SORTING
“Insertion Sort dan Shell Sort”

TIM AJAR

ALGORITMA DAN STRUKTUR DATA

2022/2023

Capaian Pembelajaran

- Mahasiswa memahami algoritma Insertion Short
- Mahasiswa memahami penerapan algoritma Shell Short

Pokok Bahasan

- Insertion Sort
- Shell Sort

Insertion Sort

Insertion Sort : Deskripsi

- ✓ Insertion sort adalah sebuah metode pengurutan data dengan menempatkan setiap elemen data pada posisinya dengan cara melakukan perbandingan dengan data–data yang ada
- ✓ Salah satu algoritma sorting paling dasar
- ✓ Sederhana, efisien (terutama untuk jumlah data yang kecil)
- ✓ Adaptif (pada data yang hampir terurut, maka akan lebih cepat)

Insertion Sort : Deskripsi

- Bertujuan untuk menjadikan bagian sisi kiri array terurutkan sampai dengan seluruh array berhasil diurutkan.
- Insertion Sort bekerja seperti banyak orang yang sedang mengurutkan kartu di tangan.

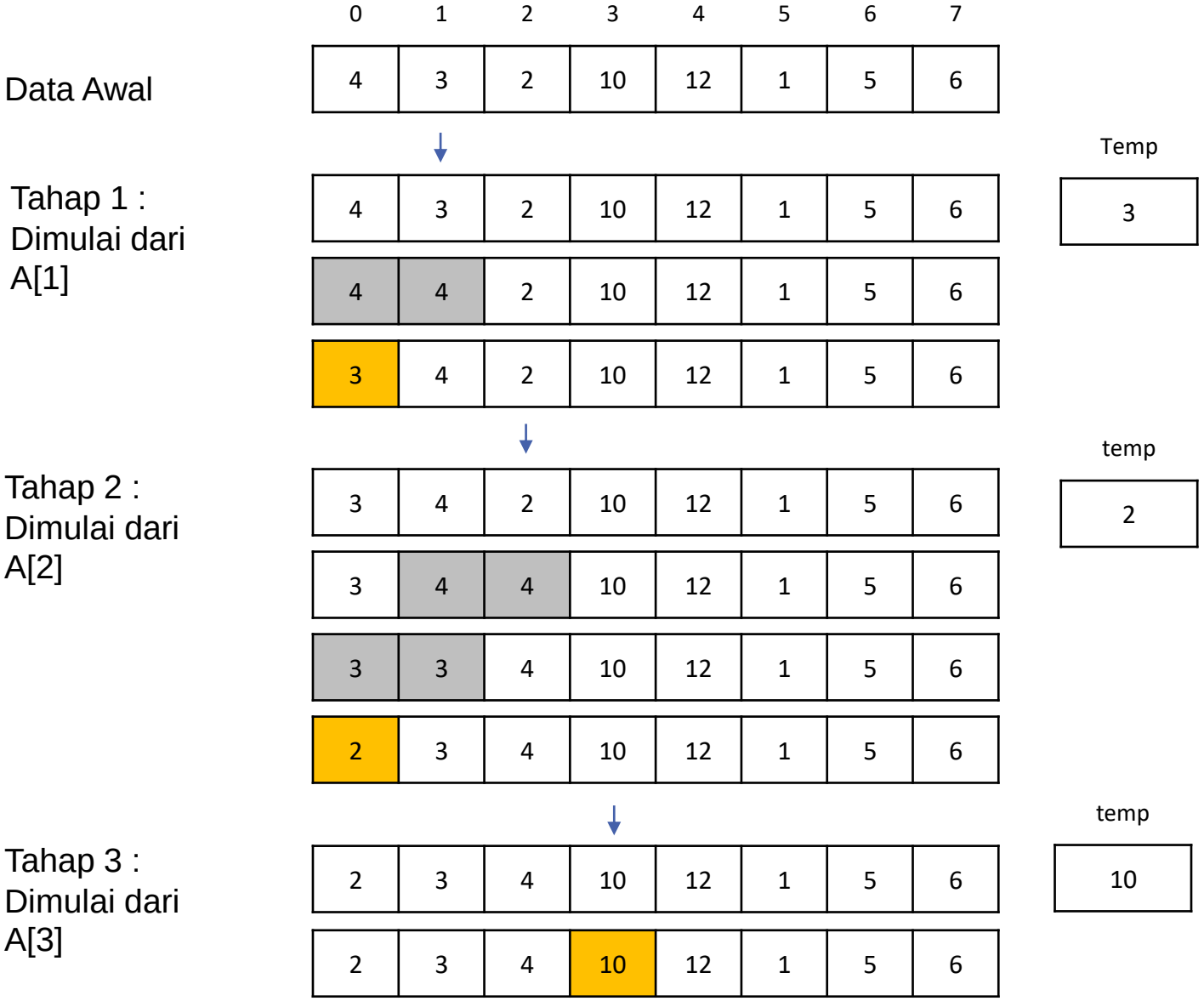
Insertion Sort

Metode Insertion Sort dilakukan dengan cara menyisipkan (insert) suatu data pada posisi yang seharusnya.

