



# 08. Algoritma Pengurutan Bubble Sort dan Shell Sort

---

ARNA FARIZA  
YULIANA SETIOWATI

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Capaian Pembelajaran

---

- Mahasiswa mampu memahami algoritma pengurutan yaitu Bubble Sort dan Shell Sort
- Mahasiswa dapat mengimplementasikan fungsi algoritma pengurutan Bubble Sort dan Shell Sort



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Materi

---

- Algoritma Bubble Sort
- Algoritma Shell Sort



## Bubble Sort

---

- Algoritma pengurutan dengan dengan cara melakukan penukaran data dengan tepat disebelahnya secara terus menerus sampai bisa dipastikan dalam satu iterasi tertentu tidak ada lagi perubahan. Jika tidak ada perubahan berarti data sudah terurut.
- Disebut pengurutan gelembung karena masing-masing kunci akan dengan lambat menggelembung ke posisinya yang tepat.
- Disebut juga exchange sort.



## Algoritma Bubble Sort

---

1.  $i \leftarrow 1$
2. selama ( $i < N-1$ ) kerjakan baris 3 s/d 7
3.      $j \leftarrow N-1$
4.     selama ( $j \geq i$ ) kerjakan baris 5 s/d 6
5.         jika ( $\text{Data}[j-1] > \text{Data}[j]$ ) maka  
            tukar  $\text{Data}[j-1]$  dengan  $\text{Data}[j]$
6.          $j \leftarrow j - 1$  ;
7.      $i \leftarrow i + 1$  ;



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

---

$N = 6$

0	1	2	3	4	5
42	35	12	77	5	101

Terdapat 6 data yang akan diurutkan menggunakan Algoritma Bubble Sort.



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

Dimulai dari  $i=1$  karena  $i < (N-1)$ ,  $1 < 5$  maka proses looping 1 dilanjutkan.

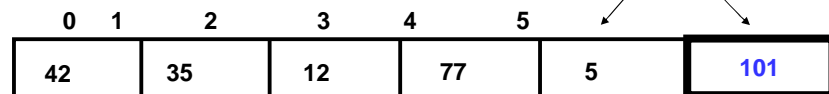
Tentukan  $j=N-1=5$ , sehingga  $j=5$ , selama  $j > i$  yaitu  $5 > 1$  maka proses looping 2 dilanjutkan.

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[4] > Data[5]$ ,  $5 > 101$  bernilai false, maka data tidak ditukar.

Selanjutnya  $j--$ , sehingga  $j = 4$

$i = 1$   $J = 5$   
 $5 > 101$  ? No Swap

No Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

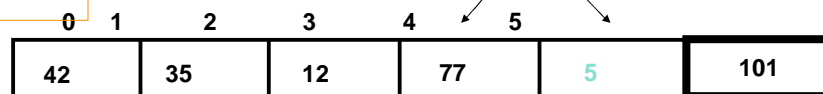
$i=1, j=4$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut,  $Data[3] > Data[4]$ ,  $77 > 5$  bernilai true, maka data ditukar.

Selanjutnya  $j--$ , sehingga  $j = 3$

$i = 1$   $J = 4$   
 $77 > 5$  ? Swap

Swap



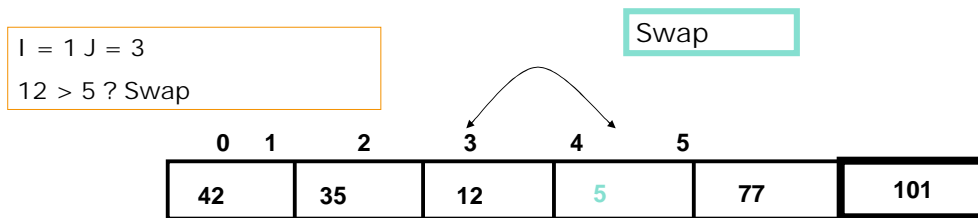
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

$i=1, j=3$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut,  $Data[2] > Data[3]$ , **12 > 5 bernilai true**, maka data ditukar.

