

SORTING

Struktur Data

Definisi Sorting

- Sorting adalah proses pengurutan data yang sebelumnya disusun secara acak sehingga menjadi secara teratur menurut suatu aturan tertentu.

- 2 jenis pengurutan :

Ascensding (naik)

Descending (turun)

- Contoh :

Data acak : 10 15 3 8 2

Terurut Ascending : 2 3 8 10 15

Terurut Descending : 15 10 8 3 2

Metoda Sorting

- Buble / Exchange Sort
- Selection Sort
- Insertion Sort
- Quick Sort

Buble / Exchange Sort

- Membandingkan elemen yang sekarang dengan elemen berikutnya, jika elemen sekarang $>$ elemen berikutnya (untuk ascending), maka dilakukan proses penukaran.
- Proses sorting dapat dimulai dari data awal atau data akhir
- Contoh : ($JUMMAX - 1 = 6 - 1 = 5$)

IterasiKe	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
Awal	22	10	15	3	2	8
1.	10	22	15	3	2	8
	10	15	22	3	2	8
	10	15	3	22	2	8
	10	15	3	2	22	8
	10	15	3	2	22	8
2.	10	15	3	2	8	22
	10	15	3	2	8	22
	10	3	15	2	8	22
	10	3	2	15	8	22
	10	3	2	8	15	22
3.	3	10	2	8	15	22
	3	2	10	8	15	22
	3	2	8	10	15	22
	3	2	8	10	15	22
	3	2	8	10	15	22
4.	2	3	8	10	15	22
	2	3	8	10	15	22
	2	3	8	10	15	22
	2	3	8	10	15	22
	2	3	8	10	15	22
5.	2	3	8	10	15	22
	2	3	8	10	15	22
	2	3	8	10	15	22
	2	3	8	10	15	22
	2	3	8	10	15	22
Akhir	2	3	8	10	15	22

- Disini terlihat ketidak efisienan dari Buble Sort yaitu harus menyelesaikan $JUMMAX-1$ dari data
- Pada table diatas iterai ke-4 saja data sudah terurut dan harusnya pada saat itu proses sudah berhenti, tapi dengan buble sort proses harus dilakukan sampai looping selesai.

Selection Sort

- Cara kerja metode ini didasarkan pada pencarian elemen dengan nilai terkecil, kemudian dilakukan penukaran dengan elemen ke-I
- Contoh :

IterasiKe	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]
Awal	<u>22</u>	10	15	3	②	8
I=1 Lok=5	2	<u>10</u>	15	③	22	8
I=2 Lok=4	2	3	<u>15</u>	10	22	⑧
I=3 Lok=6	2	3	8	<u>10</u>	22	15
I=4 Lok=4	2	3	8	10	<u>22</u>	⑮
I=5 Lok=6	2	3	8	10	15	22
Akhir	2	3	8	10	15	22

Insertion Sort

- Pengurutan dilakukan dengan cara membandingkan data ke-I (dimana I dimulai dari data ke-2 sampai dengan data terakhir) dengan data berikutnya. Jika ditemukan data yang lebih kecil maka data tersebut disisipkan kedepan sesuai posisi yang seharusnya.
- Contoh :

- **LANGKAH 1**

i = 1	2	3	4	5	6
22	10	15	3	8	2
temp	Cek		Geser		
10	temp < 22		data ke-1	→	posisi 2
Temp menempati posisi ke-1					
10	22	15	3	8	2

- **LANGKAH 2**

i = 1	2	3	4	5	6
10	<u>22</u>	15	3	8	2
temp	Cek		Geser		
15	temp < 22		data ke-2 → posisi 3		
	temp > 10				
Temp menempati posisi ke-2					
10	15	22	3	8	2

- **LANGKAH 3**

i = 1	2	3	4	5	6
10	15	<u>22</u>	3	8	2
temp	Cek		Geser		
3	temp < 22		data ke-3 → posisi 4		
	temp < 15		data ke-2 → posisi 3		
	temp < 10		data ke-1 → posisi 2		
Temp menempati posisi ke-1					
3	10	15	22	8	2

- **LANGKAH 4**

i = 1	2	3	4	5	6
3	10	15	<u>22</u>	8	2
temp	Cek		Geser		
8	temp < 22	data ke-4 → posisi 5			
	temp < 15	data ke-3 → posisi 4			
	temp < 10	data ke-2 → posisi 3			
	temp > 3				
Temp menempati posisi ke-2					
3	8	10	15	22	2

- **LANGKAH 5**

i = 1	2	3	4	5	6
3	8	10	15	<u>22</u>	2
temp	Cek		Geser		
2	temp < 22	data ke-5 → posisi 6			
	temp < 15	data ke-4 → posisi 5			
	temp < 10	data ke-3 → posisi 4			
	temp < 8	data ke-2 → posisi 3			
	temp < 3	data ke-1 → posisi 2			
Temp menempati posisi ke-1					
2	3	8	10	15	22

