

# Struktur Data dan Algoritma “Pengurutan Data”



Oleh:

Indra Hermawan

PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

# Tujuan

---

1. Menunjukkan beberapa algoritma dalam Pengurutan
2. Menunjukkan bahwa pengurutan merupakan suatu persoalan yang bisa diselesaikan dengan sejumlah algoritma yang berbeda satu sama lain lengkap dengan kelebihan dan kekurangannya
3. Dapat memilih algoritma yang paling sesuai untuk menyelesaikan suatu permasalahan pemrograman

# Pendahuluan

---

- Pengurutan data (*sorting*) didefinisikan sebagai suatu proses untuk menyusun kembali humpunan obyek menggunakan aturan tertentu.
- Menurut Microsoft Book-shelf, definisi algoritma pengurutan adalah algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen.

# Pendahuluan

---

- Ada dua macam urutan yang biasa digunakan dalam proses pengurutan yaitu
  - urut naik (*ascending*) yaitu dari data yang mempunyai nilai paling kecil sampai paling besar
  - urut turun (*descending*) yaitu data yang mempunyai nilai paling besar sampai paling kecil.

# Pendahuluan

---

Sebelum diurutkan:

5	10	1	4	3	7	8	9	2	6
---	----	---	---	---	---	---	---	---	---

Sesudah diurutkan:

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

# Pendahuluan

---

- Keuntungan dari data yang sudah dalam keadaan terurutkan antara lain :
  - data mudah dicari (misalnya dalam buku telepon atau kamus bahasa), mudah untuk dibetulkan, dihapus, disisipi atau digabungkan. Dalam keadaan terurutkan, kita mudah melakukan pengeeekan apakah ada data yang hilang
  - melakukan komppilasi program komputer jika tabel-tabel simbol harus dibentuk
  - mempercepat proses pencarian data yang harus dilakukan berulang kali.

# Pendahuluan

---

- Beberapa faktor yang berpengaruh pada efektifitas suatu algoritma pengurutan antara lain:
  - banyak data yang diurutkan
  - kapasitas pengingat apakah mampu menyimpan semua data yang kita miliki
  - tempat penyimpanan data, misalnya piringan, pita atau kartu, atau media penyimpan yang lain.

- Insertion Sort
- Selection Sort
- Binary Sort
- Bubble Sort
- Shell Sort
- Merge Shor
- Quick Short



- Algoritma penyisipan langsung dapat dituliskan sebagai berikut :
  1.  $i \leftarrow 1$
  2. selama ( $i < N$ ) kerjakan baris 3 sampai dengan 9
  3.  $x \leftarrow \text{Data}[i]$
  4.  $j \leftarrow i - 1$
  5. selama ( $x < \text{Data}[j]$ ) kerjakan baris 6 dan 7
  6.  $\text{Data}[j + 1] \leftarrow \text{Data}[j]$
  7.  $j \leftarrow j - 1$
  8.  $\text{Data}[j+1] \leftarrow x$
  9.  $i \leftarrow i + 1$

# Metode Penyisipan Langsung (*Straight Insertion Sort*)

**Tabel 6.1 Proses Pengurutan dengan Metode Penyisipan Langsung**

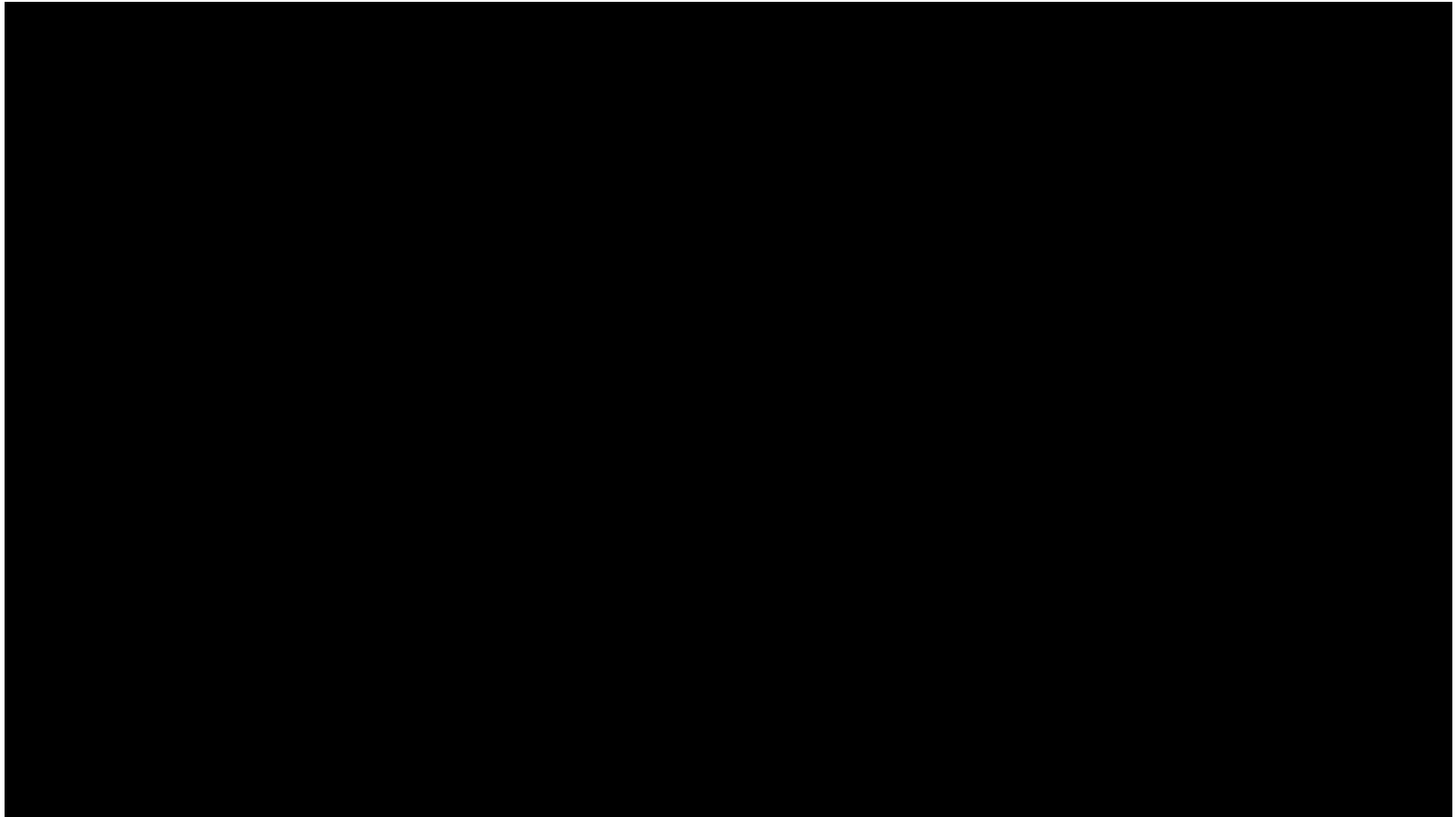
Iterasi	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
Awal	12	35	9	11	3	17	23	15	31	20
i=1	12	<b>35</b>	9	11	3	17	23	15	31	20
i=2	12	35	<b>9</b>	11	3	17	23	15	31	20
i=3	9	12	35	<b>11</b>	3	17	23	15	31	20
i=4	9	11	12	35	<b>3</b>	17	23	15	31	20
i=5	3	9	11	12	35	<b>17</b>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