Langkah-langkah dalam proses ini untuk pengurutan *ascending*, dijabarkan sebagai berikut:

1. Ambil satu data ke-i, simpan nilai ke dalam temp (i dimulai dari 1)
2. Bandingkan nilai dari data temp dengan data yang ada di sebelah kiri posisi i.
3. Cek apakah data di sebelah kiri lebih besar dari data temp.
4. Jika langkah nomor 3 bernilai benar, lakukan pergeseran data satu per-satu ke kanan, kemudian sisipkan (INSERT) data temp di bekas tempat nilai yang terakhir digeser.
5. Ulangi langkah 1 sampai dengan 4, sehingga nilai i sama dengan data terakhir.

Langkah – langkah Insertion Sort



Temp

3

Tahap 1 : Dimulai dari A[1]
Temp = A[1] = 3

4 lebih besar dari temp (3) maka geser 4 ke kanan karena sudah sampai ke kolom 0, maka sisipkan (insert) temp menggantikan 4

temp

2

Tahap 2 : Dimulai dari A[2]
Temp = A[2] = 2

4 lebih besar dari temp (2) maka geser 4 ke kanan
3 lebih besar dari temp (2) maka geser 3 ke kanan karena sudah sampai ke kolom 0, maka sisipkan (insert) temp menggantikan 3