Selanjutnya  $j--$ , sehingga  $j = 2$



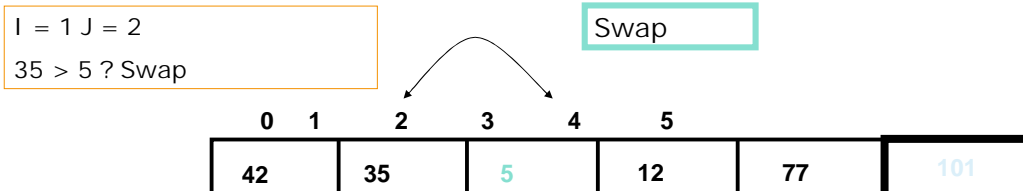
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

$i=1, j=2$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[1] > Data[2]$ , **35 > 5 bernilai true**, maka data ditukar.

Selanjutnya  $j--$ , sehingga  $j = 1$



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

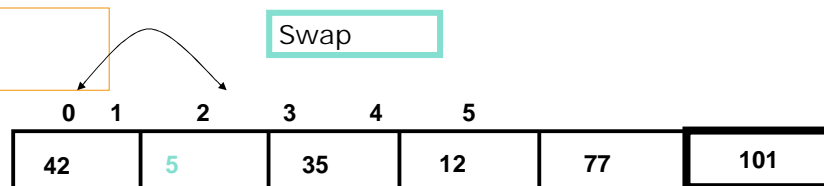
$i=1, j=1$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[0] > Data[1]$ ,  $42 > 5$  bernilai true, maka data ditukar.

Selanjutnya  $j--$ , sehingga  $j = 0$ , karena  $j >= i$ ,  $0 >= 1$  bernilai false maka proses looping 2 selesai.

Selanjutnya,  $i++$  menjadi  $i=2$

$i = 1$   $j = 1$   
 $42 > 5$  ? Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

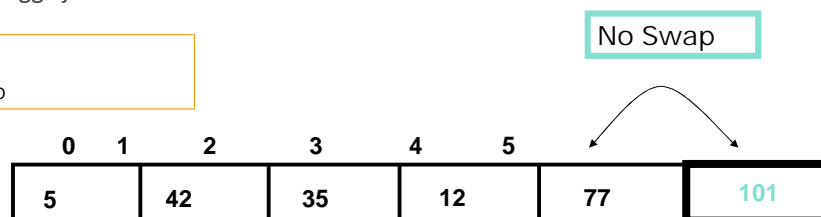
Dimulai  $i=2$  karena  $i < (N-1)$ ,  $2 < 5$ , maka proses looping 1 dilanjutkan.

Tentukan  $j=N-1=5$ , sehingga  $j=5$ , selama  $j >= i$  yaitu  $5 > 2$  maka proses looping 2 dilanjutkan.

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[4] > Data[5]$ ,  $77 < 101$  bernilai false, maka data tidak ditukar.

Selanjutnya  $j--$ , sehingga  $j = 4$

$i = 2$   $j = 5$   
 $77 > 101$  ? No Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

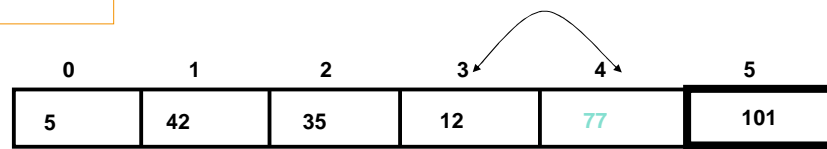
$i=2, j=4$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[3] > Data[4]$ , **12 > 77 bernilai false**, maka data tidak ditukar.

