 dan High = 9

Langkah 2: Low \leq High (jika YA ke **L-3**, jika TDK ke **L-7**)

Langkah 3: Mid = $(1+10) \div 2 = 5$ yaitu **15**

Langkah 4: $3 < 15$, maka High = $5 - 1 = 4$ yaitu **11**

Langkah 1: Low = 1 dan High = 4

Langkah 2: Low \leq High

Langkah 3: Mid = $(1+4) \div 2 = 2$ yaitu **3**

Langkah 4: $3 < 3$, ke langkah 5

Langkah 5: $3 > 3$, ke langkah 6

Langkah 6: $3 = 3$ (Pencarian berhasil)

Teknik Pencarian MAXMIN

a. Teknik STRAITMAXMIN

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

Teknik STRAITMAXMIN (*Lanjutan*)

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 yang tercapai jika elemen pada himpunan A disusun secara *increasing* (menaik).
- Dengan perbandingan waktu $n - 1$ kali satuan operasi.

Contoh :

Terdapat himpunan A yang 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 yang dilakukan.

Penyelesaian Best Case:

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

BEST CASE (Lanjutan)

Penyelesaian Best Case detail:

Contoh: $A = \{ 2, 4, 5, 10 \}$

Operasi Perbandingan : $4 - 1 = 3$

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

$\text{max} = \text{min} = 2$

FOR $i \leftarrow 1$ TO 3

Saat $i=1$ APK $4 > 2$ maka $\text{max} = 4$ dan kembali ke FOR

Saat $i=2$ APK $5 > 4$ maka $\text{max} = 5$ dan kembali ke FOR

Saat $i=3$ APK $10 > 5$ maka $\text{max} = 10$

Maka Nilai MIN=2 dan MAX=10

WORST CASE

- Terjadi jika elemen dalam himpunan disusun secara *decreasing* (menurun).
- Dengan Operasi perbandingan sebanyak **$2(n-1)$** kali satuan operasi.

Contoh :

Mencari elemen MaxMin & jumlah oprasi perbandingan yang dilakukan terhadap himpunan A yang disusun *decreasing*.

$A[0]=80, A[1]=21, A[2]=6, A[3]=-10$

WORST CASE (Lanjutan)

Penyelesaian Worst Case

- 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.

WORST CASE (Lanjutan)

Penyelesaian Worst Case Detail:

Contoh: $A = \{ 80, 21, 6, -10 \}$

Operasi Perbandingan : $2(4 - 1) = 6$

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

max = min = 80

FOR $i \leftarrow 1$ TO 3

Saat $i=1$ APK $21 > 80$ maka

ELSE APK $21 < 80$ maka min = 21

Saat $i=2$ APK $6 > 80$ maka

ELSE APK $6 < 21$ maka min = 6

Saat $i=3$ APK $-10 > 80$ maka

ELSE APK $-10 < 6$ maka min = -10

Maka Nilai MIN=-10 dan MAX=80

AVERAGE CASE

- Jika pencarian elemen MaxMin dilakukan pada elemen dalam himpunan yang tersusun secara acak (tidak decreasing/tidak increasing).
- Jumlah operasi Perbandingan yang 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 yang didapatkan & jumlah oprasi perbandingan yang dilakukan.

AVERAGE CASE (Lanjutan)

Penyelesaian Average Case:

Elemen max=9, & elemen min=-4.

Jumlah operasi perbandingan adalah $(3 \cdot 4/2 - 1) = 5$ kali satuan operasi.

AVERAGE CASE (Lanjutan)

Penyelesaian Average Case Detail:

Contoh: $A = \{ 5, -4, 9, 7 \}$

Operasi Perbandingan : $3(4/2) - 1 = 5$

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

max = min = 5

FOR $i \leftarrow 1$ TO 3

Saat $i=1$ APK $-4 > 5$ maka

ELSE APK $-4 < 5$ maka min = -4

Saat $i=2$ APK $9 > 5$ maka max = 9

Saat $i=3$ APK $7 > 9$ maka

ELSE APK $7 < -4$ maka Keluar dari perulangan

Maka Nilai MIN=-4 dan MAX=9

Teknik Pencarian Nilai MAXMIN (Lanjutan)

b. Teknik D AND C

- Dengan Prinsip Dasar Metode *Devide & Conquer* akan dapat dipecahkan suatu permasalahan proses Searching elemen Max&Min dengan teknik D and C