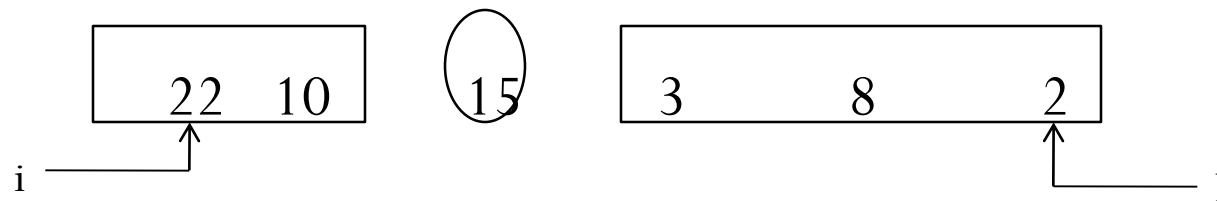
Quick Sort

- Membandingkan suatu elemen (disebut pivot) dengan elemen yang lain dan menyusunnya sedemikian rupa sehingga elemen-elemen yang lain lebih kecil dari pada pivot tersebut terletak di sebelah kiri dan elemen-elemen lain yang lebih besar daripada pivot tersebut terletak disebelah kanannya,
- Sehingga dengan demikian telah terbentuk dua sublist, yang terletak di sebelah kiri dan kanan dari pivot.
- Lalu pada sublist kiri dan sublist kanan kita anggap sebuah list baru dan kita kerjakan proses yang sama seperti sebelumnya.
- Demikian seterusnya sampai tidak terdapat sublist lagi. Sehingga didalamnya telah terjadi proses rekursif.

- Contoh :
- Bilangan yang didalam kurung merupakan pivot
- Persegi Panjang yang digambarkan dengan garis putus-putus menunjukan sublist
- i bergerak dari sudut kiri kekanan sampai mendapatkan nilai yang \geq pivot
- j bergerak dari sudut kanan kekiri sampai menemukan nilai yang $<$ pivot

- **LANGKAH 1**

- Index = 1 2 3 4 5 6

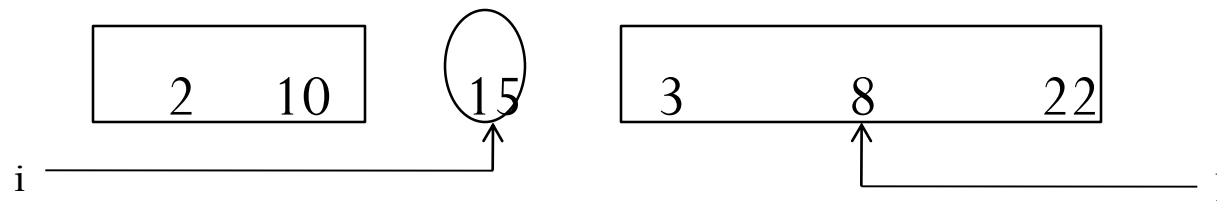


- i berhenti pada index ke-1 karena langsung mendapatkan nilai yang > dari pivot (15)
- j berhenti pada index ke-6 karena juga langsung mendapatkan nilai yang < dari pivot
- Karena $i > j$ maka data yang ditunjuk oleh i ditukar dengan data yang ditunjuk oleh j sehingga menjadi :

2 10 15 3 8 22

- **LANGKAH 2**

- Index = 1 2 3 4 5 6

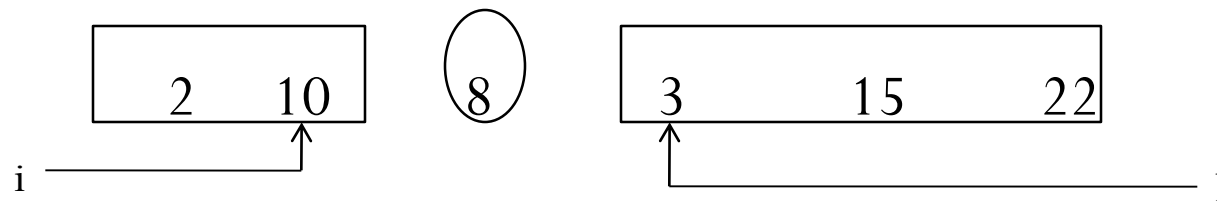


- i berhenti pada index ke-3 karena langsung mendapatkan nilai yang \geq dari pivot (15)
- j berhenti pada index ke-5 karena juga langsung mendapatkan nilai yang $<$ dari pivot
- Karena $i > j$ maka data yang ditunjuk oleh i ditukar dengan data yang ditunjuk oleh j sehingga menjadi :

2 10 8 3 15 22

- **LANGKAH 3**

- Index = 1 2 3 4 5 6



- i berhenti pada index ke-2 karena langsung mendapatkan nilai yang $>$ dari pivot (8)
- j berhenti pada index ke-4 karena juga langsung mendapatkan nilai yang $<$ dari pivot
- Karena $i > j$ maka data yang ditunjuk oleh i ditukar dengan data yang ditunjuk oleh j sehingga menjadi :

2 3 8 10 15 22

- **Latihan :**

Lakukan pengurutan menggunakan ke-4 metode diatas
untuk data : 9 2 10 4 7 20 3 8