

BAB 8 SORTIR

Pengurutan data (sorting) adalah suatu proses untuk menyusun kembali himpunan obyek menggunakan aturan tertentu.

Secara umum ada dua jenis pengurutan data yaitu :

- Pengurutan secara urut naik (Ascending)
yaitu dari data yang nilainya paling kecil sampai data yang nilainya paling besar.
- Pengurutan secara urut turun (Descending)
yaitu dari data yang mempunyai nilai yang paling besar sampai paling kecil.

Berdasarkan media yang digunakan terdapat 2 metode sortir :

- Sortir Internal
Metode ini dipakai jika himpunan data yang akan disortir kecil, sehingga proses sortir tidak membutuhkan tempat yang besar di memori utama komputer.
- Sortir Eksternal
Metode ini dipakai jika himpunan data yang akan disortir cukup besar, sehingga dibutuhkan media atau alat tambahan seperti Magnetik Tape, Disket dan sebagainya.

Dua hal yang mempengaruhi kecepatan algoritma sortir adalah :

- Jumlah operasi perbandingan yang dilakukan.
- Jumlah operasi pemindahan data dilakukan.

Pada garis besarnya ada tiga teknik utama yang dapat dilakukan dalam melakukan sortir yaitu :

- Sortir Penyisipan atau Insertion Sort.
- Sortir Pemilihan atau Selection Sort.
- Sortir Penukaran atau Exchange Sort.

Asumsi : Sortir secara Ascending

1. Sortir Penyisipan

Diketahui himpunan data :

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
44	55	12	42	94	18	7	67

$i = 2$ bandingkan $el(2)$ dengan $el(1)$ yaitu $55 > 44$ (tidak dilakukan pertukaran)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
44	55	12	42	94	18	7	67

di sini $a[1]$ dan $a[2]$ sudah terurut

$i = 3$ bandingkan $el(3)$ dengan $el(2)$ yaitu $12 < 55$ (dilakukan pertukaran)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
44	12	55	42	94	18	7	67

bandingkan $el(2)$ dengan $el(1)$ yaitu $12 < 44$ (dilakukan pertukaran)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
12	44	55	42	94	18	7	67

di sini $a[1]$, $a[2]$ dan $a[3]$ sudah terurut

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
12	44	55	42	94	18	7	67

$i = 4$ bandingkan $el(4)$ dengan $el(3)$ yaitu $42 < 55$ (dilakukan pertukaran)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
12	44	42	55	94	18	7	67

bandingkan $el(3)$ dengan $el(2)$ yaitu $42 < 44$ (dilakukan pertukaran)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
12	42	44	55	94	18	7	67

bandingkan el(2) dengan el(1) yaitu $42 > 12$ (tidak dilakukan pertukaran)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
12	42	44	55	94	18	7	67

di sini $a[1], \dots, a[4]$ sudah terurut.

$i = 5$	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	12	42	44	55	94	18	7	67

di sini $a[1], \dots, a[5]$ sudah terurut.

$i = 6$	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	12	18	42	44	55	94	7	67

di sini $a[1], \dots, a[6]$ sudah terurut.

$i = 7$	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	7	12	18	42	44	55	94	67

di sini $a[1], \dots, a[6]$ sudah terurut.

$i = 8$	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	7	12	18	42	44	55	67	94

di sini $a[1], \dots, a[8]$ sudah terurut.

Jadi pada setiap langkah ke i , $a[1], \dots, a[i]$ sudah terurut

2. Sortir Pemilihan

Algoritma :

1. Pilih data dengan key terkecil.
2. Tukarkan data tersebut dengan elemen $a[i]$

Diketahui himpunan data :

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
44	55	12	42	94	18	7	67

$i = 1, \text{Lok} = 7 \rightarrow$ el(7) ditukarkan dengan el(1)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	55	12	42	94	18	44	67

di sini $a[1]$ sudah terurut

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	55	12	42	94	18	44	67

$i = 2, \text{Lok} = 3 \rightarrow$ el(3) ditukarkan dengan el(2)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	55	42	94	18	44	67

di sini $a[1]$ dan $a[2]$ sudah terurut

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	55	42	94	18	44	67

$i = 3, \text{Lok} = 6 \rightarrow$ el(6) ditukarkan dengan el(3)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	94	55	44	67

di sini $a[1], \dots, a[3]$ sudah terurut

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	94	55	44	67

$i = 4, \text{Lok} = 4 \rightarrow$ el(4) tetap, karena yang menjadi tujuan adalah $a[4]$