- Proses pengurutan pada tabel 6.1 dapat dijelaskan sebagai berikut:
  - Pada saat  $i=1$ ,  $x$  sama dengan  $\text{Data}[1] = 35$  dan  $j=0$ . Karena  $\text{Data}[0] = 12$  dan  $35 > 12$  maka proses dilanjutkan untuk  $i=2$ .
  - Pada saat  $i=2$ ,  $x = \text{Data}[2] = 9$  dan  $j=1$ . Karena  $\text{Data}[1] = 35$  dan  $9 < 35$ , maka dilakukan pergeseran sampai ditemukan data yang lebih kecil dari 9. Hasil pergeseran ini,  $\text{Data}[1] = 12$  dan  $\text{Data}[2] = 35$  sedangkan  $\text{Data}[0] = x = 9$ .
  - Pada saat  $i=3$ ,  $x = \text{Data}[3] = 11$  dan  $j=3$ . Karena  $\text{Data}[2] = 35$  dan  $11 < 35$ , maka dilakukan pergeseran sampai ditemukan data yang lebih kecil dari 11. Hasil pergeseran ini,  $\text{Data}[2] = 12$  dan  $\text{Data}[3] = 35$  sedangkan  $\text{Data}[1] = x = 11$ .
  - Dan seterusnya.



# Metode Penyisipan Langsung (*Straight Insertion Sort*)

---



## Metode Penyisipan Biner (*Binary Insertion Sort*)

---

- Metode ini merupakan pengembangan dari metode penyisipan langsung.
- Dengan cara penyisipan langsung, perbandingan selalu dimulai dari elemen pertama (data ke-0), sehingga untuk menyisipkan elemen ke  $i$  kita harus melakukan perbandingan sebanyak  $i$  kali.

- Algoritma penyisipan biner dapat dituliskan sebagai berikut :

1  $i \leftarrow 1$

2 selama ( $i < N$ ) kerjakan baris 3 sampai dengan 14

3  $x \leftarrow \text{Data}[i]$

4  $l \leftarrow 0$

5  $r \leftarrow i - 1$

6 selama ( $l \leq r$ ) kerjakan baris 7 dan 8

7  $m \leftarrow (l + r) / 2$

8 jika ( $x < \text{Data}[m]$ ) maka  $r \leftarrow m - 1$ , jika tidak  $l \leftarrow m + 1$

9  $j \leftarrow i - 1$

10 selama ( $j \geq l$ ) kerjakan baris 11 dan 12

11  $\text{Data}[j+1] \leftarrow \text{Data}[j]$

12  $j \leftarrow j - 1$

13  $\text{Data}[l] \leftarrow x$

14  $i \leftarrow i + 1$

## Metode Seleksi (*Selection Sort*)

---

- Metode seleksi melakukan pengurutan dengan cara mencari data yang terkecil kemudian menukarkannya dengan data yang digunakan sebagai acuan atau sering dinamakan pivot.

## Metode Seleksi (*Selection Sort*)

---

- langkah pertama dicari data terkecil dari data pertama sampai data terakhir.
- Kemudian data terkecil ditukar dengan data pertama. Dengan demikian, data pertama sekarang mempunyai nilai paling kecil dibanding data yang lain.
- Langkah kedua, data terkecil kita cari mulai dari data kedua sampai terakhir. Data terkecil yang kita peroleh ditukar dengan data kedua dan demikian seterusnya sampai semua elemen dalam keadaan terurutkan.



## Metode Seleksi (*Selection Sort*)

- Algoritma seleksi dapat dituliskan sebagai berikut :