temp

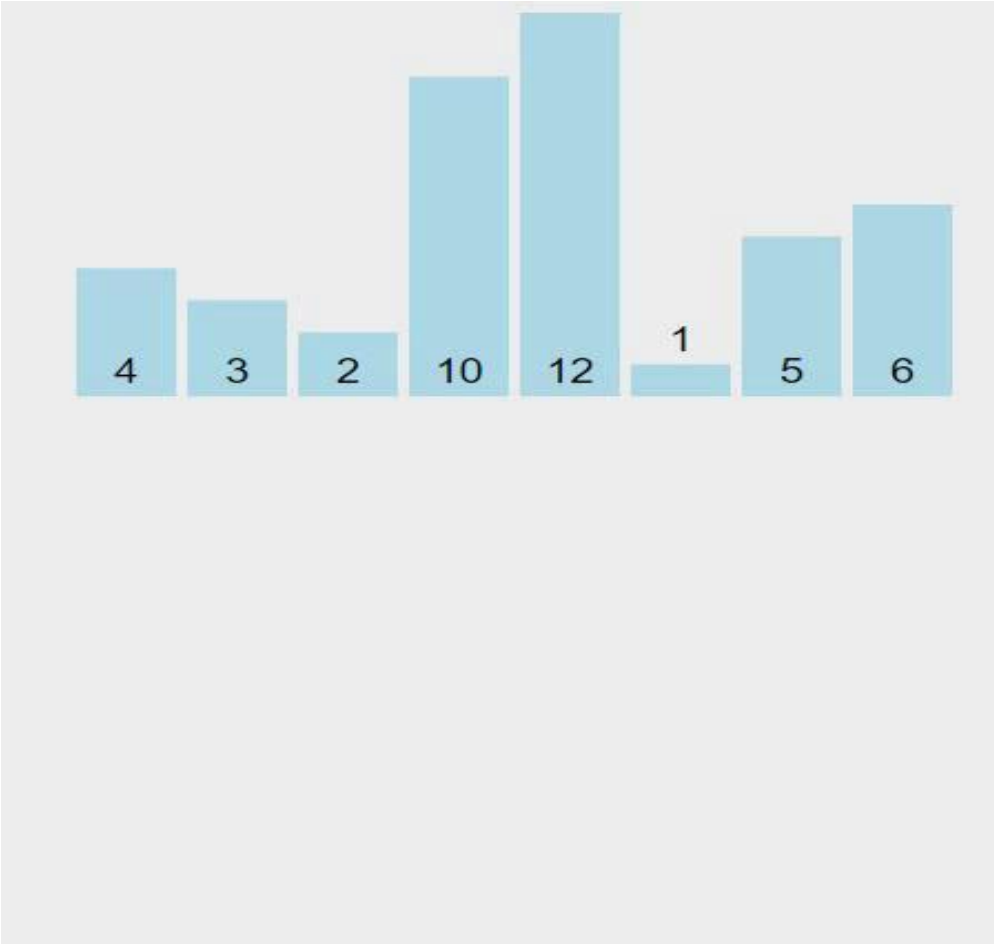
10

Tahap 3 : Dimulai dari A[3]
Temp = A[3] = 10
4 tidak lebih besar dari 10 maka proses selesai. Kemudian insert temp menggantikan 10 (untuk konsistensi algoritma)

Ilustrasi Insertion Sort (*Ascending*) - 1

Hasil tiap tahap dapat digambarkan sebagai berikut

Ilustrasi Insertion Sort (*Ascending*) - 2



Algoritma (Pseudo Code) Insertion Sort (*Ascending*)

```
Jika terdapat array data dengan panjang array n, index
terkecil 0, index terbesar n-1
for i = 1 to n-1
      temp = data[i]
      j = i-1
      while j>=0 and data[j]>temp do
            data[j+1] = data[j]
            j--
      endwhile
      data[j+1] = temp
endfor
```

Kelebihan Insertion Sort

1. Sederhana dalam penerapannya.
2. Mangkus dalam data yang kecil.
3. Jika list sudah terurut atau sebagian terurut maka Insertion Sort akan lebih cepat dibandingkan dengan Quicksort.
4. Mangkus dalam data yang sebagian sudah terurut.
5. Lebih mangkus dibanding Bubble Sort dan Selection Sort.
6. Loop dalam pada Insertion Sort sangat cepat, sehingga membuatnya salah satu algoritma pengurutan tercepat pada jumlah elemen yang sedikit.
7. Stabil.

Kekurangan Insertion Sort

1. Banyaknya operasi yang diperlukan dalam mencari posisi yang tepat untuk elemen larik.
2. Untuk larik yang jumlahnya besar ini tidak praktis.
3. Jika list terurut terbalik sehingga setiap eksekusi dari perintah harus memindai dan mengganti seluruh bagian sebelum menyisipkan elemen berikutnya.
4. Membutuhkan waktu yang lebih banyak pada kasus data yang tidak terurut, sehingga tidak cocok dalam pengurutan elemen dalam jumlah besar.