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	94	55	44	67

di sini $a[1], \dots, a[4]$ sudah terurut

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	94	55	44	67

$i = 5, \text{Lok} = 7 \rightarrow$ el(7) ditukarkan dengan el(5)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	44	55	94	67

di sini $a[1], \dots, a[5]$ sudah terurut

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	44	55	94	67

$i = 6, \text{Lok} = 6 \rightarrow \text{el}(6)$ tetap, karena yang menjadi tujuan adalah $a[6]$

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	44	55	94	67

di sini $a[1], \dots, a[6]$ sudah terurut

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	44	55	94	67

$i = 7, \text{Lok} = 8 \rightarrow \text{el}(8)$ ditukarkan dengan $\text{el}(7)$

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	44	55	67	94

di sini $a[1], \dots, a[7]$ sudah terurut

Proses ini dilakukan sampai dengan langkah ke $I-1$
 Elemen yang sudah terurut sebagai berikut :

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	44	55	67	94

Perbedaan utama antara Sortir Penyisipan dan Sortir Pemilihan adalah sebagai berikut :

- Pada Sortir Penyisipan, pada setiap langkah hanya diperhatikan **satu data** saja, kemudian untuk mencari tempat data diletakkan, dilihat **semua data** yang akan menjadi tujuan.
- Sebaliknya pada Sortir Pemilihan, pada tiap langkah **dipilih data dari semua** barisan data, kemudian diletakkan sebagai **satu data** baru pada subdaftar tujuan.

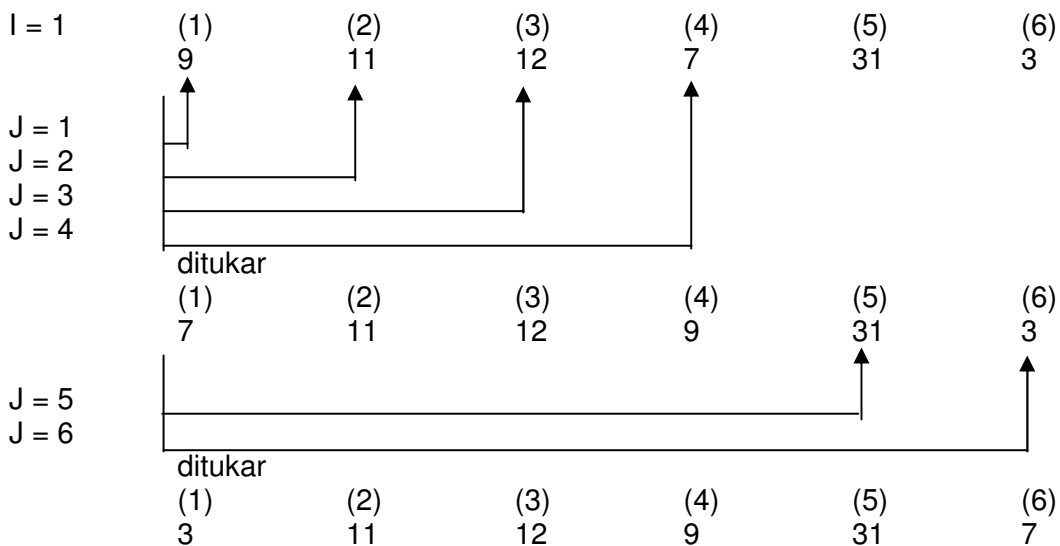
3. Sortir Penukaran

a. Sortir Gelembung (Bubble Sort)