1  $i \leftarrow 0$

2 selama ( $i < N-1$ ) kerjakan baris 3 sampai dengan 9

3  $k \leftarrow i$

4  $j \leftarrow i + 1$

5 Selama ( $j < N$ ) kerjakan baris 6 dan 7

6 Jika ( $\text{Data}[k] > \text{Data}[j]$ ) maka  $k \leftarrow j$

7  $j \leftarrow j + 1$

8 Tukar  $\text{Data}[i]$  dengan  $\text{Data}[k]$

9  $i \leftarrow i + 1$

# Metode Seleksi (*Selection Sort*)

**Tabel 6.2 Proses Pengurutan dengan Metode Seleksi**

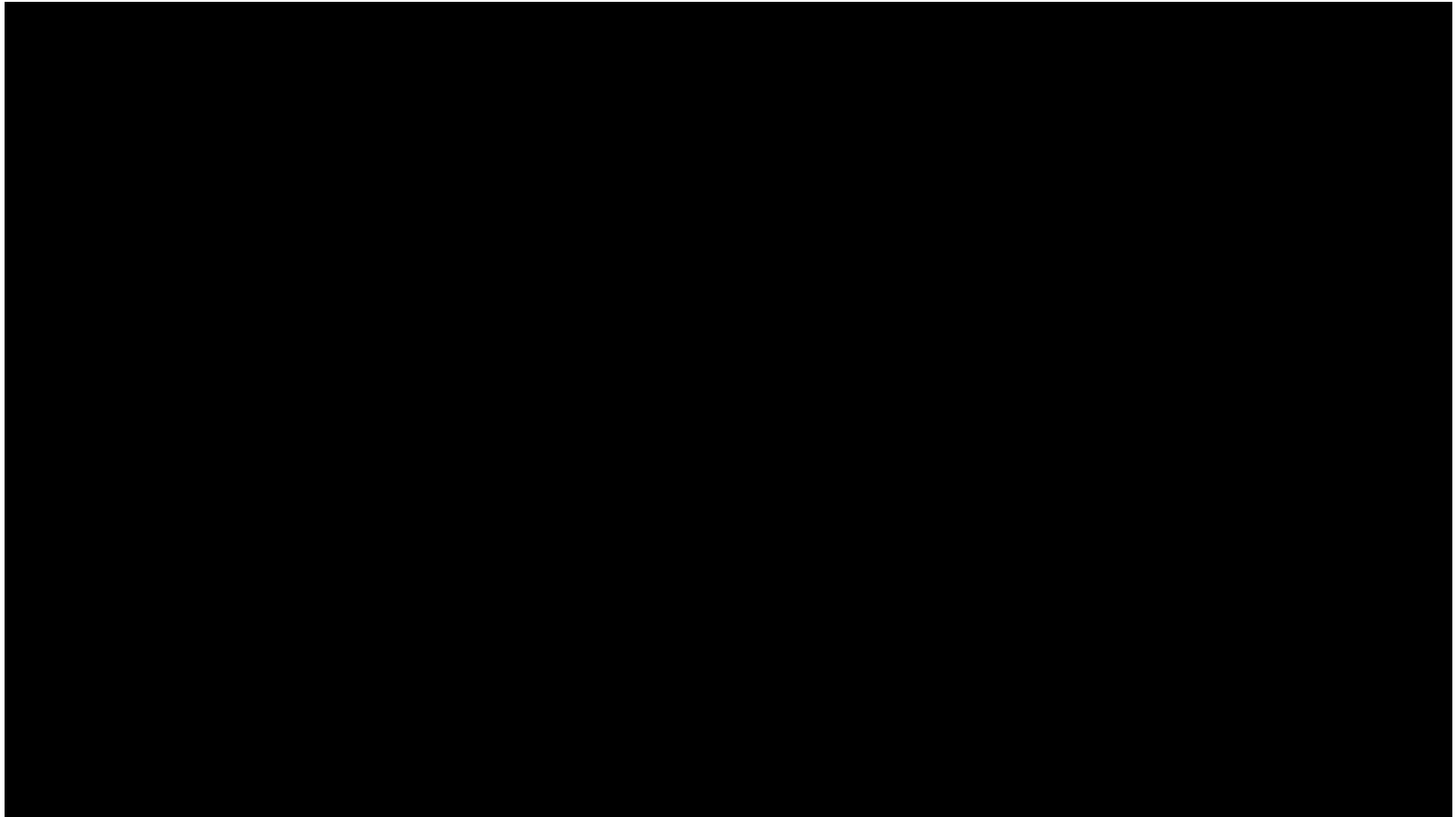
Iterasi	Data[0]	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
Awal	12	35	9	11	3	17	23	15	31	20
i=0	<b>12</b>	35	9	11	<b>3</b>	17	23	15	31	20
i=1	3	<b>35</b>	<b>9</b>	11	12	17	23	15	31	20
i=2	3	9	<b>35</b>	<b>11</b>	12	17	23	15	31	20
i=3	3	9	11	<b>35</b>	<b>12</b>	17	23	15	31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

## Metode Seleksi (*Selection Sort*)

- Proses pengurutan pada tabel 6.2 dapat dijelaskan sebagai berikut:
  - Pada saat  $i=0$ , data terkecil antara data ke-1 s/d ke-9 adalah data ke-4, yaitu 3, maka data ke-0 yaitu 12 ditukar dengan data ke-4 yaitu 3.
  - Pada saat  $i=1$ , data terkecil antara data ke-2 s/d ke-9 adalah data ke-2, yaitu 9, maka data ke-1 yaitu 35 ditukar dengan data ke-2 yaitu 9.
  - Pada saat  $i=2$ , data terkecil antara data ke-3 s/d ke-9 adalah data ke-3, yaitu 11, maka data ke-2 yaitu 35 ditukar dengan data ke-3 yaitu 11.
  - Pada saat  $i=3$ , data terkecil antara data ke-4 s/d ke-9 adalah data ke-4, yaitu 12, maka data ke-3 yaitu 35 ditukar dengan data ke-4 yaitu 12.
  - Dan seterusnya.

# Metode Seleksi (*Selection Sort*)

---



## Metode Gelembung (*Bubble sort*)

---

- Metode gelembung (*bubble sort*) sering juga disebut dengan metode penukaran (*exchange sort*) adalah metode yang mengurutkan data dengan cara membandingkan masing-masing elemen, kemudian melakukan penukaran bila perlu.

## Metode Gelembung (*Bubble sort*)

---

- Proses pengurutan metode gelembung ini menggunakan dua kalang.
- Kalang pertama melakukan pengulangan dari elemen ke 2 sampai dengan elemen ke  $N-1$  (misalnya variable  $i$ ), sedangkan kalang kedua melakukan pengulangan menurun dari elemen ke  $N$  sampai elemen ke  $i$  (misalnya variable  $j$ ).
- Pada setiap pengulangan, elemen ke  $j-1$  dibandingkan dengan elemen ke  $j$ . Apabila data ke  $j-1$  lebih besar daripada data ke  $j$ , dilakukan penukaran.

## Metode Gelembung (*Bubble sort*)

- Algoritma gelembung dapat dituliskan sebagai berikut :
  - 1  $i \leftarrow 0$
  - 2 selama ( $i < N-1$ ) kerjakan baris 3 sampai dengan 7
  - 3  $j \leftarrow N - 1$
  - 4 Selama ( $j \geq i$ ) kerjakan baris 5 sampai dengan 7
  - 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  $i \leftarrow i + 1$

# Metode Gelembung (*Bubble sort*)

**Tabel 6.3 Proses Pengurutan dengan Metode Gelembung**

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
i=1; j=9	12	35	9	11	3	17	23	15	<b>31</b>	<b>20</b>
j=8	12	35	9	11	3	17	23	<b>15</b>	<b>20</b>	31
j=7	12	35	9	11	3	17	<b>23</b>	<b>15</b>	20	31
j=6	12	35	9	11	3	<b>17</b>	<b>15</b>	23	20	31
j=5	12	35	9	11	<b>3</b>	<b>15</b>	17	23	20	31
j=4	12	35	9	<b>11</b>	<b>3</b>	15	17	23	20	31
j=3	12	35	<b>9</b>	<b>3</b>	11	15	17	23	20	31
j=2	12	<b>35</b>	<b>3</b>	9	11	15	17	23	20	31
j=1	<b>12</b>	<b>3</b>	35	9	11	15	17	23	20	31
i=2; j=9	3	12	35	9	11	15	17	23	<b>20</b>	<b>31</b>
j=8	3	12	35	9	11	15	17	<b>23</b>	<b>20</b>	31
j=7	3	12	35	9	11	15	<b>17</b>	<b>20</b>	23	31



