Algoritma Pengurutan

SORTING 2

Struktur Data

Ratih Kumalasari N
UNP Kediri

fppt.com

Tujuan

- Memahami algoritma pengurutan
- Mengerti algoritma bubble, selection, insertion, merge sort

fppt.com

Materi

- Algoritma pengurutan
- Bubble sort
- Selection sort
- Insertion sort
- merge sort

fppt.com

INSERTION SORT

fppt.com

Insertion Sort

- Cara pengurutannya : dimulai dari data ke-2 sampai dengan data terakhir, jika ditemukan data yang lebih kecil, maka akan ditempatkan (diinsert) diposisi yang seharusnya.
- Ascending : ketika perbandingan ambil data yang paling kecil.
- Descending : ketika perbandingan ambil data yang paling besar.

fppt.com

Algoritma Insertion Sort Ascending

1. Ambil satu data ke-i simpan di temp
2. Bandingkan data temp dengan data yang ada disebelah kiri satu per-satu
3. Cek apakah data temp lebih kecil dari data sebelah kiri.
4. Jika langkah nomor 3 bernilai “true” : lakukan pergeseran data satu-persatu kemudian pada posisi yang tepat sisipkan data temp.
5. Ulangi langkah 1 sampai 4, hingga i sama dengan n

fppt.com

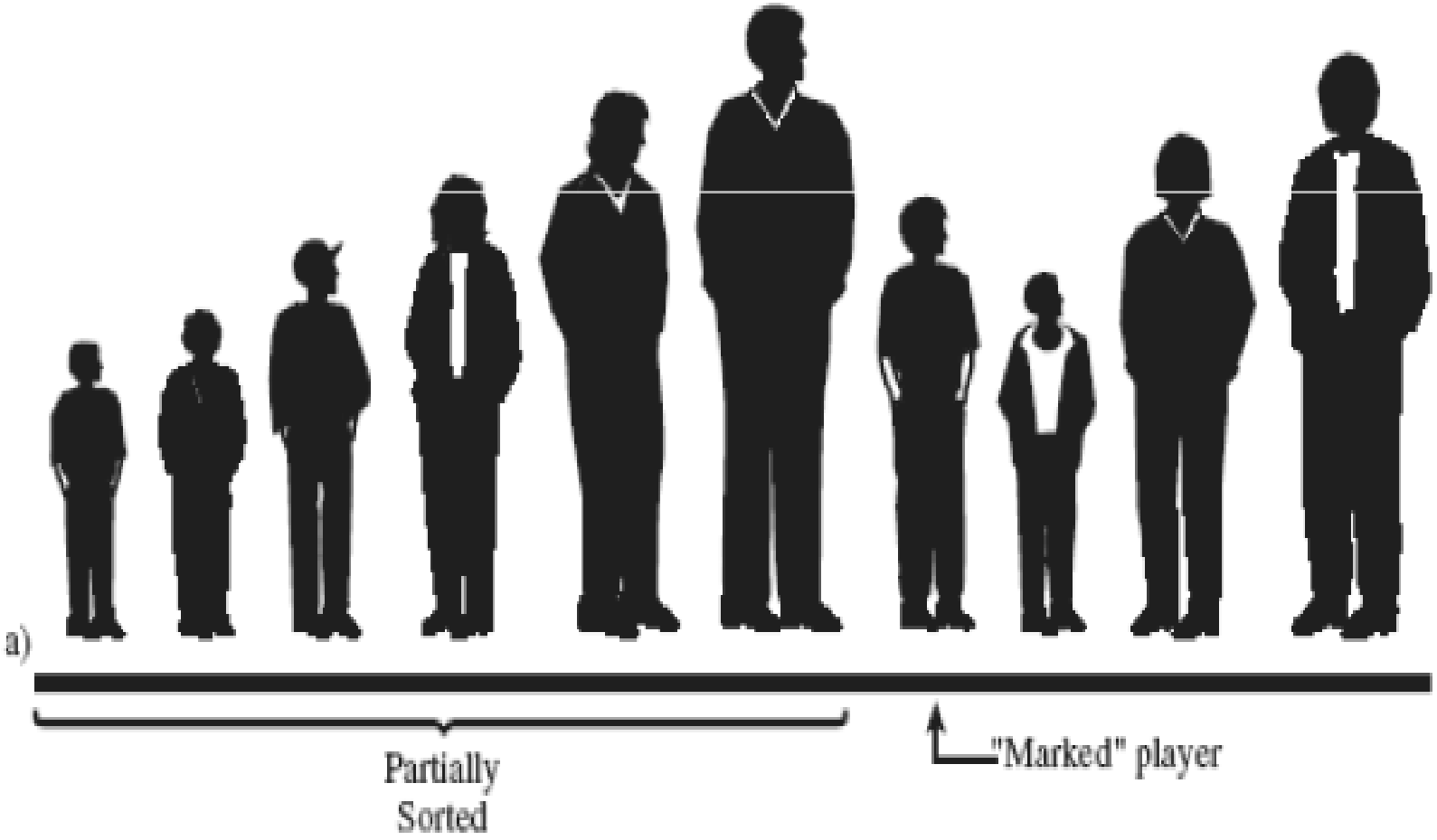
Insertion Sort Ascending

- Misalkan sekumpulan data disimpan pada A[8] dengan urutan awal = 7 3 4 1 8 2 6 5
- Temp = variabel penampung

Proses	Tampung	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]
	Data awal	7	3	4	1	8	2	6	5
1	temp = A[1]	3	7	4	1	8	2	6	5
2	temp = A[2]	3	4	7	1	8	2	6	5
3	temp = A[3]	1	3	4	7	8	2	6	5
4	temp = A[4]	1	3	4	7	8	2	6	5
5	temp = A[5]	1	2	3	4	7	8	6	5
6	temp = A[6]	1	2	3	4	6	7	8	5
7	temp = A[7]	1	2	3	4	5	6	7	8