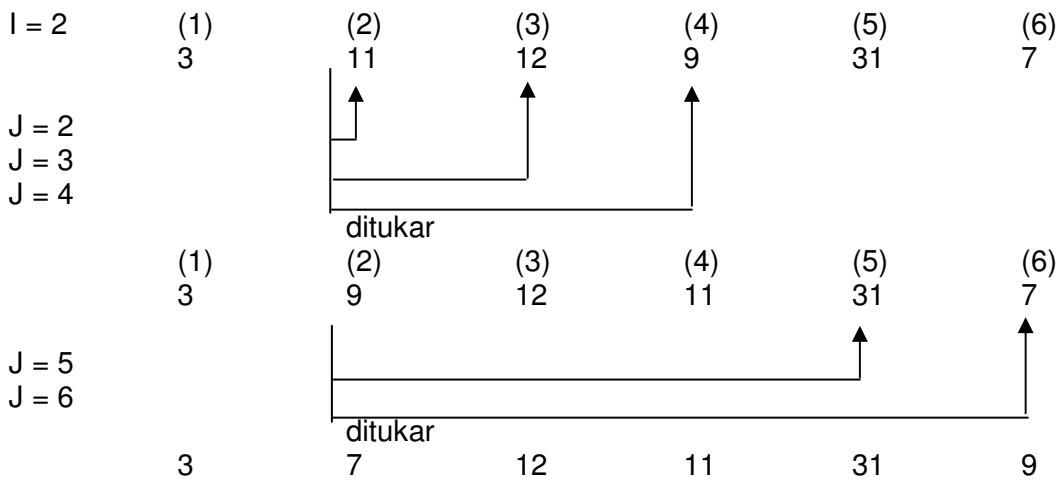
Secara umum, kelompok bilangan itu akan memiliki n bilangan. Dengan demikian, kita akan menemukan $n-1$ kali letak penyortiran. Letak pertama menggunakan indeks $I = 1$, letak kedua menggunakan indeks $I = 2$, dan seterusnya sampai ke letak ke- $(n - 1)$ yang menggunakan indeks $I = n - 1$.

Pada letak pertama, kita menggunakan indeks $J = 1, J = 2$ sampai ke $J = n$. Pada letak kedua, kita menggunakan indeks $J = 2, J = 3$ sampai ke $J = n$. Pada letak ketiga, kita menggunakan indeks $J = 3, J = 4$ sampai ke $J = n$, dan demikian seterusnya. Atau pada umumnya, nilai indeks J bergerak dari $J = 1$ sampai ke $J = n$.