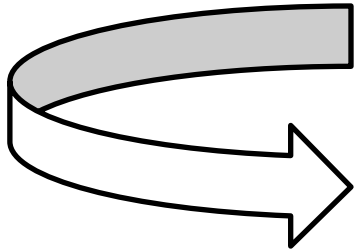
Shell Sort

Shell Sort: Deskripsi

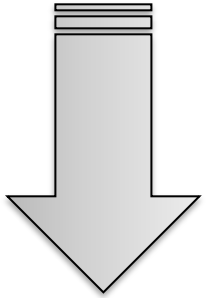
Metode ini mengurutkan data dengan cara membandingkan suatu data dengan data lain yang memiliki jarak tertentu sehingga membentuk sebuah sub-array, kemudian dilakukan penukaran bila diperlukan

Shell Sort : Deskripsi

TUJUAN



Meminimalkan jumlah Swap



1. **Stepping**
2. **Sorting**

Konsep Shell Sort

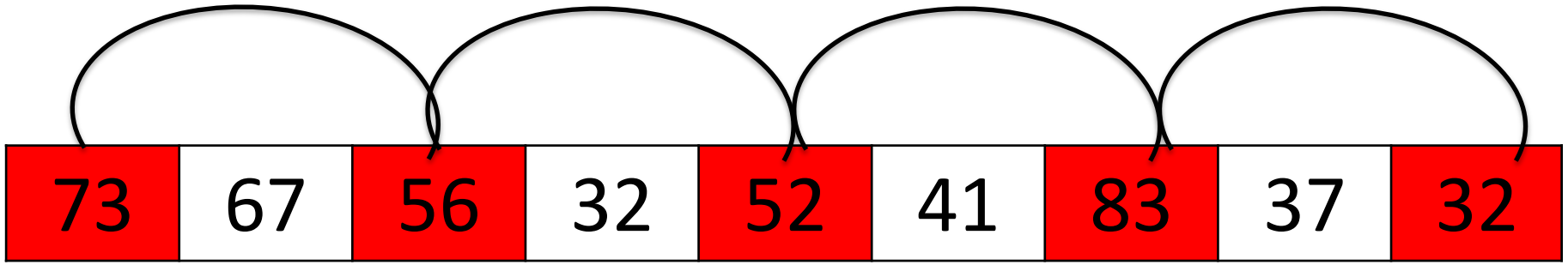
- ✓ Menghitung n (banyaknya data yang akan di-sorting)
- ✓ Menentukan stepping (lompatan) awal kemudian mensorting sub-arraynya
- ✓ Turunkan stepping, kemudian sort lagi

(Lakukan penurunan stepping sampai nilai stepping=1)

- ✓ **Stepping** adalah memecah array menjadi sub array dengan melakukan lompatan

Shell Sort : Stepping

Contoh Stepping=2



Shell Sort : Algoritma … (1)

Tahap 1

1. **Menentukan jarak mula-mula dari data** yang akan dibandingkan, yaitu **N / 2**.
2. Data pertama dibandingkan dengan data dengan jarak N / 2.
3. Apabila data pertama lebih besar dari data ke N / 2 tersebut maka kedua data tersebut ditukar.
4. Kemudian data kedua dibandingkan dengan jarak yang sama yaitu N / 2.
5. Demikian seterusnya sampai seluruh data dibandingkan sehingga semua data ke-j selalu lebih kecil daripada data ke-(j + N / 2).

Shell Sort : Algoritma … (2)

Tahap 2

6. Pada proses berikutnya, **digunakan jarak (N / 2) / 2** atau N / 4.
7. Data pertama dibandingkan dengan data dengan jarak N / 4.
8. Apabila data pertama lebih besar dari data ke N / 4 tersebut maka kedua data tersebut ditukar.
9. 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).