# Metode Gelembung (*Bubble sort*)

j=6	3	12	35	9	11	<b>15</b>	<b>17</b>	20	23	31
j=5	3	12	35	9	<b>11</b>	<b>15</b>	17	20	23	31
j=4	3	12	35	<b>9</b>	<b>11</b>	15	17	20	23	31
j=3	3	12	<b>35</b>	<b>9</b>	11	15	17	20	23	31
j=2	3	<b>12</b>	<b>9</b>	35	11	15	17	20	23	31
<hr/>										
i=3; j=9	3	9	12	35	11	15	17	20	<b>23</b>	<b>31</b>
j=8	3	9	12	35	11	15	17	<b>20</b>	<b>23</b>	31
j=7	3	9	12	35	11	15	<b>17</b>	<b>20</b>	23	31
j=6	3	9	12	35	11	<b>15</b>	<b>17</b>	20	23	31
j=5	3	9	12	35	<b>11</b>	<b>15</b>	17	20	23	31
j=4	3	9	12	<b>35</b>	<b>11</b>	15	17	20	23	31
j=3	3	9	<b>12</b>	<b>11</b>	35	15	17	20	23	31
<hr/>										
i=4; j=9	3	9	11	12	35	15	17	20	<b>23</b>	<b>31</b>
j=8	3	9	11	12	35	15	17	<b>20</b>	<b>23</b>	31
j=7	3	9	11	12	35	15	<b>17</b>	<b>20</b>	23	31
j=6	3	9	11	12	35	<b>15</b>	<b>17</b>	20	23	31
j=5	3	9	11	12	<b>35</b>	<b>15</b>	17	20	23	31
j=4	3	9	11	<b>12</b>	<b>15</b>	35	17	20	23	31

# Metode Gelembung (*Bubble sort*)

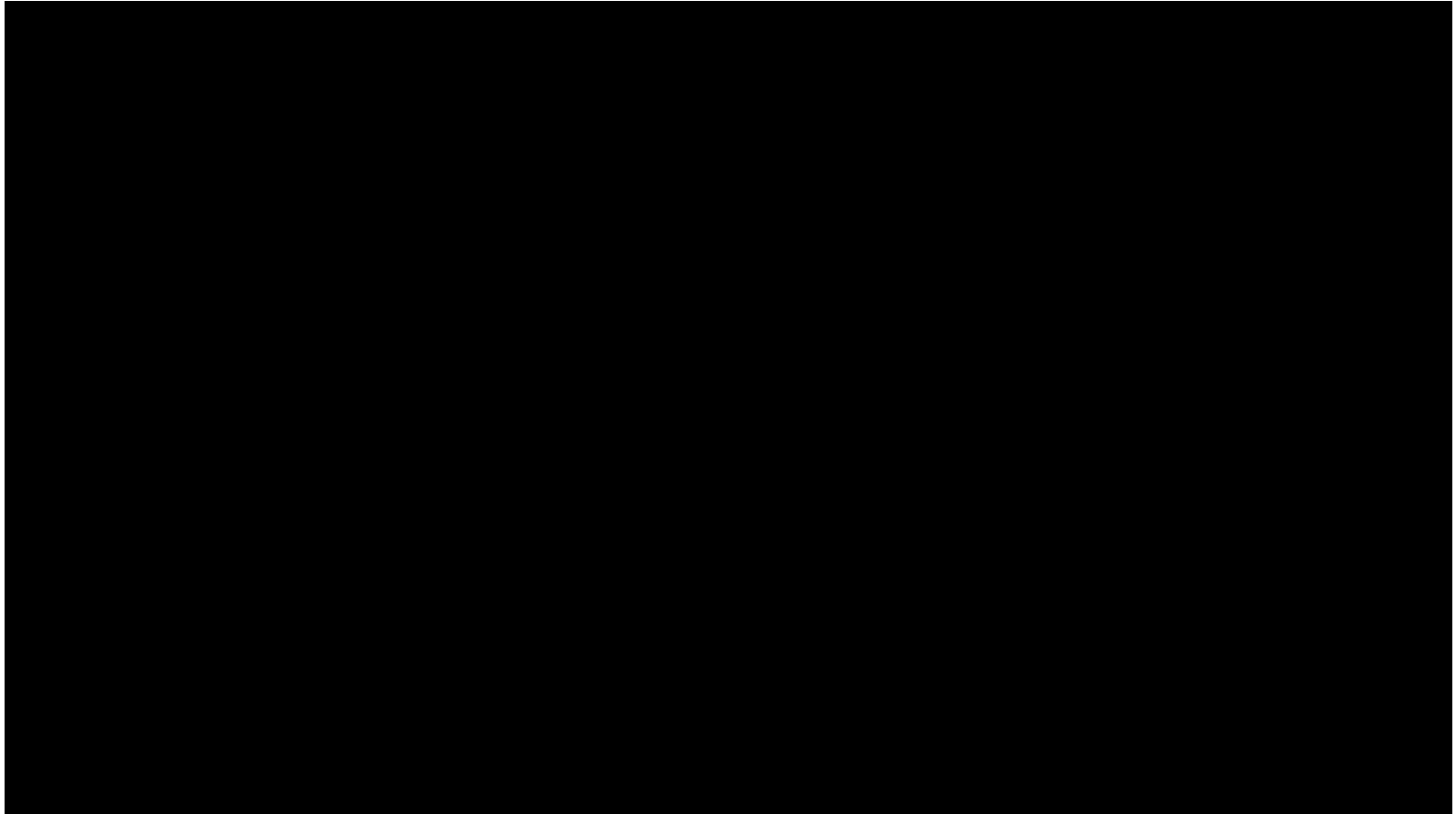
i=5; j=9	3	9	11	12	15	35	17	20	<b>23</b>	<b>31</b>
j=8	3	9	11	12	15	35	17	<b>20</b>	<b>23</b>	31
j=7	3	9	11	12	15	35	<b>17</b>	<b>20</b>	23	31
j=6	3	9	11	12	15	<b>35</b>	<b>17</b>	20	23	31
j=5	3	9	11	12	<b>15</b>	<b>17</b>	35	20	23	31
i=6; j=9	3	9	11	12	15	17	35	20	<b>23</b>	<b>31</b>
j=8	3	9	11	12	15	17	35	<b>20</b>	<b>23</b>	31
j=7	3	9	11	12	15	17	<b>35</b>	<b>20</b>	23	31
j=6	3	9	11	12	15	<b>17</b>	<b>20</b>	35	23	31
i=7; j=9	3	9	11	12	15	17	20	35	<b>23</b>	<b>31</b>
j=8	3	9	11	12	15	17	20	<b>35</b>	<b>23</b>	31
j=7	3	9	11	12	15	17	<b>20</b>	<b>23</b>	35	31
i=8; j=9	3	9	11	12	15	17	20	23	<b>35</b>	<b>31</b>
j=8	3	9	11	12	15	17	20	23	31	35
Akhir	3	9	11	12	15	17	20	23	31	35