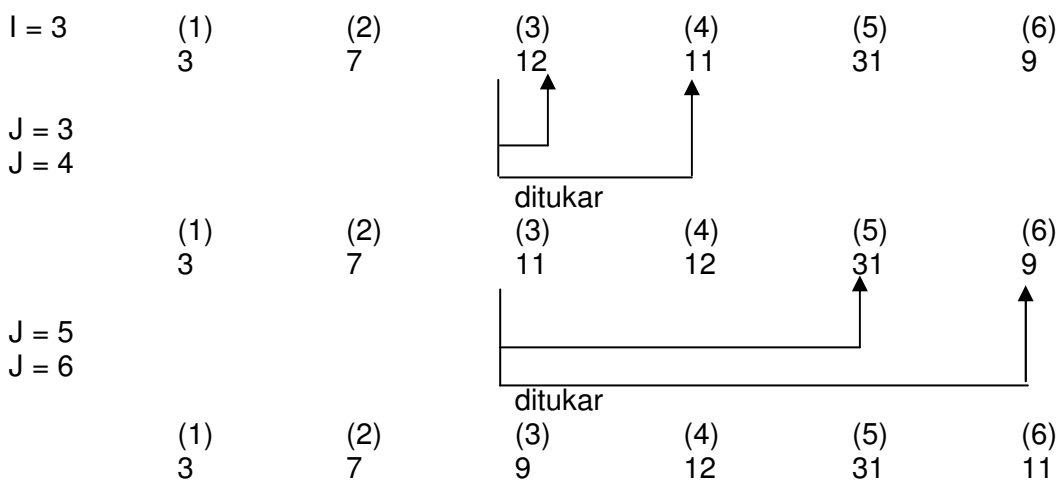
Penyortiran pada letak pertama



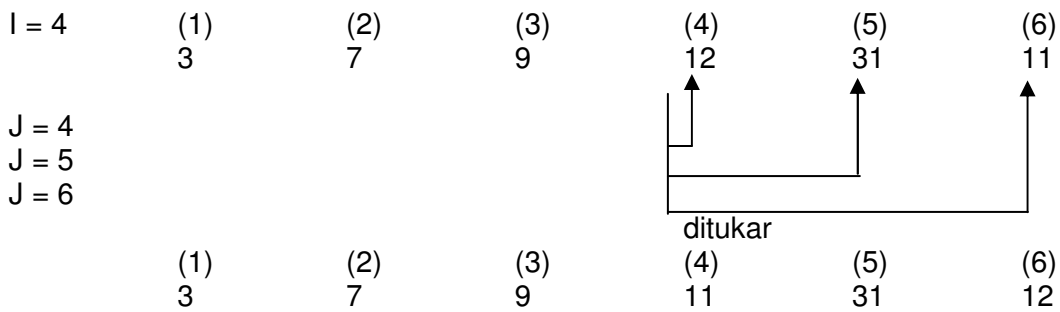
Penyortiran pada letak kedua



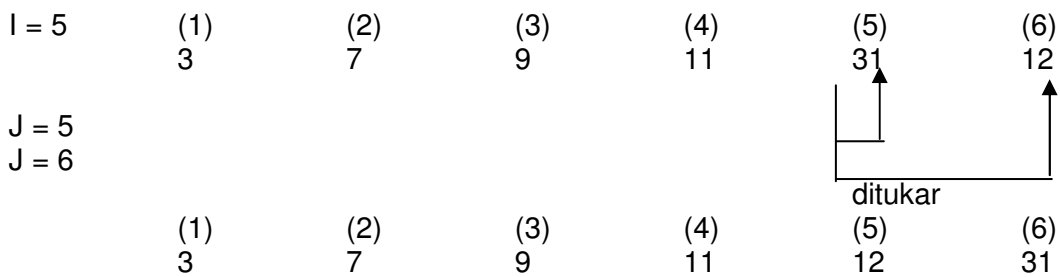
Penyortiran pada letak ketiga



Penyortiran pada letak keempat



Penyortiran pada letak kelima



b. Sortir Biasa (Common Sort)

Misalkan kita mempunyai n buah elemen yang belum terurut. Dalam sortir ini, kita mempunyai suatu indeks (l) yang menyatakan kedudukan elemen ($ke-l$) dari himpunan elemen, dan suatu panji (P) yang menandakan terjadi atau tidaknya pertukaran posisi elemen dalam himpunan data. Dalam keadaan awal, harga $l = 1$ dan $P = 0$, kemudian lakukan langkah sebagai berikut ini :

- Jika $el(l) < el(l+1)$, maka posisi $el(l)$ dibiarkan tetap. l bertambah 1, menjadi $l = 2$. Patokan kita sekarang adalah $el(l+1)$. $el(l+1)$ kita bandingkan dengan elemen berikutnya. Proses di atas dilakukan lagi sampai didapat elemen berikutnya yang $>$ dari $el(l+1)$. Pada saat itu dilakukan langkah b.
- Jika $el(l) > el(l+1)$, maka posisi $el(l)$ dan $el(l+1)$ dipertukarkan. Jika terjadi pertukaran seperti di atas, P berubah dari 0 menjadi 1 ($P = 1$). Langkah berikutnya adalah membandingkan $el(l+1)$ dengan elemen berikutnya. Jika $el(l+1) < el(l+2)$ maka kita lakukan langkah a kembali. Jika $el(l+1) > el(l+2)$ maka posisi $el(l+1)$ dan $el(l+2)$ dipertukarkan.
- Setelah mencapai elemen terakhir, jika $P = 0$ maka proses sortir selesai. Jika $P = 1$, maka proses sortir harus diulangi kembali, terhadap urutan yang tadi.

Demikian seterusnya kita lakukan langkah a dan b sampai dengan elemen ke n . Jika sampai dengan elemen ke n harga P masih sama dengan 1 ($P = 1$), maka sortir diulangi kembali sampai didapatkan $P = 0$. Pada saat pengulangan sortir, harga P dibuat menjadi 0 ($P = 0$) dan l dibuat menjadi 1 ($l = 1$)

Contoh :

Pandang 6 buah elemen yang belum terurut sebagai berikut :

$P = 0$	(1)	(2)	(3)	(4)	(5)	(6)
$l = 1$	7	11	12	3	31	9

Bandingkan $el(1)$ dan $el(2) \longrightarrow 7 < 11$ (tetap)

$P = 0$	(1)	(2)	(3)	(4)	(5)	(6)
$l = 2$	7	11	12	3	31	9

Bandingkan $el(2)$ dan $el(3) \longrightarrow 11 < 12$ (tetap)