Selanjutnya  $j--$ , sehingga  $j = 3$

$i = 2, j = 4$   
 $12 > 77 ?$  No Swap

No Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

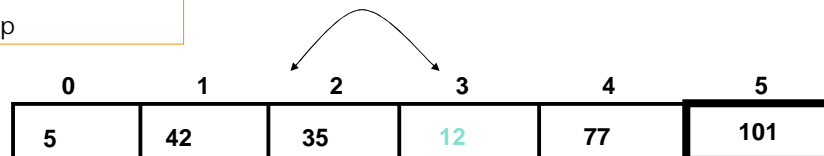
$i=2, j=3$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[2] > Data[3]$ , **35 > 12 bernilai true**, maka data ditukar.

Selanjutnya  $j--$ , sehingga  $j = 2$

$i = 2, j = 3$   
 $35 > 12 ?$  Swap

Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

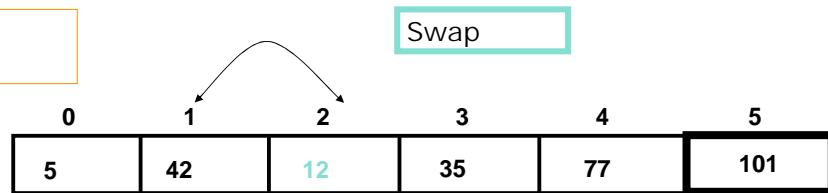
$i=2, j=2$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[1] > Data[2]$ , **42 > 12 bernilai true**, maka data ditukar.

Selanjutnya  $j--$ , sehingga  $j = 1$ , karena  $j \geq i$ ,  $1 \geq 2$  bernilai false maka proses looping 2 selesai.

Selanjutnya,  $i++$  menjadi  $i=3$

$i = 2, j = 2$   
 $42 > 12 ? \text{ Swap}$



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

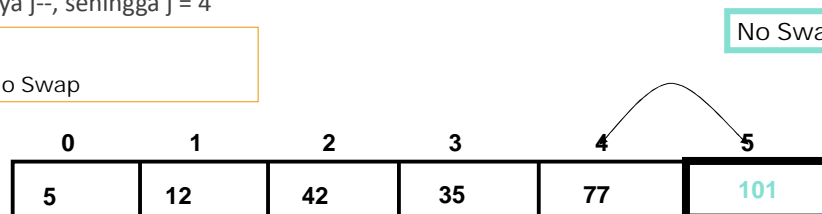
Dimulai  $i=3$  karena  $i < (N-1)$ ,  $3 < 5$ , maka proses looping 1 dilanjutkan.

Tentukan  $j=N-1=5$ , sehingga  $j=5$ , selama  $j \geq i$  yaitu  $5 \geq 3$  maka proses looping 2 dilanjutkan.

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[4] > Data[5]$ , **77 < 101 bernilai false**, maka data tidak ditukar.

Selanjutnya  $j--$ , sehingga  $j = 4$

$i = 3, j = 5$   
 $77 > 101 ? \text{ No Swap}$



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA PENS-ITS



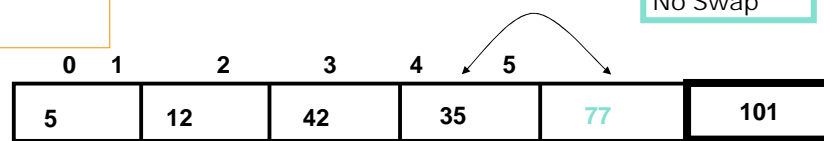
## Cara Kerja Algoritma Bubble Sort

$i=3, j=4$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[3] > Data[4]$ ,  $35 > 77$  bernilai **false**, maka data tidak ditukar.