- **Contoh :**

Tentukan elemen MaxMin suatu array A yang terdiri dari 9 bilangan :

$$A[1] = 22,$$

$$A[4] = -8,$$

$$A[7] = 17$$

$$A[2] = 13,$$

$$A[5] = 15,$$

$$A[8] = 31$$

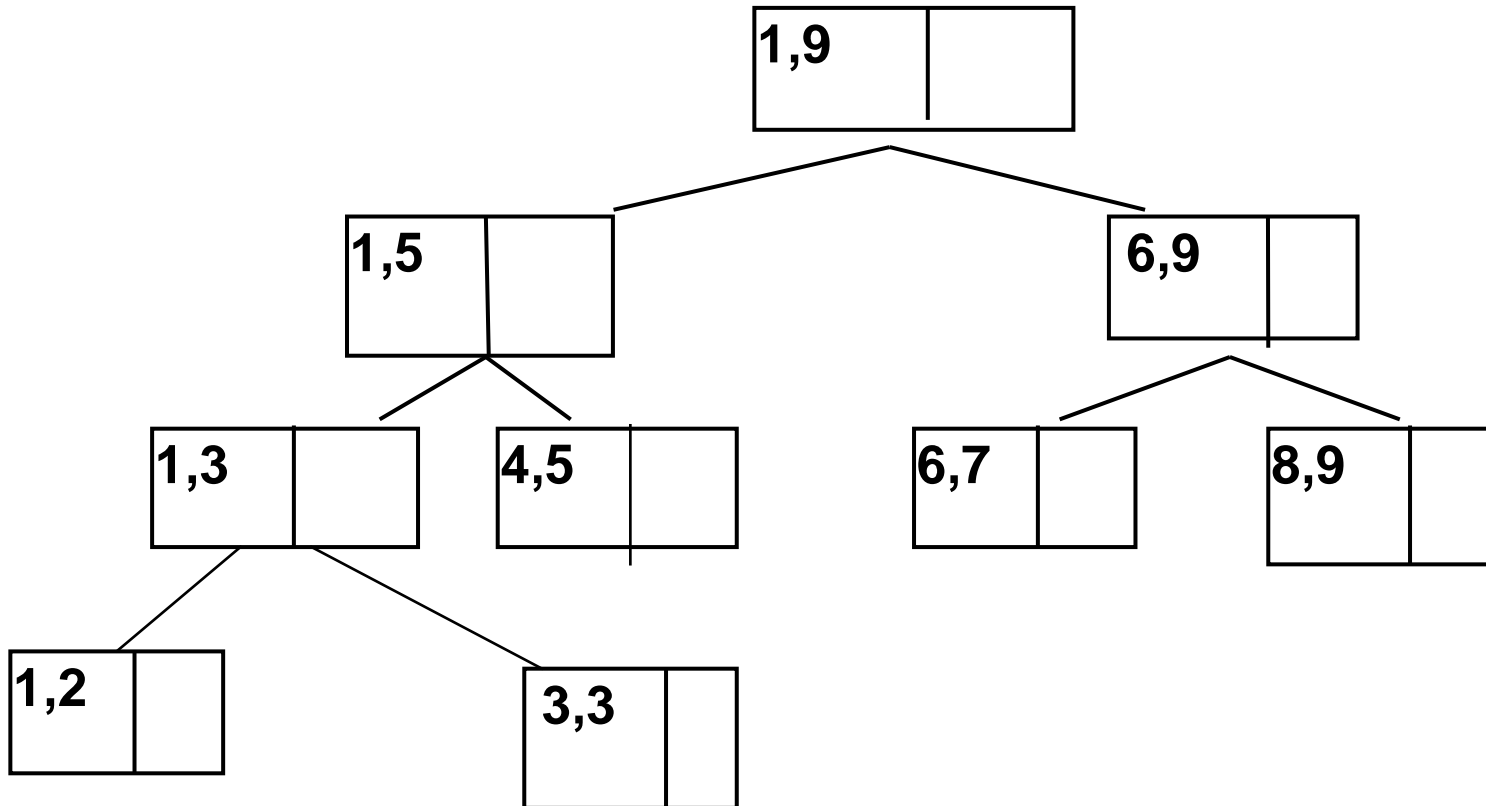
$$A[3] = -5,$$

$$A[6] = 60,$$

$$A[9] = 47$$

Teknik D AND C (Lanjutan)

Penyelesaian *Teknik D and C*



Penyelesaian Teknik D AND C

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

$A = \{ 22, 13, -5, -8, 15, 60, 17, 31, 47 \}$

Posisi penggabungan →

Max, Min