$P = 0$	(1)	(2)	(3)	(4)	(5)	(6)
$l = 3$	7	11	12	3	31	9

Bandingkan $el(3)$ dan $el(4) \longrightarrow 12 > 3$ (tukar) dan harga $P = 1$

$P = 1$	(1)	(2)	(3)	(4)	(5)	(6)
$l = 4$	7	11	3	12	31	9

Bandingkan $el(4)$ dan $el(5) \longrightarrow 12 < 31$ (tetap)

$P = 1$	(1)	(2)	(3)	(4)	(5)	(6)
$l = 5$	7	11	3	12	31	9

Bandingkan $el(5)$ dan $el(6) \longrightarrow 31 > 9$ (tukar) dan harga $P = 1$

$P = 1$	(1)	(2)	(3)	(4)	(5)	(6)
	7	11	3	12	9	31

Karena harga $P = 1$, ulangi kembali proses sortir, dengan harga $P = 0$ dan $l = 1$

$P = 0$	(1)	(2)	(3)	(4)	(5)	(6)
$l = 1$	7	11	3	12	9	31

Bandingkan $el(1)$ dan $el(2) \longrightarrow 7 < 11$ (tetap)

$P = 0$	(1)	(2)	(3)	(4)	(5)	(6)
$l = 2$	7	11	3	12	9	31

Bandingkan $el(2)$ dan $el(3) \longrightarrow 11 > 3$ (tukar) dan harga $P = 1$

$P = 1$	(1)	(2)	(3)	(4)	(5)	(6)
$l = 3$	7	3	11	12	9	31

Bandingkan $el(3)$ dan $el(4) \longrightarrow 11 < 12$ (tetap)

P = 1	(1)	(2)	(3)	(4)	(5)	(6)
I = 4	7	3	11	12	9	31

Bandingkan el(4) dan el(5) → $12 > 9$ (tukar) dan harga P = 1

P = 1	(1)	(2)	(3)	(4)	(5)	(6)
I = 5	7	3	11	9	12	31

Bandingkan el(5) dan el(6) → $12 < 31$ (tetap)

P = 1	(1)	(2)	(3)	(4)	(5)	(6)
	7	3	11	9	12	31

Karena harga P = 1, ulangi kembali proses sortir, dengan harga P = 0 dan I = 1

P = 0	(1)	(2)	(3)	(4)	(5)	(6)
I = 1	7	3	11	9	12	31

Bandingkan el(1) dan el(2) → $7 > 3$ (tukar) dan harga P = 1

P = 1	(1)	(2)	(3)	(4)	(5)	(6)
I = 2	3	7	11	9	12	31

Bandingkan el(2) dan el(3) → $7 < 11$ (tetap)

P = 1	(1)	(2)	(3)	(4)	(5)	(6)
I = 3	3	7	11	9	12	31

Bandingkan el(3) dan el(4) → $11 > 9$ (tukar) dan harga P = 1

P = 1	(1)	(2)	(3)	(4)	(5)	(6)
I = 4	3	7	9	11	12	31

Bandingkan el(4) dan el(5) → $11 < 12$ (tetap)

P = 1	(1)	(2)	(3)	(4)	(5)	(6)
I = 5	3	7	9	11	12	31

Bandingkan el(5) dan el(6) → $12 < 31$ (tetap)

P = 1	(1)	(2)	(3)	(4)	(5)	(6)
	3	7	9	11	12	31

Kita lihat bahwa urutan elemen telah terurut dari kecil ke besar, tetapi harga P masih sama dengan 1. Jadi, kita lakukan sortir elemen di atas dengan harga P = 0 dan I = 1

P = 0	(1)	(2)	(3)	(4)	(5)	(6)
I = 1	3	7	9	11	12	31

Bandingkan el(1) dan el(2) → $3 < 7$ (tetap)

P = 0	(1)	(2)	(3)	(4)	(5)	(6)
I = 2	3	7	9	11	12	31

Bandingkan el(2) dan el(3) → $7 < 9$ (tetap)

P = 0	(1)	(2)	(3)	(4)	(5)	(6)
I = 3	3	7	9	11	12	31

Bandingkan el(3) dan el(4) → $9 < 11$ (tetap)

P = 0	(1)	(2)	(3)	(4)	(5)	(6)
I = 4	3	7	9	11	12	31

Bandingkan el(4) dan el(5) → $11 < 12$ (tetap)

P = 0	(1)	(2)	(3)	(4)	(5)	(6)
I = 5	3	7	9	11	12	31

Bandingkan el(5) dan el(6) → $12 < 31$ (tetap)

Karena harga P = 0 maka proses sortir selesai.