Selanjutnya  $j--$ , sehingga  $j = 3$

$i = 3, j = 4$   
 $35 > 77 ?$  No Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA PENS-ITS

## Cara Kerja Algoritma Bubble Sort

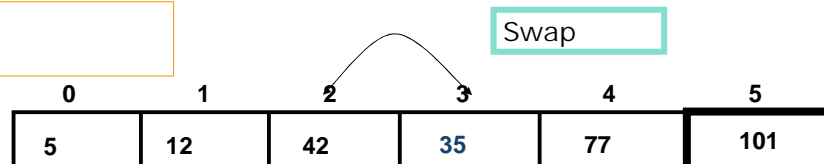
$i=3, j=3$

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[2] > Data[3]$ ,  $42 > 35$  bernilai **true**, maka data ditukar.

Selanjutnya  $j--$ , sehingga  $j = 2$ , karena  $j > i$ ,  $2 > 3$  bernilai **false** maka proses looping 2 selesai.

Selanjutnya,  $i++$  menjadi  $i=4$

$i = 3, j = 3$   
 $42 > 35 ?$  Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Cara Kerja Algoritma Bubble Sort

Dimulai  $i=4$  karena  $i < (N-1)$ ,  $4 < 5$ , maka proses looping 1 dilanjutkan.

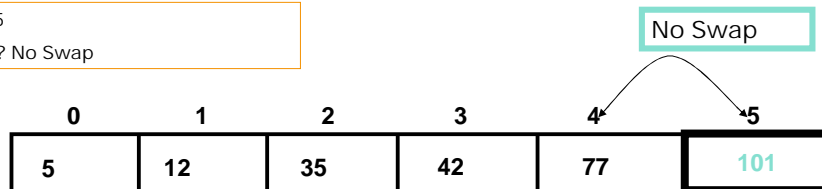
Tentukan  $j=N-1=5$ , sehingga  $j=5$ , selama  $j > i$  yaitu  $5 > 4$  maka proses looping 2 dilanjutkan.

Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[4] > Data[5]$ ,  $77 < 101$  bernilai false, maka data tidak ditukar.

Selanjutnya  $j--$ , sehingga  $j = 4$

$i = 4$   $j = 5$

$77 > 101$  ? No Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA PENS-ITS

## Cara Kerja Algoritma Bubble Sort

$i=4$ ,  $j=4$

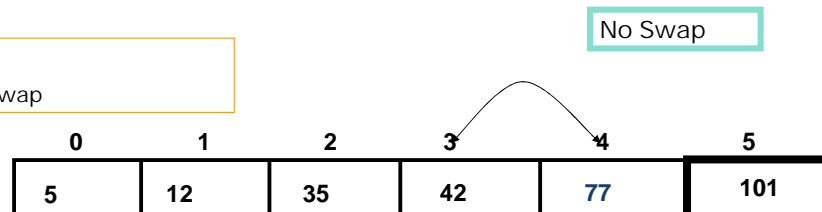
Jika  $Data[j-1] > Data[j]$  maka tukar data tersebut.  $Data[3] > Data[4]$ ,  $42 > 77$  bernilai false, maka data tidak ditukar.

Selanjutnya  $j--$ , sehingga  $j = 3$ , karena  $j > i$ ,  $3 > 4$  bernilai false maka proses looping 2 selesai.

Selanjutnya,  $i++$  menjadi  $i=5$ , karena  $i < N-1$ ,  $5 < 5$  bernilai false, maka proses looping 1 selesai

$i = 4$   $j = 4$

$42 > 77$  ? No Swap



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Analisa Bubble Sort

---

### BEST CASE:

- Array sudah dalam keadaan terurut naik
- Jumlah perbandingan key :  $n-1$
- Jumlah swap = 0
- Jumlah pergeseran : 0



## Analisa Bubble Sort

---

### ■ WORST CASE

- Array dalam urutan kebalikannya
- Jumlah perbandingan key :  
 $(1 + 2 + \dots + n-1) = n * (n-1) / 2$
- Jumlah swap =  $(1 + 2 + \dots + n-1) = n * (n-1) / 2$
- Jumlah pergeseran :  $3 * n * (n-1) / 2$





# Shell Sort

---

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Algoritma Shell Sort

---

Metode ini disebut juga dengan metode pertambahan menurun (*diminishing increment sort*).