## Metode Gelembung (*Bubble sort*)

- Proses pengurutan pada tabel 6.3 dapat dijelaskan sebagai berikut:
  - Pada saat  $i=1$ , nilai  $j$  diulang dari 9 sampai dengan 1. Pada pengulangan pertama  $Data[9]$  dibandingkan  $Data[8]$ , karena  $20 < 31$  maka  $Data[9]$  dan  $Data[8]$  ditukar. Pada pengulangan kedua  $Data[8]$  dibandingkan  $Data[7]$ , karena  $20 > 15$  maka proses dilanjutkan. Demikian seterusnya sampai  $j=1$ .
  - Pada saat  $i=2$ , nilai  $j$  diulang dari 9 sampai dengan 2. Pada pengulangan pertama  $Data[9]$  dibandingkan  $Data[8]$ , karena  $31 > 20$  maka proses dilanjutkan. Pada pengulangan kedua  $Data[8]$  dibandingkan  $Data[7]$ , karena  $20 < 23$   $Data[8]$  dan  $Data[7]$  ditukar. Demikian seterusnya sampai  $j=2$ .
  - Pada saat  $i=3$ , nilai  $j$  diulang dari 9 sampai dengan 3. Pada pengulangan pertama  $Data[9]$  dibandingkan  $Data[8]$ , karena  $31 > 23$  maka proses dilanjutkan. Pada pengulangan kedua  $Data[8]$  dibandingkan  $Data[7]$ , karena  $23 > 20$  maka proses dilanjutkan. Demikian seterusnya sampai  $j=3$ .
  - Dan seterusnya sampai dengan  $i=8$ .

# Metode Gelembung (*Bubble sort*)

---



# Metode Shell (Shell Sort)

---

- Metode ini disebut juga dengan metode pertambahan menurun (diminishing increment).
- 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, kemudian dilakukan penukaran bila diperlukan

## Metode Shell (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- $(j + N / 2)$ .

## Metode Shell (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  $N / 4$ .
- Demikian seterusnya sampai seluruh data dibandingkan sehingga semua data ke- $j$  lebih kecil daripada data ke- $(j + N / 4)$ .

## Metode Shell (Shell Sort)

- Algoritma metode Shell dapat dituliskan sebagai berikut :
  - 1 Jarak  $\leftarrow$  N
  - 2 Selama (Jarak  $>$  1) kerjakan baris 3 sampai dengan 9
  - 3 Jarak  $\leftarrow$  Jarak / 2. Sudah  $\leftarrow$  false
  - 4 Kerjakan baris 4 sampai dengan 8 selama Sudah = false
  - 5 Sudah  $\leftarrow$  true
  - 6 j  $\leftarrow$  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  $\leftarrow$  true
  - 9 j  $\leftarrow$  j + 1



# Metode Shell (Shell Sort)

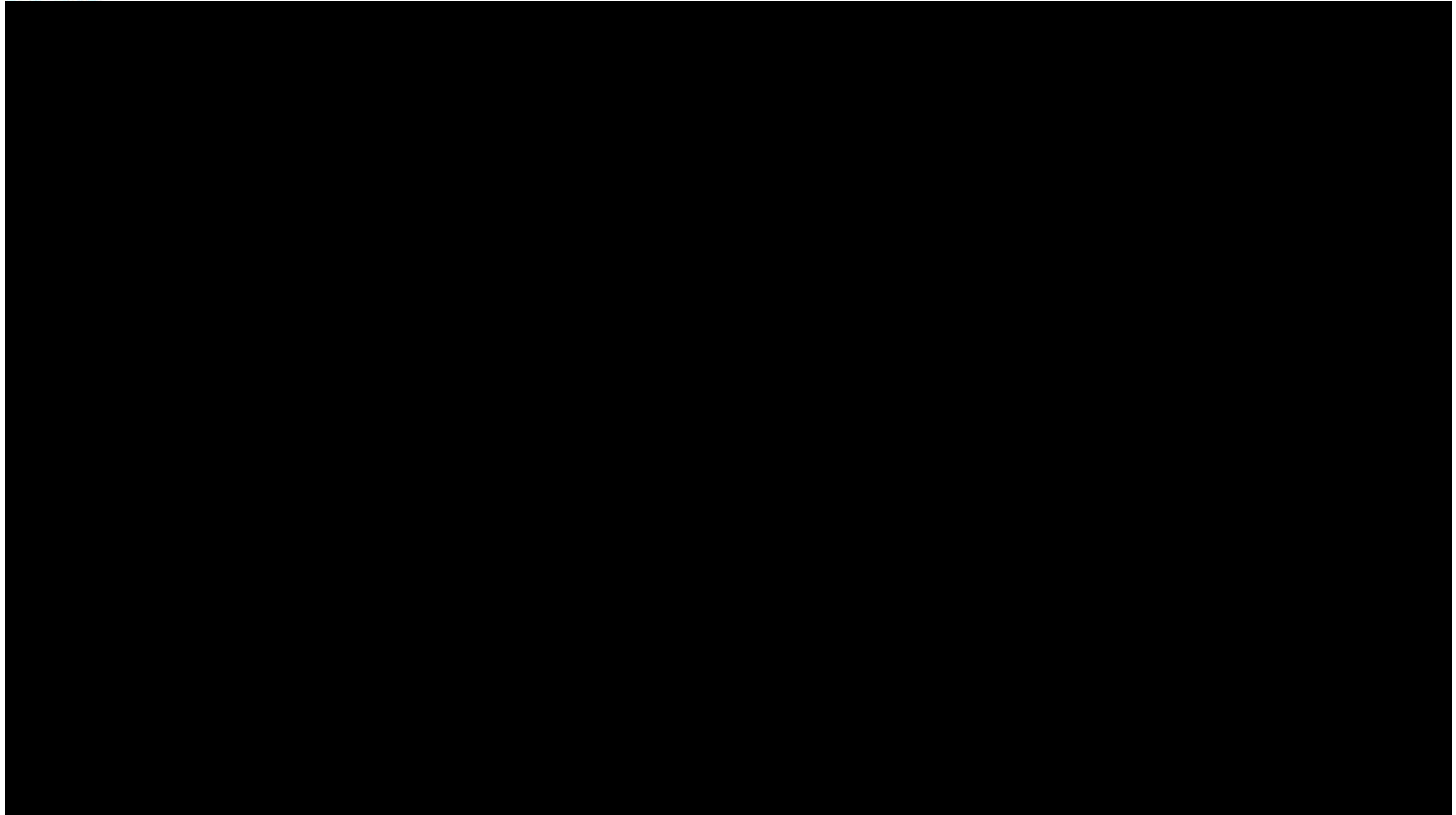
**Tabel 6.4 Proses Pengurutan dengan Metode Shell**