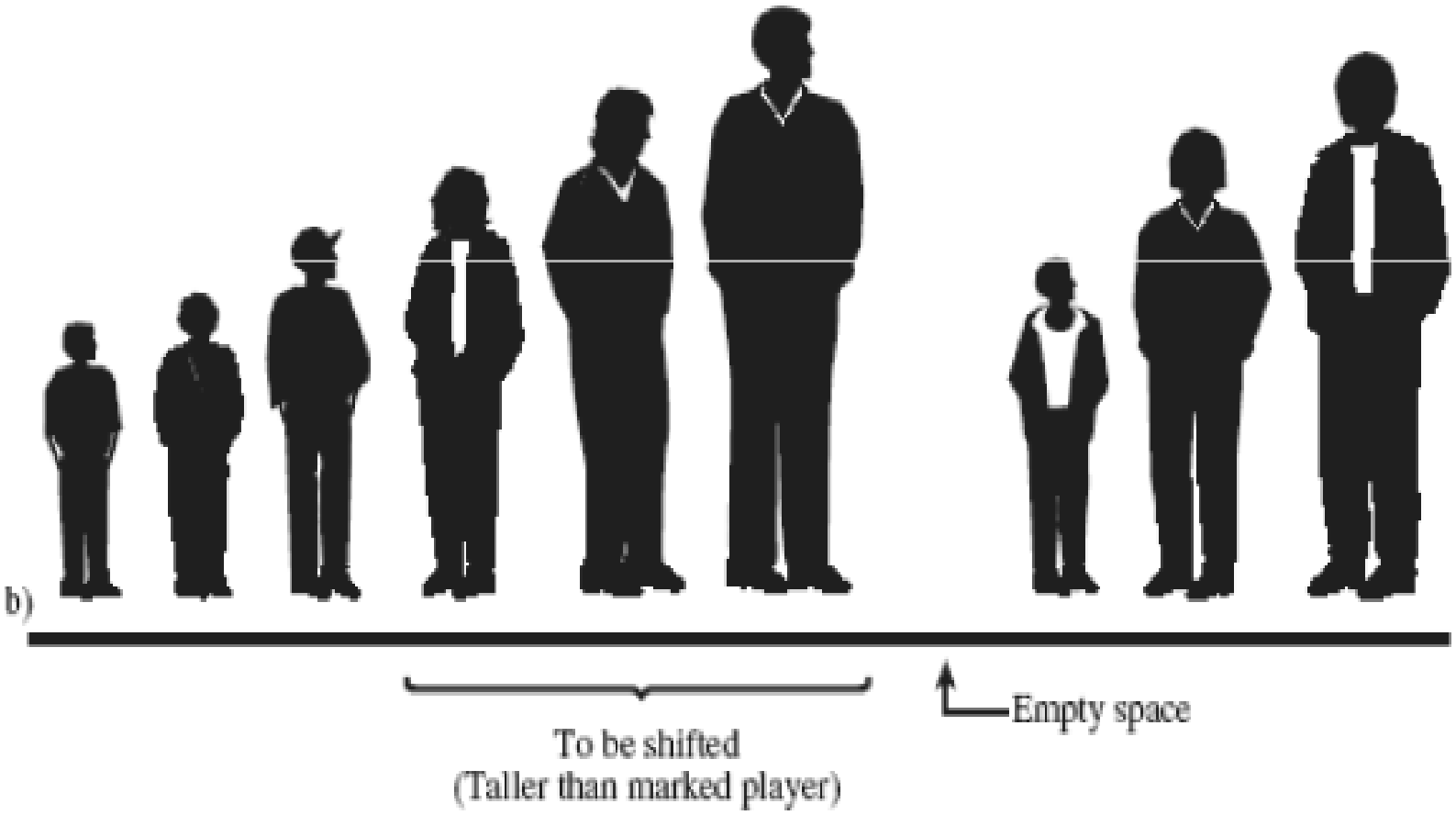
fppt.com

Ilustrasi Insertion Sort 1



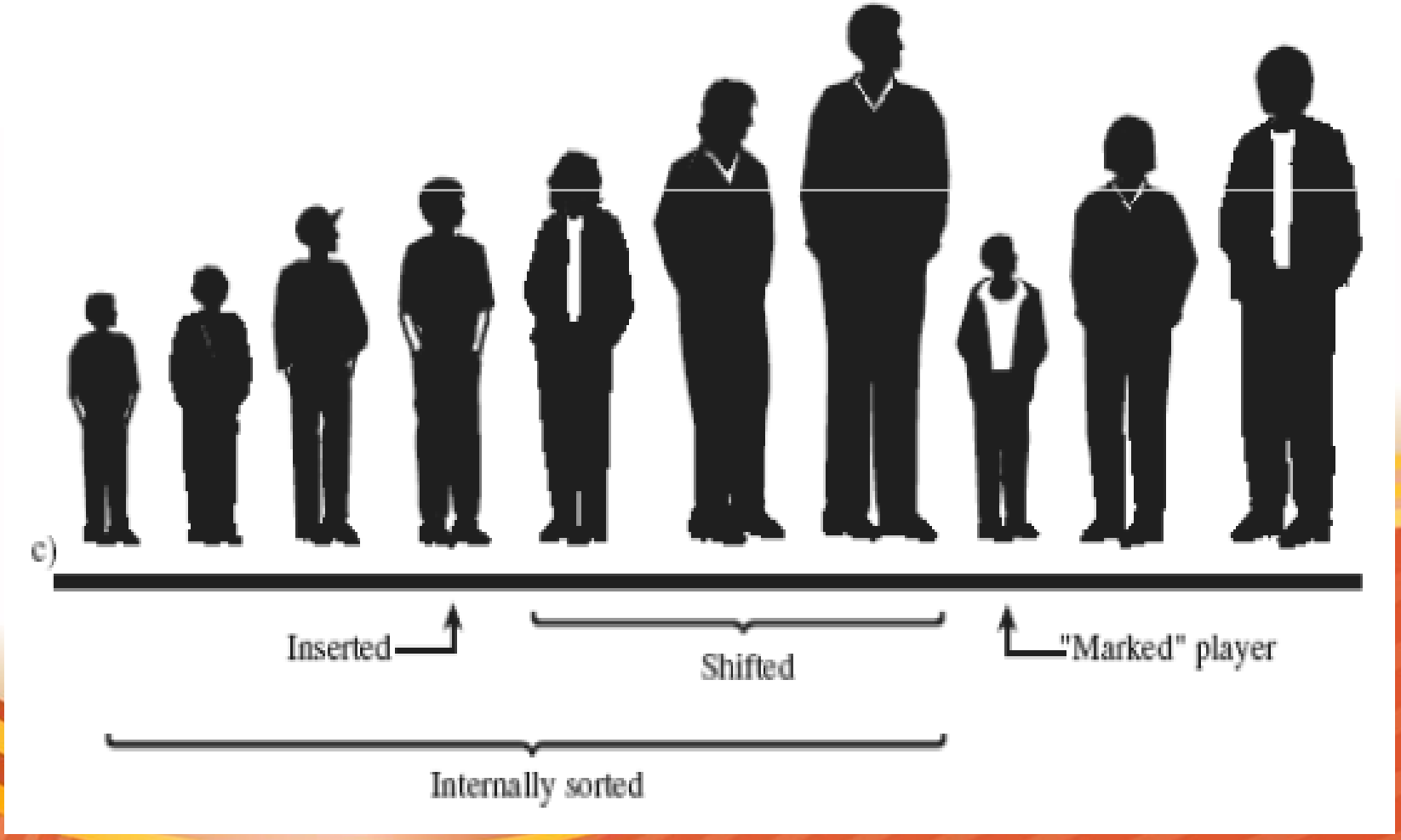
fppt.com

Ilustrasi Insertion Sort 2



fppt.com

Ilustrasi Insertion Sort 3



fppt.com

Algoritma Insertion Sort

Ambil data pertama

Pergeseran data

```
for(i=1;i<n;i++){
        t=X(i);
        for (j =i-1; j >= 0; j--) {
                if (t < X(j)){
                        X(j+1) = X(j);
                        X(j)=t;
                }
                else {
                        X(j+1) = t;
                        break;
            }
        }
    }
```

fppt.com

Time Complexity

- **<u>Proses pergeseran (copy)</u>**

  worst case : $T(n) = n(n-1)/2 = (n^2 - n)/2 = \mathbf{O(n^2)}$

  best case : $T(n) = \mathbf{0}$

- **<u>Proses perbandingan</u>**

  worst case : $T(n) = n(n-1)/2 = (n^2 - n)/2 = \mathbf{O(n^2)}$

  best case : $T(n) = n-1 = \mathbf{O(n)}$

fppt.com

MERGE SORT

fppt.com

Merge Sort

- Cara pengurutannya : memecah proses menjadi sub-proses.
- Bagi dua deret data, bandingkan, kemudian gabungkan (begitu seterusnya sampai selesai).
- Teknik pemecahan proses menggunakan rekursif.

fppt.com

Merge Sort (Merging)

- Permasalahan utama dipecah menjadi sub-masalah, kemudian solusi dari sub-masalah akan membimbing menuju solusi permasalahan utama.
- Mengimplementasikan konsep rekursi.
- Proses rekursi berhenti jika mencapai elemen dasar. Hal ini terjadi bilamana bagian yang akan diurutkan menyisakan tepat satu elemen. Sisa pengurutan satu elemen tersebut menandakan bahwa bagian tersebut telah terurut sesuai rangkaian.

