

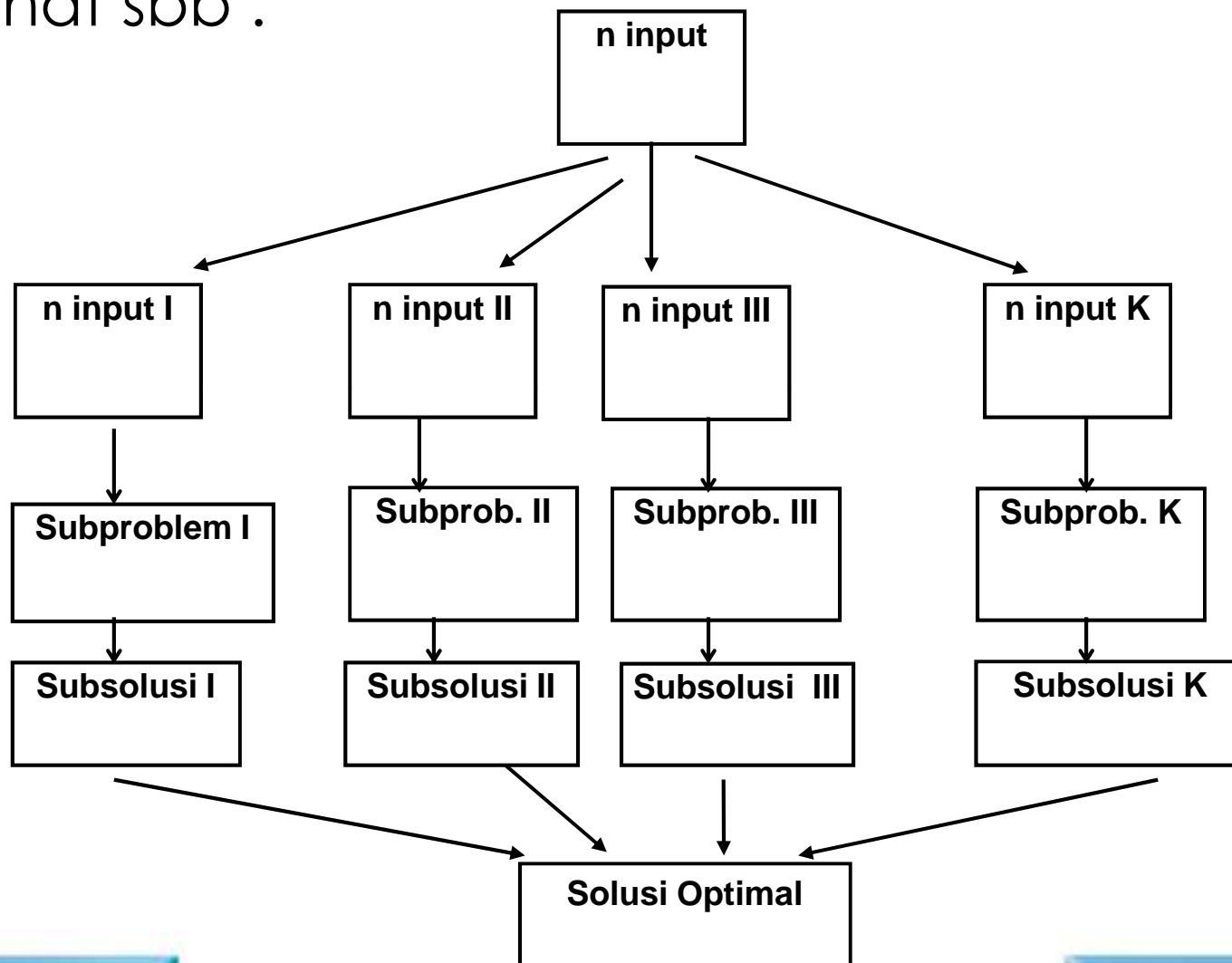


## *PERTEMUAN 10*

# **METODE DEVIDE AND CONQUER**



- Bentuk Umum Proses Metode D And C dpt dilihat sbb :