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	<b>35</b>	9	11	3	17	<b>23</b>	15	31	20
Jarak=2	12	23	9	11	3	17	35	15	31	20
i=2	<b>12</b>	23	<b>9</b>	11	3	17	35	15	31	20
i=3	9	<b>23</b>	12	<b>11</b>	3	17	35	15	31	20
i=4	9	11	<b>12</b>	23	<b>3</b>	17	35	15	31	20
i=5	9	11	3	<b>23</b>	12	<b>17</b>	35	15	31	20
i=7	9	11	3	17	12	<b>23</b>	35	<b>15</b>	31	20
i=8	9	11	3	17	12	15	<b>35</b>	23	<b>31</b>	20
i=9	9	11	3	17	12	15	31	<b>23</b>	35	<b>20</b>
i=2	<b>9</b>	11	<b>3</b>	17	12	15	31	20	35	23
i=3	3	<b>11</b>	9	<b>17</b>	12	15	31	20	35	23
Jarak=1	3	11	9	15	12	17	31	20	35	23
i=2	3	<b>11</b>	<b>9</b>	15	12	17	31	20	35	23
i=4	3	9	11	<b>15</b>	<b>12</b>	17	31	20	35	23
i=7	3	9	11	12	15	17	<b>31</b>	<b>20</b>	35	23
i=9	3	9	11	12	15	17	20	31	<b>35</b>	<b>23</b>
i=1	<b>3</b>	<b>9</b>	11	12	15	17	20	31	23	35
i=8	3	9	11	12	15	17	20	<b>31</b>	<b>23</b>	35
i=9	3	9	11	12	15	17	20	23	<b>31</b>	<b>35</b>
Akhir	3	9	11	12	15	17	20	23	31	35

## Metode Shell (Shell Sort)

- Proses pengurutan pada tabel 6.4 dapat dijelaskan sebagai berikut:
  - 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  $\text{Data}[0] > \text{Data}[2]$ . Proses pengulangan ini berhenti bila Sudah=true.
  - Demikian seterusnya sampai Jarak=1.

# Metode Shell (Shell Sort)



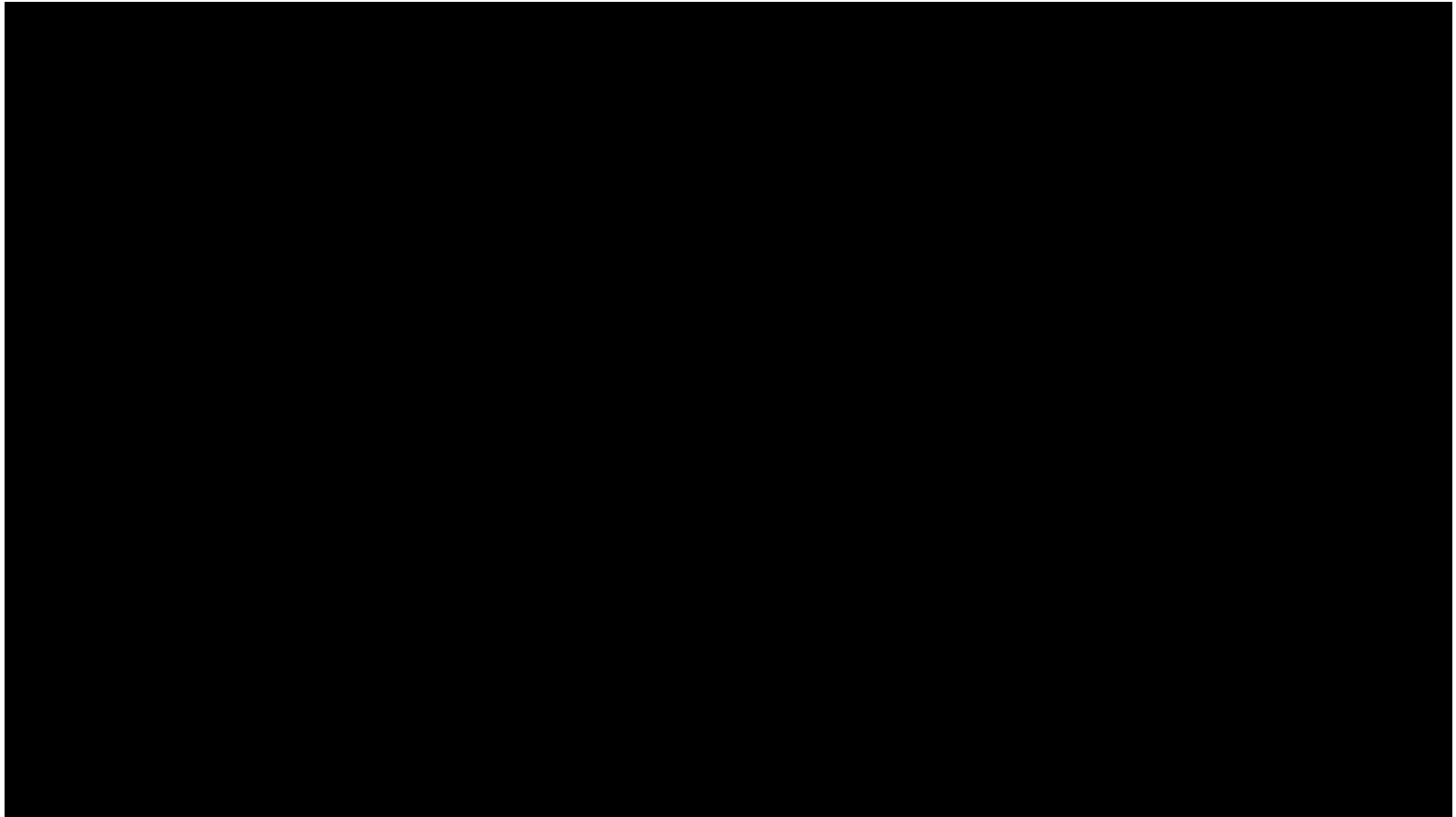
# Metode Quick (Quick Sort)

---

- Metode Quick sering disebut juga metode partisi (*partition exchange sort*).
- *Metode* ini diperkenalkan pertama kali oleh C.A.R. Hoare pada tahun 1962.