fppt.com

Contoh Merge Sort

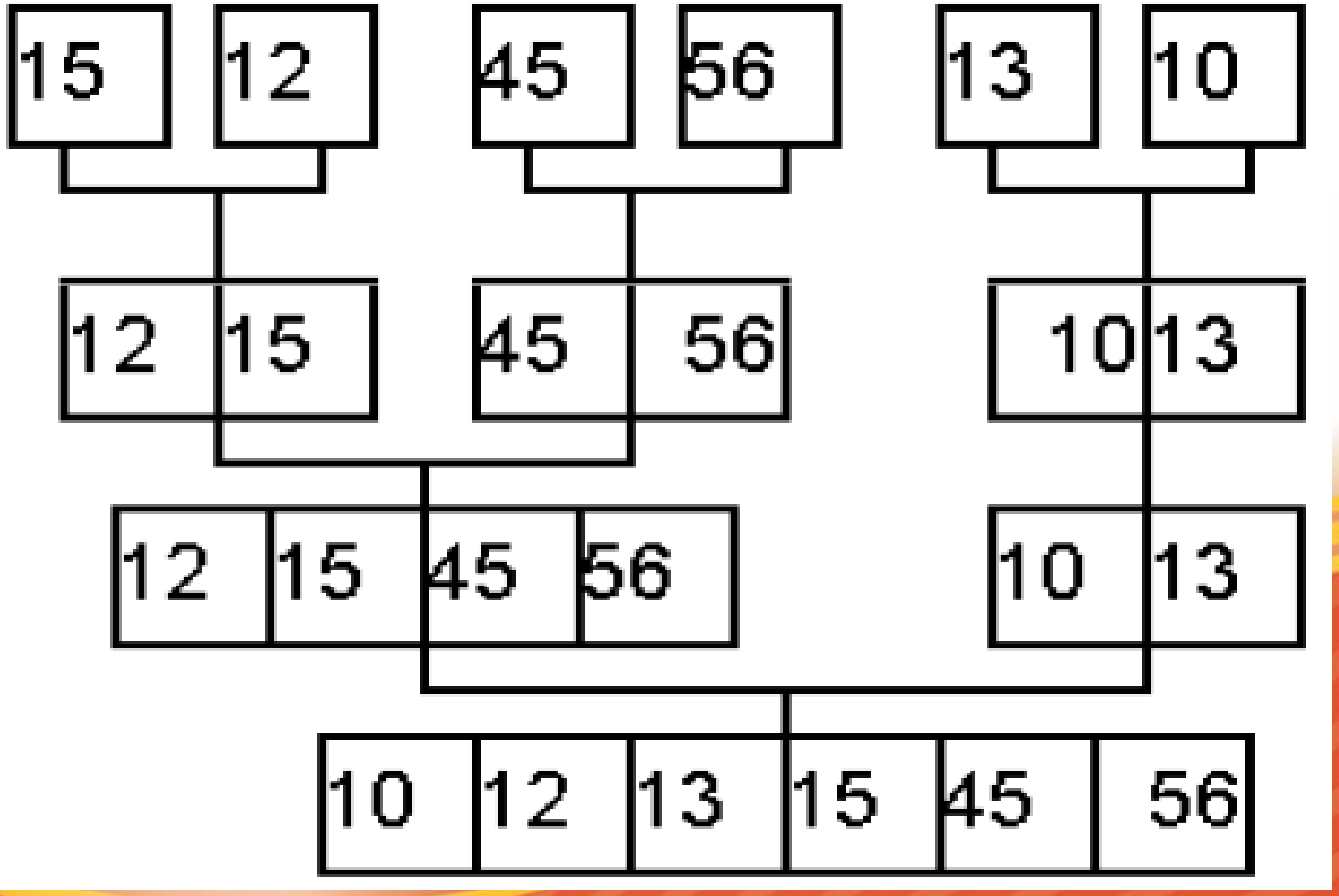
- Array A berisi 6 elemen sbb :

  15 12 45 56 13 10

- Masing-masing proses dalam algoritma merge sort akan dimulai dari elemen awal dalam A dan menggabungkan (merge) pasangan subarray yang terurut.