Metode ini dikembangkan oleh Donald L. Shell pada tahun 1959, sehingga sering disebut dengan Metode Shell Sort.

Metode ini mengurutkan data dengan cara membandingkan suatu data dengan data lain yang memiliki jarak tertentu – sehingga membentuk sebuah sub-list-, kemudian dilakukan penukaran bila diperlukan



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Algoritma Shell Sort

---

Pertama-tama adalah menentukan jarak mula-mula dari data yang akan dibandingkan, yaitu  $N / 2$ . Data pertama dibandingkan dengan data dengan jarak  $N / 2$ . Apabila data pertama lebih besar dari data ke  $N / 2$  tersebut maka kedua data tersebut ditukar. Kemudian data kedua dibandingkan dengan jarak yang sama yaitu  $N / 2$ . Demikian seterusnya sampai seluruh data dibandingkan sehingga semua data ke- $j$  selalu lebih kecil daripada data ke- $(j + N / 2)$ .



## Algoritma Shell Sort

---

Pada proses berikutnya, digunakan jarak  $(N / 2) / 2$  atau  $N / 4$ . Data pertama dibandingkan dengan data dengan jarak  $N / 4$ . Apabila data pertama lebih besar dari data ke  $N / 4$  tersebut maka kedua data tersebut ditukar. Kemudian data kedua dibandingkan dengan jarak yang sama yaitu  $N / 4$ . Demikianlah seterusnya hingga seluruh data dibandingkan sehingga semua data ke- $j$  lebih kecil daripada data ke- $(j + N / 4)$ .

Pada proses berikutnya, digunakan jarak  $(N / 4) / 2$  atau  $N / 8$ . Demikian seterusnya sampai jarak yang digunakan adalah 1.



## Proses Shell Sort

---

Jarak yang dipakai didasarkan pada *increment value* atau *sequence number k*

Misalnya Sequence number yang dipakai adalah 5,3,1. Tidak ada pembuktian di sini bahwa bilangan-bilangan tersebut adalah sequence number terbaik

Setiap sub-list berisi setiap elemen ke- $k$  dari kumpulan elemen yang asli



## Proses Shell Sort

---

Contoh: Jika  $k = 5$  maka sub-list nya adalah sebagai berikut :

- s[0] s[5] s[10] ...
- s[1] s[6] s[11] ...
- s[2] s[7] s[12] ...
- dst

Begitu juga jika  $k = 3$  maka sub-list nya adalah:

- s[0] s[3] s[6] ...
- s[1] s[4] s[7] ...
- dst



## Proses Shell Sort

---

Buatlah sub-list yang didasarkan pada jarak (Sequence number) yang dipilih  
Urutkan masing-masing sub-list tersebut  
Gabungkan seluruh sub-list



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Proses Shell Sort

---

Urutkan sekumpulan elemen di bawah ini , misalnya diberikan sequence number : 5, 3, 1

30	62	53	42	17	97	91	38
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Proses Shell Sort

30 62 53 42 17 97 91 38

Step 1: Buat sub list  $k = 5$

S[0] S[5]

S[1] S[6]

S[2] S[7]

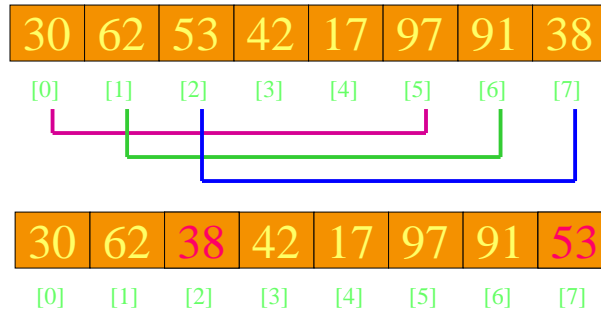
S[3]

S[0] < S[5] OK

S[1] < S[6] OK

S[2] > S[7] not OK.