# **SORTING**

- 1. Metode Selection Sort**
- 2. Metode Buble Sort**
- 3. Metode Merge Sort**
- 4. Metode Quick Sort**
- 5. Metode Insertion.**

**Hal yg mempengaruhi Kecepatan Algoritma Sort :  
Jumlah Operasi Perbandingan & Jumlah Operasi  
Pemindahan Data**



## ***SELECTION SORT***

**Tehnik pengurutan dgn cara pemilihan elemen atau proses kerja dgn memilih elemen data terkecil utk kemudian dibandingkan & ditukarkan dgn elemen pd data awal, dst s/d seluruh elemen shg akan menghasilkan pola data yg telah disort.**



## **Prinsip Kerja dari Teknik Selection Sort ini adalah :**

- 1. Pengecekan dimulai data ke-1 sampai dengan data ke-n**
- 2. Tentukan bilangan dengan Index terkecil dari data bilangan tersebut**
- 3. Tukar bilangan dengan Index terkecil tersebut dengan bilangan pertama (  $I = 1$  ) dari data bilangan tersebut**
- 4. Lakukan langkah 2 dan 3 untuk bilangan berikutnya (  $I = I + 1$  ) sampai didapatkan urutan yg optimal.**



**Contoh : 22 10 15 3 8 2**

### Iterasi 1

|                    | <b>1</b>                        | <b>2</b>  | <b>3</b>  | <b>4</b>  | <b>5</b> | <b>6</b>  |
|--------------------|---------------------------------|-----------|-----------|-----------|----------|-----------|
| <b>Langkah 1 :</b> | <b>22</b>                       | <b>10</b> | <b>15</b> | <b>3</b>  | <b>8</b> | <b>2</b>  |
| <b>Langkah 2</b>   | <b>:</b>                        | <b>22</b> | <b>10</b> | <b>15</b> | <b>3</b> | <b>8</b>  |
|                    | <b>2</b>                        |           |           |           |          |           |
| <b>Langkah 3 :</b> | <b>2</b>                        | <b>10</b> | <b>15</b> | <b>3</b>  | <b>8</b> | <b>22</b> |
| <b>Langkah 4 :</b> | <b>Ulangi langkah 2 dan 3 .</b> |           |           |           |          |           |



## Iterasi 2

|            |                          |    |    |    |   |    |
|------------|--------------------------|----|----|----|---|----|
| Langkah 1: | 2                        | 10 | 15 | 3  | 8 | 22 |
| Langkah 2: | 2                        | 10 | 15 | 3  | 8 | 22 |
| Langkah 3: | 2                        | 3  | 15 | 10 | 8 | 22 |
| Langkah 4: | Ulangi langkah 2 dan 3 . |    |    |    |   |    |

Lakukan Iterasi selanjutnya sampai iterasi ke-6



## **BUBBLE SORT**

Teknik Sort yg bekerja dgn menggunakan prinsip gelembung (*bubble*) udara yg akan bergerak naik ke atas secara satuper satu.

Prinsip Kerja dari *Bubble Sort* adalah :

1. Pengecekan mulai dari data ke-1 sampai data ke-n
2. Bandingkan data ke-n dengan data sebelumnya (n-1)
3. Jika lebih kecil maka pindahkan bilangan tersebut dengan bilangan yg ada didepannya ( sebelumnya ) satu persatu (n-1,n-2,n-3,....dst)
4. Jika lebih besar maka tidak terjadi pemindahan
5. Ulangi langkah 2 dan 3 s/d sort optimal.





**Contoh : 22 10 15 3 8 2**

**terasi 1**

|                   | <b>1</b>                      | <b>2</b>  | <b>3</b>  | <b>4</b> | <b>5</b> | <b>6</b> |
|-------------------|-------------------------------|-----------|-----------|----------|----------|----------|
| <b>Langkah 1:</b> | <b>22</b>                     | <b>10</b> | <b>15</b> | <b>3</b> | <b>8</b> | <b>2</b> |
| <b>Langkah 2:</b> | <b>22</b>                     | <b>10</b> | <b>15</b> | <b>3</b> | <b>8</b> | <b>2</b> |
| <b>Langkah 3:</b> | <b>22</b>                     | <b>10</b> | <b>15</b> | <b>3</b> | <b>2</b> | <b>8</b> |
| <b>Langkah 4:</b> | <b>Ulangi langkah 2 dan 3</b> |           |           |          |          |          |

**Hasil iterasi 1 : 2 22 10 15 3 8**



## Iterasi 2

Langkah 1:        2        22        10        15        3        8

Langkah 2:        2        22        10        15        3        8

-  $8 > 3$ , maka 8 tidak pindah,  
untuk selanjutnya bandingkan  
data sebelumnya yaitu 3.