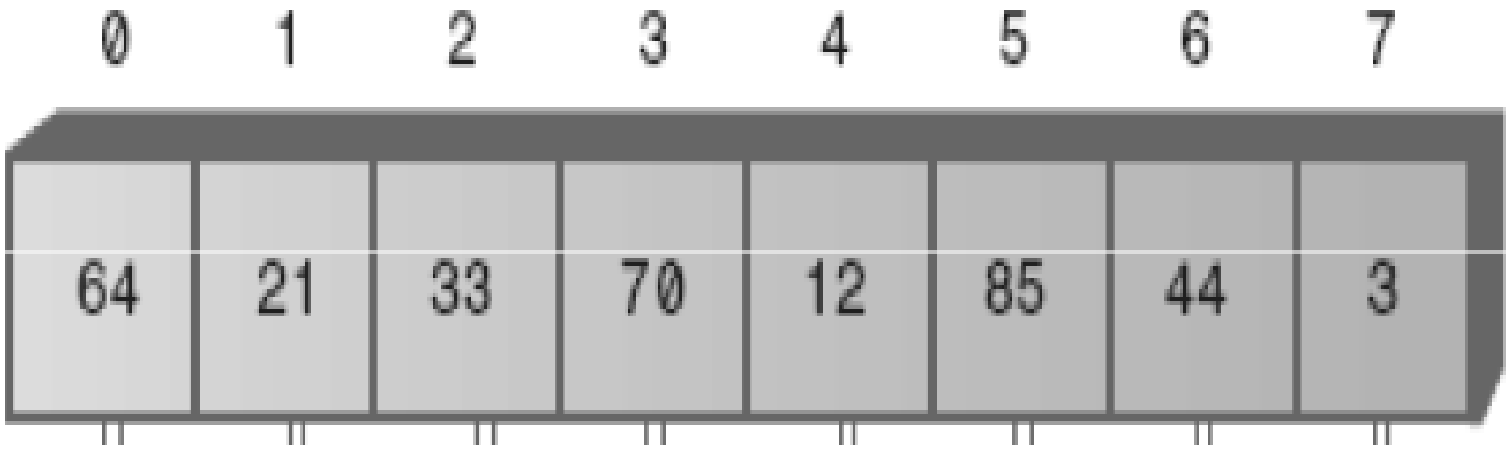
fppt.com

Ilustrasi Merge Sort 1



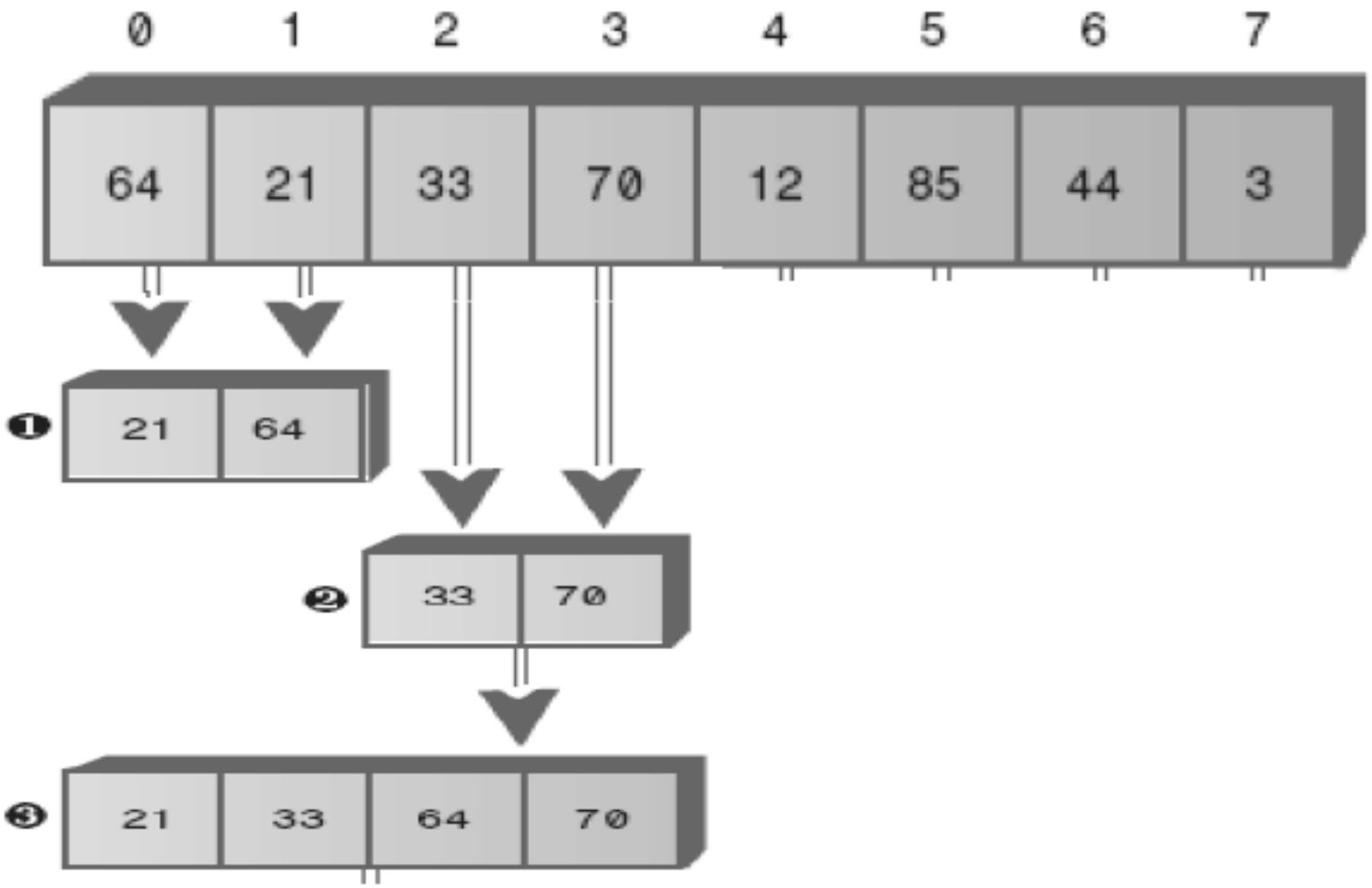
fppt.com

Ilustrasi Merge Sort 2