Tukar



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Proses Shell Sort

30 62 53 42 17 97 91 38

Step 1: Buat sub list  $k = 3$

S[0] S[3] S[6]

S[1] S[4] S[7]

S[2] S[5]

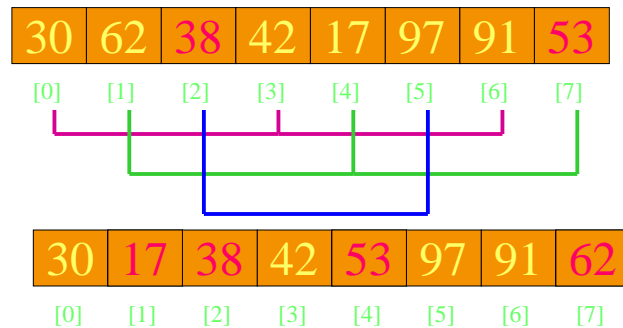
Step 2 - 3: Urutkan sub list & gabungkan

S[0] S[3] S[6] 30, 42, 91 OK

S[1] S[4] S[7] 62, 17, 53 not OK

SORT them 17, 53, 62

S[2] S[5] 38, 97 OK



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA



## Proses Shell Sort

30 62 53 42 17 97 91 38

Step 1: Buat sub list  $k=1$

S[0] S[1] S[2] S[3] S[4] S[5] S[6] S[7]

30	17	38	42	53	97	91	62
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]

Step 2 - 3: Urutkan sub list & gabungkan

Sorting akan seperti insertion sort

17	30	38	42	53	62	91	97
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]

### Kerjakan



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Algoritma Metode Shell Sort

1. jarak  $\leftarrow N$
2. selama (jarak  $> 1$ ) kerjakan 3-12
3.   jarak  $\leftarrow$  jarak / 2
4.   Sudah  $\leftarrow 1$
5.   selama Sudah = 1 kerjakan 6-12
6.     Sudah  $\leftarrow 0$
7.     j  $\leftarrow 0$
8.     selama (j  $< N$  - jarak) kerjakan 9-13
9.       i  $\leftarrow j$  + jarak
10.       if (Data[j]  $>$  Data[i])
11.         Tukar(&Data[j], &Data[i])
12.         Sudah = 1
13.       j++ ;



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Analisa Metode Shell Sort

---

Running time dari metode Shell Sort bergantung pada beratnya pemilihan sequence number.

Disarankan untuk memilih sequence number dimulai dari  $N/2$ , kemudian membaginya lagi dengan 2, seterusnya hingga mencapai 1.

Shell sort menggunakan 3 nested loop, untuk merepresentasikan sebuah pengembangan yang substansial terhadap metode insertion sort



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

### Pembandingan Running time (millisecond) antara insertion and Shell

---

N	insertion	Shellsort
1000	122	11
2000	483	26
4000	1936	61
8000	7950	153
16000	32560	358

Ref: Mark Allan Wiess  
(Florida International University)



POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

## Kesimpulan

---

Algoritma Bubble Sort adalah Algoritma pengurutan dengan dengan cara melakukan penukaran data dengan tepat disebelahnya secara terus menerus sampai bisa dipastikan dalam satu iterasi tertentu tidak ada lagi perubahan. Jika tidak ada perubahan berarti data sudah terurut.

Algoritma Shell Sort adalah Algoritma ini mengurutkan data dengan cara membandingkan suatu data dengan data lain yang memiliki jarak tertentu – sehingga membentuk sebuah sub-list-, kemudian dilakukan penukaran bila diperlukan



## Latihan Soal

---

- Urutkan data di bawah ini dengan Algoritma Bubble Sort dan Shell Sort, jelaskan pula langkah-langkahnya !
- **9 1 2 5 6 4**