Jadi urutan elemen setelah disortir adalah sebagai berikut :

(1)	(2)	(3)	(4)	(5)	(6)
3	7	9	11	12	31

SHELLSORT : Mempercepat Sortir Penyisipan

Dalam hal ini, data dibagi dalam beberapa kelompok yang berbeda. Pada setiap kelompok dilakukan Sortir Penyisipan. Kemudian banyak kelompok diperkecil (diciutkan) sehingga banyak data dalam masing-masing kelompok bertambah. Lalu diberlakukan lagi Algoritma Sortir Penyisipan. Hal ini terus dilanjutkan sampai banyaknya kelompok yang ada hanya tinggal satu dan mengandung seluruh data.

Hal tersebut menunjukkan bahwa teknik ini lebih cepat dibandingkan dengan teknik Sortir Penyisipan biasa. Hal ini disebabkan karena pada tahap awal dilakukan beberapa sortir penyisipan dengan jumlah data sedikit. Karena jumlah kelompok masih cukup banyak. Pada akhir proses harus dilakukan sortir penyisipan pada seluruh data, tetapi sortir secara partial yang dilakukan sebelumnya sudah menyebabkan data terurut sebagian sehingga pada akhirnya jumlah operasi pada sortir penyisipan yang dilakukan tidak terlalu besar. Jadi pada teknik ini, sortir pada kelompok yang kecil akan sangat mempengaruhi kecepatan sortir pada kelompok berikutnya, yang mempunyai data sedikit lebih banyak.

Contoh :

Diketahui 8 buah data yang akan dibagi menjadi 4 kelompok, masing-masing dengan 2 data lalu dibagi ke dalam 2 kelompok data, masing-masing dengan 4 data, dan terakhir menjadi 1 kelompok dengan 8 data. Untuk setiap tingkat dan kelompok dilakukan Sortir penyisipan.

Keadaan awal

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	44	55	12	42	94	18	7	67
Kelompok 1 :	(1)	(2)				(1)	(2)	
	44	94				44	94	
Kelompok 2 :		55	18			18	55	
Kelompok 3 :		12	7			7	12	
Kelompok 4 :		42	67			42	67	

Hasilnya sekarang :

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
44	18	7	42	94	55	12	67

Sekarang gabungkan kelompok 1 dan 3 serta kelompok 2 dan 4. Kemudian lakukan Sortir Penyisipan Parsial :

	(1)	(2)	(3)	(4)		(1)	(2)	(3)	(4)
Kelompok 1 – 3 :	44	94	7	12	→	7	12	44	94
Kelompok 2 – 4 :	18	55	42	67	→	18	42	55	67

Hasilnya sekarang :

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	18	12	42	44	55	94	67

Sekarang gabungkan menjadi satu kelompok dan kita lakukan Sortir Penyisipan.