Langkah 3:        2        22        10        3        15        8

Langkah 4:        Ulangi langkah 2 dan 3

Hasil Iterasi 2 : 2        3        22        10        15        8

Lakukan Iterasi selanjutnya sampai iterasi ke- 6



## QuickSort

Metode QuickSort sering disebut metode *partition exchange sort*, Diperkenalkan oleh C.A.R. Hoare. Pada metode ini jarak kedua elemen yang akan ditukarkan nilainya ditentukan cukup besar.

Misal ada  $N$  elemen dalam keadaan urut turun, adalah mungkin untuk mengurutkan  $N$  elemen tersebut dengan  $N/2$  kali, yakni pertama kali menukarkan elemen paling kiri dengan paling kanan, kemudian secara bertahap menuju ke elemen yang ada di tengah. Tetapi hal ini hanya bisa dilakukan jika kita tahu pasti bahwa urutannya adalah urut turun.

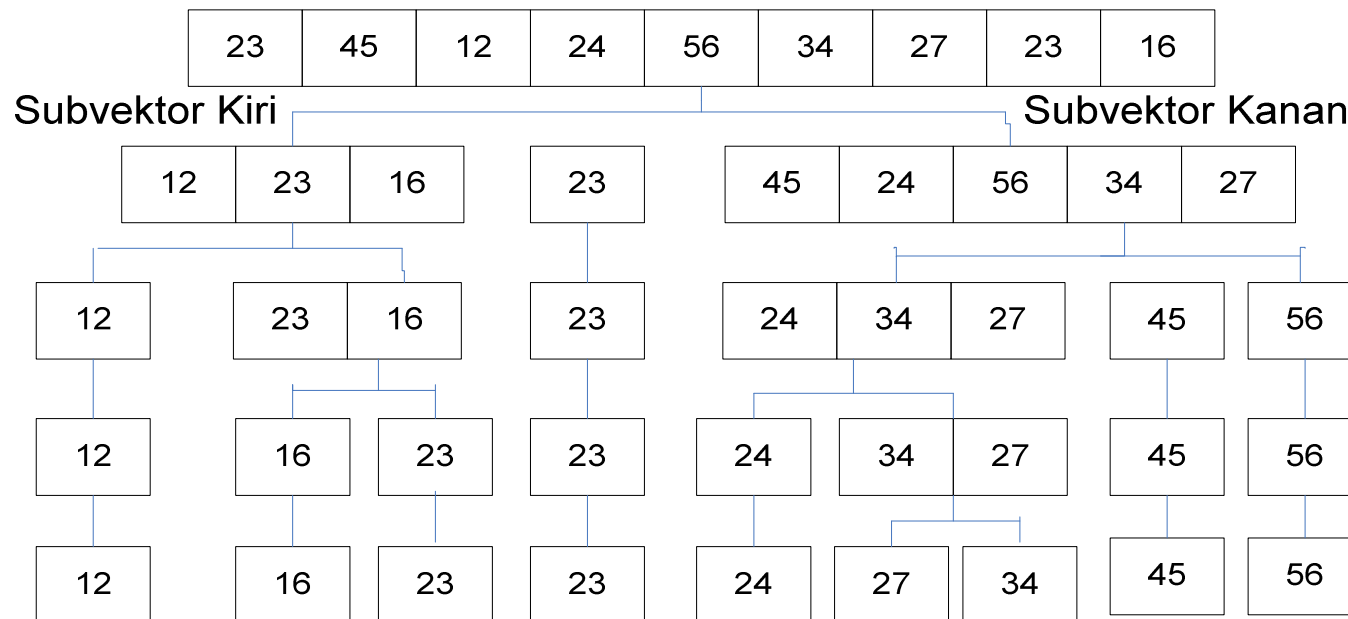


Secara garis besar metode ini dijelaskan sebagai berikut, misal kita akan mengurutkan vektor  $A$  yang mempunyai  $N$  elemen. Kita pilih sembarang dari vektor tersebut, biasanya elemen pertama misalnya  $X$ . kemudian semua elemen tersebut disusun dengan menempatkan  $X$  pada posisi  $J$  sedemikian rupa sehingga elemen ke 1 sampai ke  $j-1$  mempunyai nilai lebih kecil dari  $X$  dan elemen ke  $J+1$  sampai ke  $N$  mempunyai nilai lebih besar dari  $X$ . Dengan demikian kita mempunyai dua buah subvektor, subvektor pertama nilai elemennya lebih kecil dari  $X$ , subvektor kedua nilai elemennya lebih besar dari  $X$ .



Pada langkah berikutnya, proses diatas diulang pada kedua subvektor, sehingga kita akan mempunyai empat subvektor. Proses diatas diulang pada setiap subvektor sehingga seluruh vektor semua elemennya menjadi terurutkan.

Contoh:



Sumber: Struktur Data (Insap Santosa, 2006 hal: 337-340)



## ***INSERTION SORT***

Prinsip dasar Insertion adalah secara berulang-ulang menyisipkan / memasukkan setiap elemen. ke dlm posisinya / tempatnya yg benar.