# Metode Quick (Quick Sort)

---

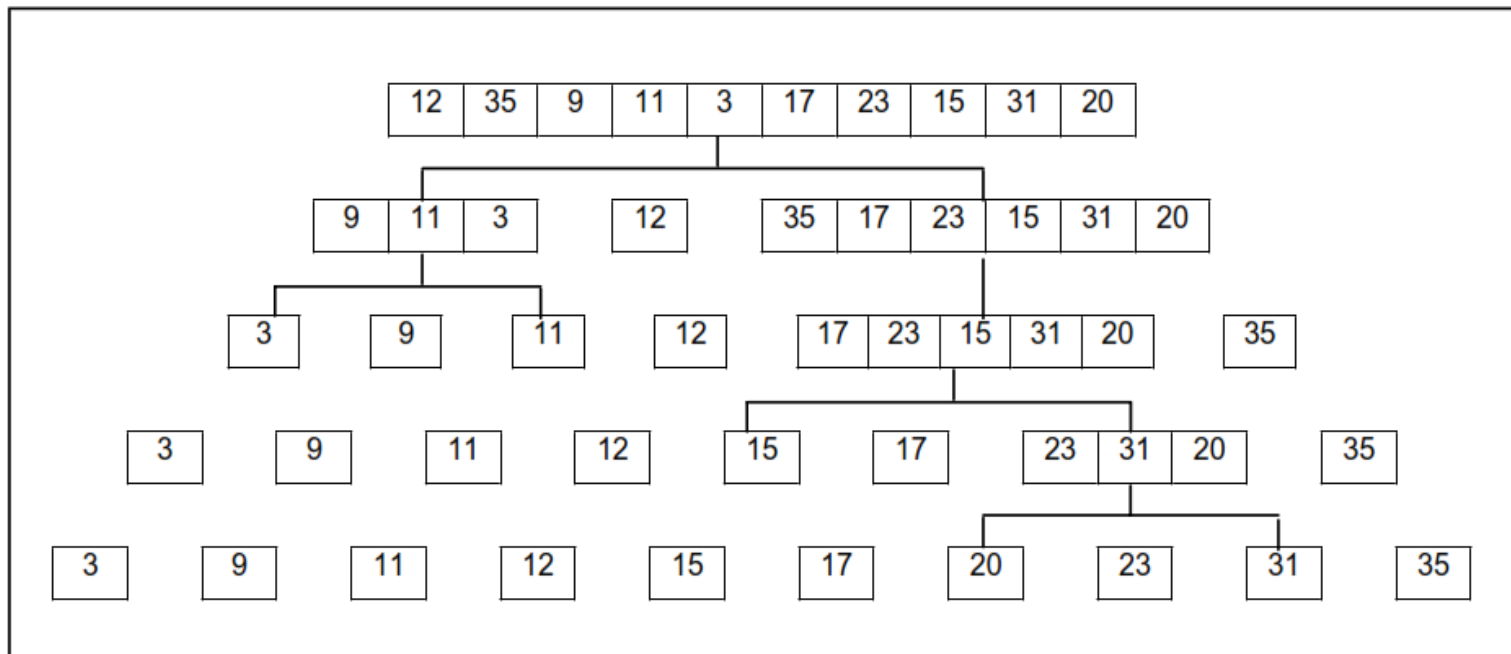


## Metode Quick (Quick Sort)

- mula-mula dipilih data tertentu yang disebut pivot, misalnya  $x$ .
- Pivot dipilih untuk mengatur data di sebelah kiri agar lebih kecil daripada pivot dan data di sebelah kanan agar lebih besar daripada pivot.
- Pivot ini diletakkan pada posisi ke  $j$  sedemikian sehingga data antara 1 sampai dengan  $j-1$  lebih kecil daripada  $x$ . Sedangkan data pada posisi ke  $j+1$  sampai  $N$  lebih besar daripada  $x$ . Caranya dengan menukarkan data diantara posisi 1 sampai dengan  $j-1$  yang lebih besar daripada  $x$  dengan data diantara posisi  $j+1$  sampai dengan  $N$  yang lebih kecil daripada  $x$ .

# Metode Quick (Quick Sort)

- Ilustrasi dari metode quick dapat dilihat pada Gambar 6.1



**Gambar 6.1 Ilustrasi Metode Quick Sort**

# Metode Quick (Quick Sort)

12      35      9      11      3      17      23      15      31      20  
 $i=0$   $j=9$

3      35      9      11      12      17      23      15      31      20  
 $i=0$   $j=4$

3      12      9      11      35      17      23      15      31      20  
 $i=1$   $j=4$

3      11      9      12      35      17      23      15      31      20  
 $i=1$   $j=3$

3	11	9
---	----	---

12
----

35	17	23	15	31	20
----	----	----	----	----	----



- Algoritma quick sort non rekursif dapat dituliskan sebagai berikut :
  - 1 Tumpukan[1].Kiri  $\leftarrow$  0
  - 2 Tumpukan[1].Kanan  $\leftarrow$  N-1
  - 3 Selama ujung  $\neq$  0 kerjakan baris 4 sampai dengan 22
  - 4 L  $\leftarrow$  Tumpukan[ujung].Kiri
  - 5 R  $\leftarrow$  Tumpukan[ujung].Kanan
  - 6 ujung  $\leftarrow$  ujung - 1
  - 7 Selama (R > L) kerjakan baris sampai 8 dengan 22
  - 8 i  $\leftarrow$  L
  - 9 j  $\leftarrow$  R
  - 10 x  $\leftarrow$  Data[(L + R) / 2]

- Algoritma quick sort non rekursif dapat dituliskan sebagai berikut :

11 Selama  $i \leq j$  kerjakan baris 12 sampai dengan 14

12 Selama  $(Data[i] < x)$ ,  $i \leftarrow i + 1$

13 Selama  $(x < Data[j])$ ,  $j \leftarrow j - 1$

14 Jika  $(i \leq j)$  maka kerjakan baris 15 sampai dengan 17, jika tidak ke baris 11

15 Tukar  $Data[i]$  dengan  $Data[j]$

16  $i \leftarrow i + 1$

17  $j \leftarrow j - 1$

18 Jika  $(L < i)$  maka kerjakan baris 19 sampai dengan 21

19  $ujung \leftarrow ujung + 1$

20  $Tumpukan[ujung].Kiri = i$

21  $Tumpukan[ujung].Kanan = R$

22  $R \leftarrow j$

Algoritma quick Rekursif dapat dituliskan sebagai berikut :

- 1  $x \leftarrow \text{Data}[(L + R) / 2]$
- 2  $i \leftarrow L$
- 3  $j \leftarrow R$
- 4 Selama ( $i \leq j$ ) kerjakan baris 5 sampai dengan 12
- 5 Selama ( $\text{Data}[i] < x$ ) kerjakan  $i \leftarrow i + 1$
- 6 Selama ( $\text{Data}[j] > x$ ) kerjakan  $j \leftarrow j - 1$
- 7 Jika ( $i \leq j$ ) maka kerjakan baris 8 sampai dengan 10; jika tidak kerjakan baris 11
- 8 Tukar  $\text{Data}[i]$  dengan  $\text{Data}[j]$
- 9  $i \leftarrow i + 1$
- 10  $j \leftarrow j - 1$
- 11 Jika ( $L < j$ ) kerjakan lagi baris 1 dengan  $R = j$
- 12 Jika ( $i < R$ ) kerjakan lagi baris 1 dengan  $L = i$

## Metode Penggabungan (Merge Sort)

---

Prinsip dari metode penggabungan sebagai berikut :

- mula-mula diberikan dua kumpulan data yang sudah dalam keadaan urut.
- Kedua kumpulan data tersebut harus dijadikan satu table sehingga dalam keadaan urut.