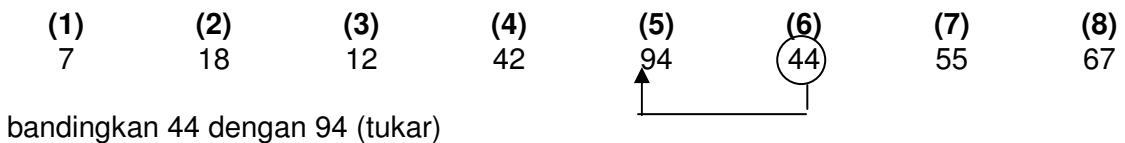
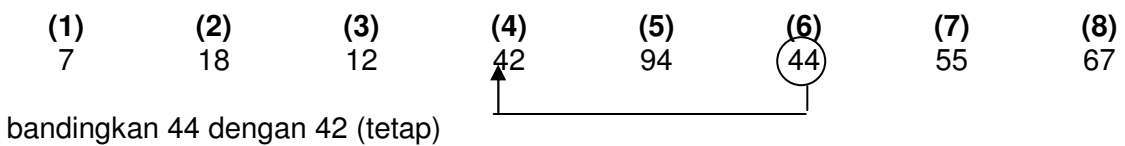
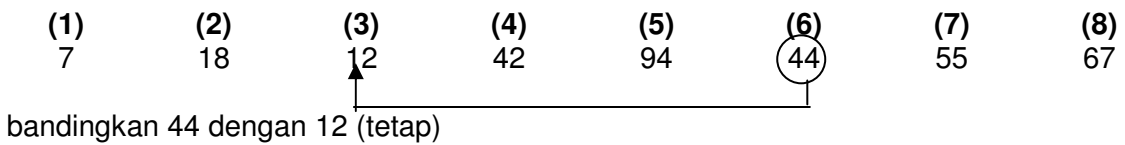
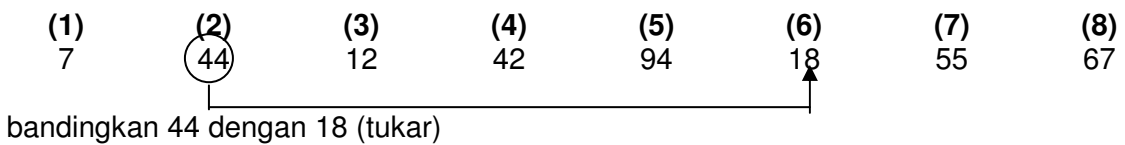
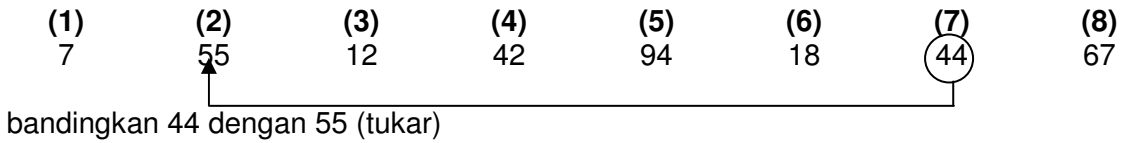
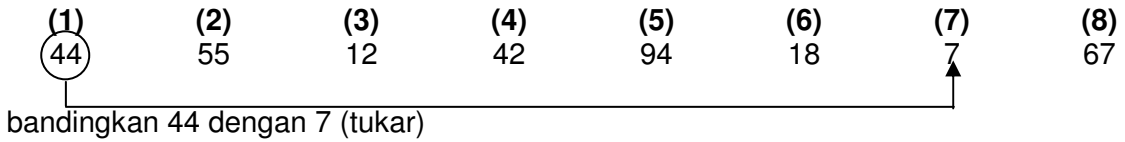
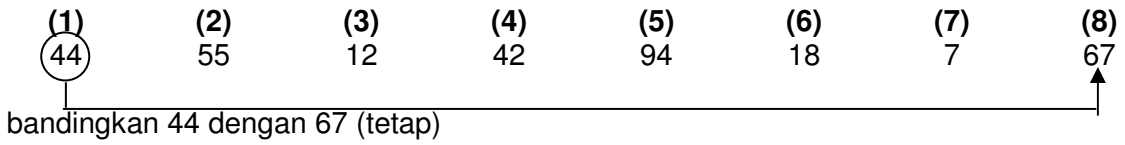
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	18	12	42	44	55	94	67

Diperoleh hasil akhir :

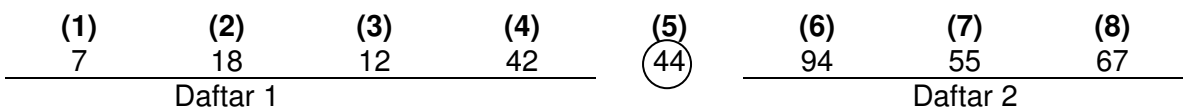
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
7	12	18	42	44	55	67	94

QUICKSORT

Quicksort adalah sebuah Algoritma Sortir dari model atau tipe Divide and Conquer, sama seperti metode Shellsort. :



Perhatikan, ternyata angka-angka dikiri 44, < 44 dan angka-angka di kanan 44, > 44. Semua angka yang < 44 membentuk daftar sendiri, demikian pula angka-angka yang > 44, seperti tampak di bawah ini



Jadi angka 44 pada posisi terakhir merupakan tempat yang tepat. Tahap reduksi di atas dapat diulang terhadap masing-masing daftar yang mengandung 2 atau lebih elemen. Hal ini diselesaikan dengan menggunakan 2 stack, yang disebut LOWER dan UPPER.

Lower : 1
Upper : 8