- 1. Prinsip Kerja *Insertion Sort* adalah**
- 2. Pengecekan mulai dari data ke-1 sampai data ke-n**
- 3. Bandingkan data ke- $l$  (  $l =$  data ke-2 s/d data ke-n )**
- 4. Bandingkan data ke- $l$  tersebut dengan data sebelumnya ( $l-1$ ), Jika lebih kecil maka data tersebut dapat disisipkan ke data awal sesuai dgn posisisi yg seharusnya**
- 5. Lakukan langkah 2 dan 3 untuk bilangan berikutnya (  $l = l+1$  ) sampai didapatkan urutan yg optimal.**



**Contoh : 22 10 15 3 8 2**

**Iterasi 1**

**1 2 3 4 5 6**

**Langkah 1: 22 10 15 3 8 2**

**Langkah 2: 22 10 15 3 8 2**

**Langkah 3: 10 22 15 3 8 2**

**Langkah 4: Ulangi langkah 2 dan 3**



## Iterasi 2

Langkah 1:      10    22    15    3    8    2

Langkah 2:      10    22    15    3    8    2

Langkah 3:      10    15    22    3    8    2

Langkah 4: Ulangi langkah 2 dan 3

Lakukan Iterasi selanjutnya sampai iterasi ke- 6

Catatan : Setiap ada pemindahan, maka elemen yang sudah ada akan di insert sehingga akan bergeser kebelakang.





## **MERGE SORT**

**Prinsip Kerja Merge Sort adalah :**

- **Kelompokan deret bilangan kedalam 2 bagian, 4 bagian, 8 bagian, .....dst  $\rightarrow (2^n)$**
- **Urutkan secara langsung bilangan dalam kelompok tsb.**
- **Lakukan langkah diatas untuk kondisi bilangan yg lain sampai didapatkan urutan yg optimal .**



**Contoh : 22 10 15 3 8 2**

### Iterasi 1

|            | 1  | 2  | 3  | 4  | 5 | 6 |
|------------|----|----|----|----|---|---|
| Langkah 1: | 22 | 10 | 15 | 3  | 8 | 2 |
| Langkah 2: | 10 | 22 | 3  | 15 | 2 | 8 |

### Iterasi 2

|            |    |    |    |    |   |   |
|------------|----|----|----|----|---|---|
| Langkah 1: | 10 | 22 | 3  | 15 | 2 | 8 |
| Langkah 2: | 3  | 10 | 15 | 22 | 2 | 8 |



## Iterasi 3

Langkah 1:3      10      15      22      2      8

Langkah 2:2      3      8      10      15      22