# Metode Penggabungan (Merge Sort)

Misalnya kumpulan data pertama (T1) adalah sebagai berikut :

3	11	12	23	31
---	----	----	----	----

kumpulan data kedua (T2) adalah sebagai berikut :

9	15	17	20	35
---	----	----	----	----

hasil sebagai berikut :

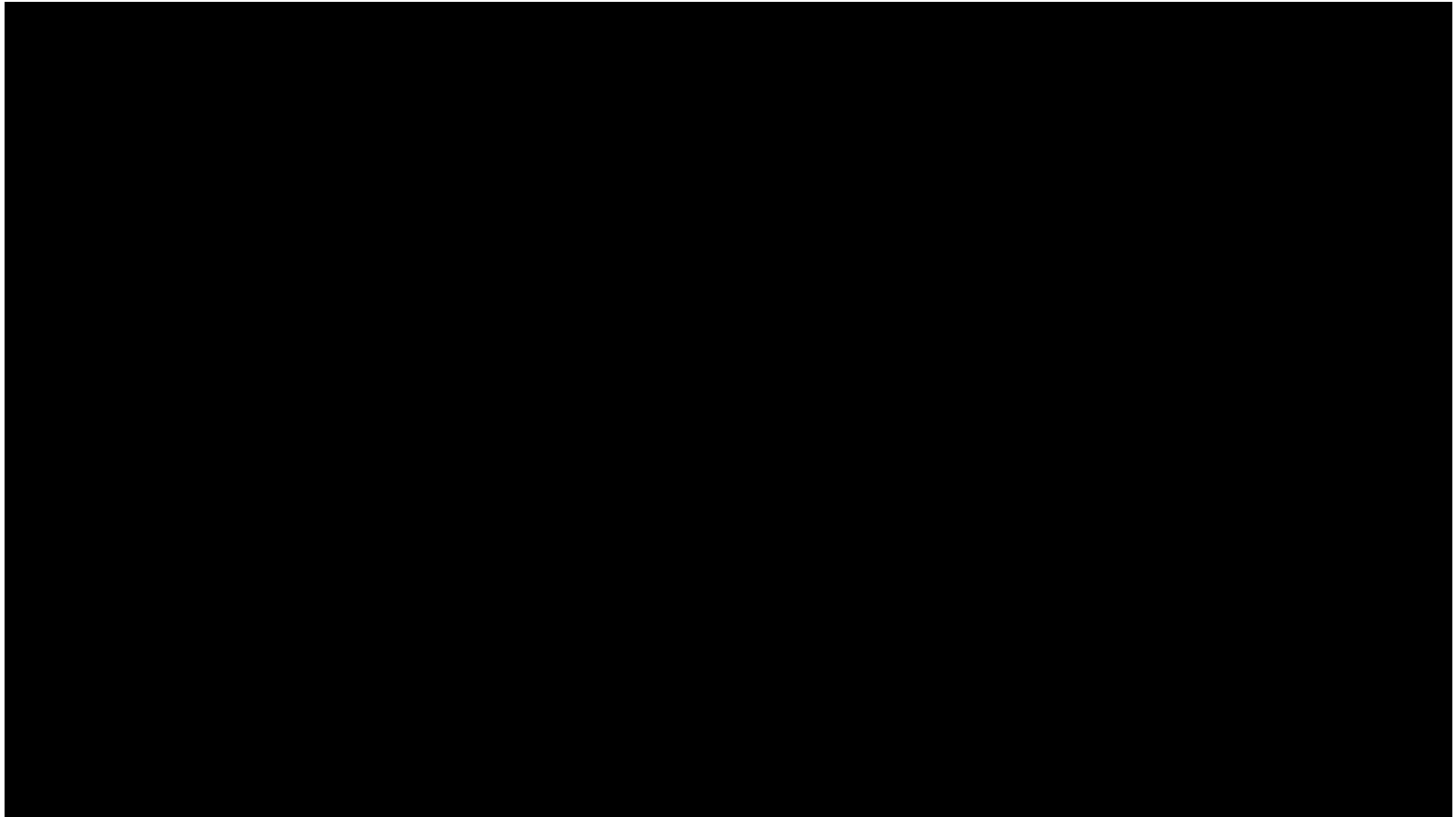
3	9	11	12	15	17	20	23	31	35
---	---	----	----	----	----	----	----	----	----

Proses penggabungan ini dapat dijelaskan sebagai berikut :

- mula-mula diambil data pertama dari T1 yaitu 3 dan data pertama dari T2 yaitu 9.
- Data ini dibandingkan, kemudian yang lebih kecil diletakkan sebagai data pertama dari hasil pengurutan, misalnya T3. Jadi T3 akan memiliki satu data yaitu 3.
- Data yang lebih besar yaitu 9 kemudian dibandingkan dengan data kedua dari T1, yaitu 11.
- Ternyata 9 lebih kecil dari 11, sehingga 9 diletakkan sebagai data kedua dari T3.

# Metode Penggabungan (Merge Sort)

---



Algoritma penggabungan dapat dituliskan sebagai berikut :

1  $i \leftarrow 0$

2  $j \leftarrow 0$

3  $J3 \leftarrow 0$

4 Kerjakan baris 5 sampai dengan 7 selama  $(i < J1)$  dan  $(j < J2)$

5 Jika  $(T1[i] < T2[j])$  maka  $T3[J3] \leftarrow T1[i]$ ,  $i \leftarrow i + 1$

6 Jika  $(T1[i] \geq T2[j])$  maka  $T3[J3] \leftarrow T2[j]$ ,  $j \leftarrow j + 1$

7  $J3 \leftarrow J3 + 1$

8 Jika  $(i \geq J1)$  maka kerjakan baris 9, jika tidak kerjakan baris 15



Algoritma penggabungan dapat dituliskan sebagai berikut :

9  $i \leftarrow j$

10 Selama ( $i < J2$ ) kerjakan baris 11 sampai dengan 13

11  $T3[J3] \leftarrow T2[i]$

12  $i \leftarrow i + 1$

13  $J3 \leftarrow J3 + 1$

14 Selesai

15  $j \leftarrow i$

16 Selama ( $j < J1$ ) kerjakan baris 17 sampai dengan 19

17  $T3[J3] \leftarrow T1[j]$

18  $j \leftarrow j + 1$

19  $J3 \leftarrow J3 + 1$

Buatlah program pengurutan menggunakan metode:

- Insertion Sort
- Binary Insertion Sort
- Selection Sort
- Bubble Sort