Shell Sort : Algoritma … (2)

Tahap 3

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

Algoritma Pseudocode

1. Jarak ← N
2. Selama (Jarak > 1) kerjakan baris 3 sampai dengan 9
3. Jarak ← Jarak / 2. Sudah ← false
4. Kerjakan baris 4 sampai dengan 8 selama Sudah = false
5. Sudah ← true
6. j ← 0
7. Selama (j < N – Jarak) kerjakan baris 8 dan 9
8. Jika (Data[j] > Data[j + Jarak]) maka tukar Data[j], Data[j + Jarak].
   Sudah ← true
9. j ← j + 1

Contoh 1: Proses Sorting Menggunakan Shell Sort …(1)

Iterasi	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
Awal	12	35	9	11	3	17	23	15	31	20
Jarak=5	12	35	9	11	3	17	23	15	31	20
i=6	12	**35**	9	11	3	17	**23**	15	31	20
Jarak=2	12	23	9	11	3	17	35	15	31	20
i=2	**12**	23	**9**	11	3	17	35	15	31	20
i=3	9	**23**	12	**11**	3	17	35	15	31	20
i=4	9	11	**12**	23	**3**	17	35	15	31	20
i=5	9	11	3	**23**	12	**17**	35	15	31	20
i=7	9	11	3	17	12	**23**	35	**15**	31	20
i=8	9	11	3	17	12	15	**35**	23	**31**	20
i=9	9	11	3	17	12	15	31	**23**	35	**20**
i=2	**9**	11	**3**	17	12	15	31	20	35	23
i=3	3	**11**	9	**17**	12	15	31	20	35	23
Jarak=1	3	11	9	15	12	17	31	20	35	23
i=2	3	**11**	**9**	15	12	17	31	20	35	23
i=4	3	9	11	**15**	**12**	17	31	20	35	23
i=7	3	9	11	12	15	17	**31**	**20**	35	23
i=9	3	9	11	12	15	17	20	31	**35**	**23**
i=1	**3**	**9**	11	12	15	17	20	31	23	35
i=8	3	9	11	12	15	17	20	**31**	**23**	35
i=9	3	9	11	12	15	17	20	23	**31**	**35**
Akhir	3	9	11	12	15	17	20	23	31	35

Contoh 1: Proses Sorting Menggunakan Shell Sort …(2)

- Pada saat Jarak = 5, j diulang dari 0 sampai dengan 4. Pada pengulangan pertama, Data[0] dibandingkan dengan Data[5]. Karena 12<17, maka tidak terjadi penukaran. Kemudian Data[1] dibandingkan dengan Data[6]. Karena 35>23 maka Data[1] ditukar dengan Data[6]. Demikian seterusnya sampai j=4.
- Pada saat Jarak = 5/2 = 2, j diulang dari 0 sampai dengan 7. Pada pengulangan pertama, Data[0] dibandingkan dengan Data[2]. Karena 12>9 maka Data[0] ditukar dengan Data[2]. Kemudian Data[1] dibandingkan dengan Data[3] juga terjadi penukaran karena 23>11. Demikian seterusnya sampai j=7. Perhatikan untuk Jarak = 2 proses pengulangan harus dilakukan lagi karena ternyata Data[0] > Data[2]. Proses pengulangan ini berhenti bila Sudah=true.
- Demikian seterusnya sampai Jarak=1.

Contoh 2: Shell Sort

Contoh Data

73	67	56	32	52	41	83	37	32	10

Contoh 2: Proses Shell Sort

Kelebihan Shell Sort

1. Operasi pertukarannya hanya dilakukan sekali saja.
2. Algoritma ini sangat rapat dan mudah untuk diimplementasikan.
3. Waktu pengurutan dapat lebih ditekan.
4. Mudah menggabungkannya kembali.

Kekurangan Shell Sort

1. Membutuhkan method tambahan
2. Sulit untuk membagi masalah.

Latihan

1. Gambarkan proses penyelesaian kasus pengurutan data menggunakan Insertion Sort untuk data = {2,35,14,27,67,19,23,46} secara Ascending
2. Gambarkan proses penyelesaian kasus pengurutan data menggunakan Shell Sort untuk data = {89,14,67,9,65,25,78,17} secara Descending!
3. Buatlah flowchart khusus proses pengurutan menggunakan Insertion Sort dan Shell Short secara Ascending!