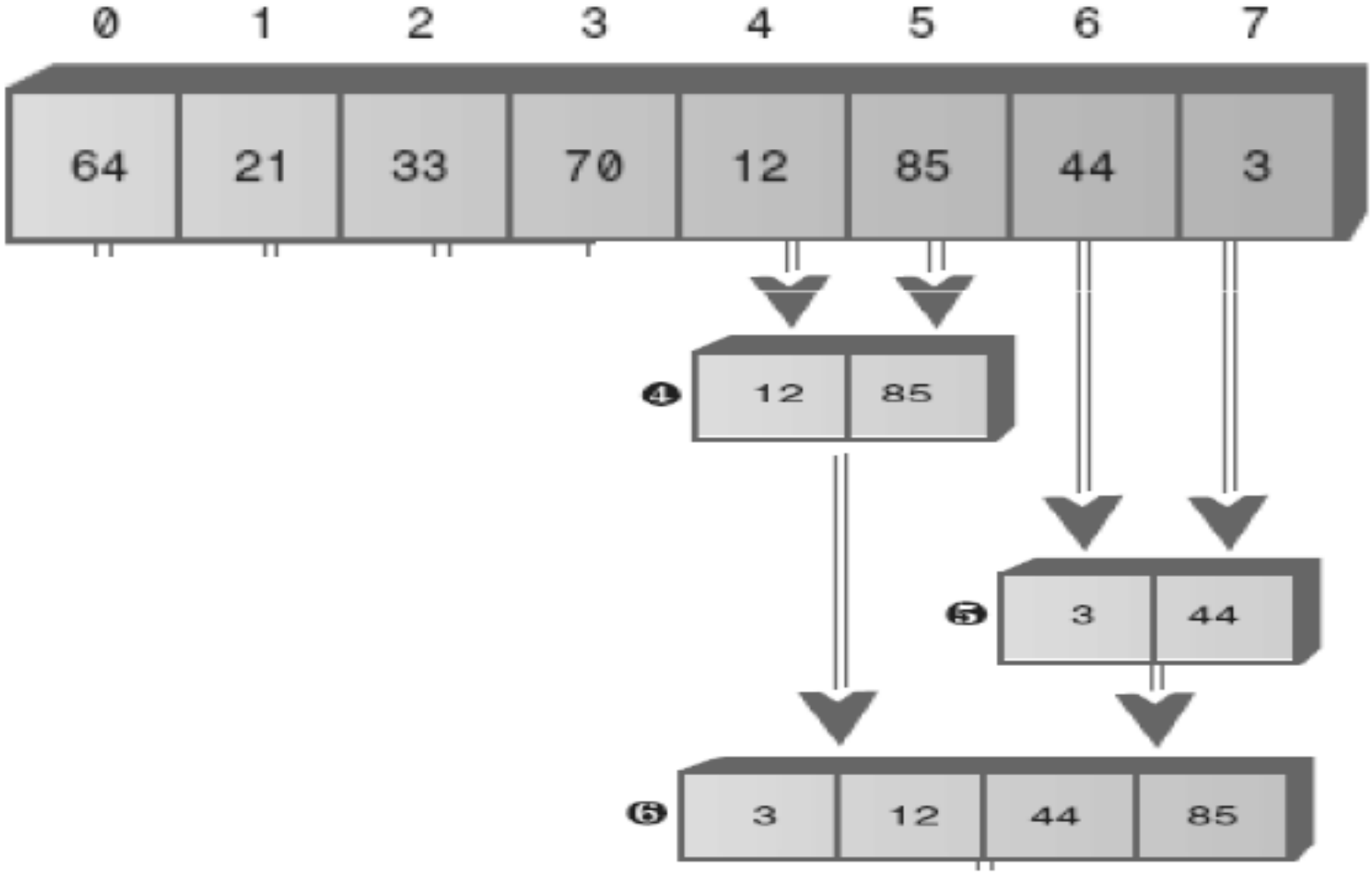
fppt.com

Ilustrasi Merge Sort 2



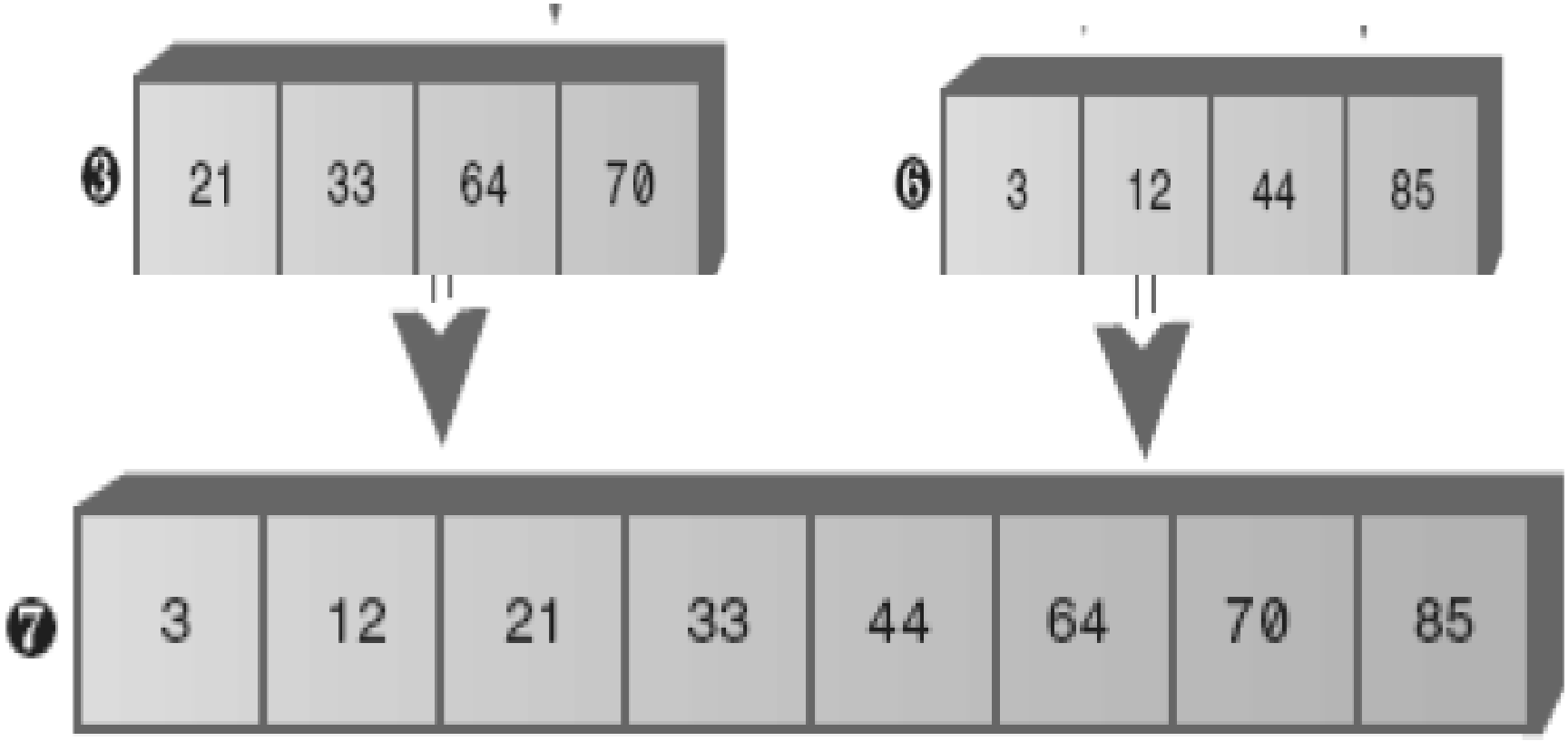
fppt.com

Ilustrasi Merge Sort 2



fppt.com

Ilustrasi Merge Sort 2



fppt.com

Ilustrasi Merge Sort 3

Rangkaian data:

7	2	5	6

Membagi rangkaian menjadi dua bagian:
LeftArr RightArr

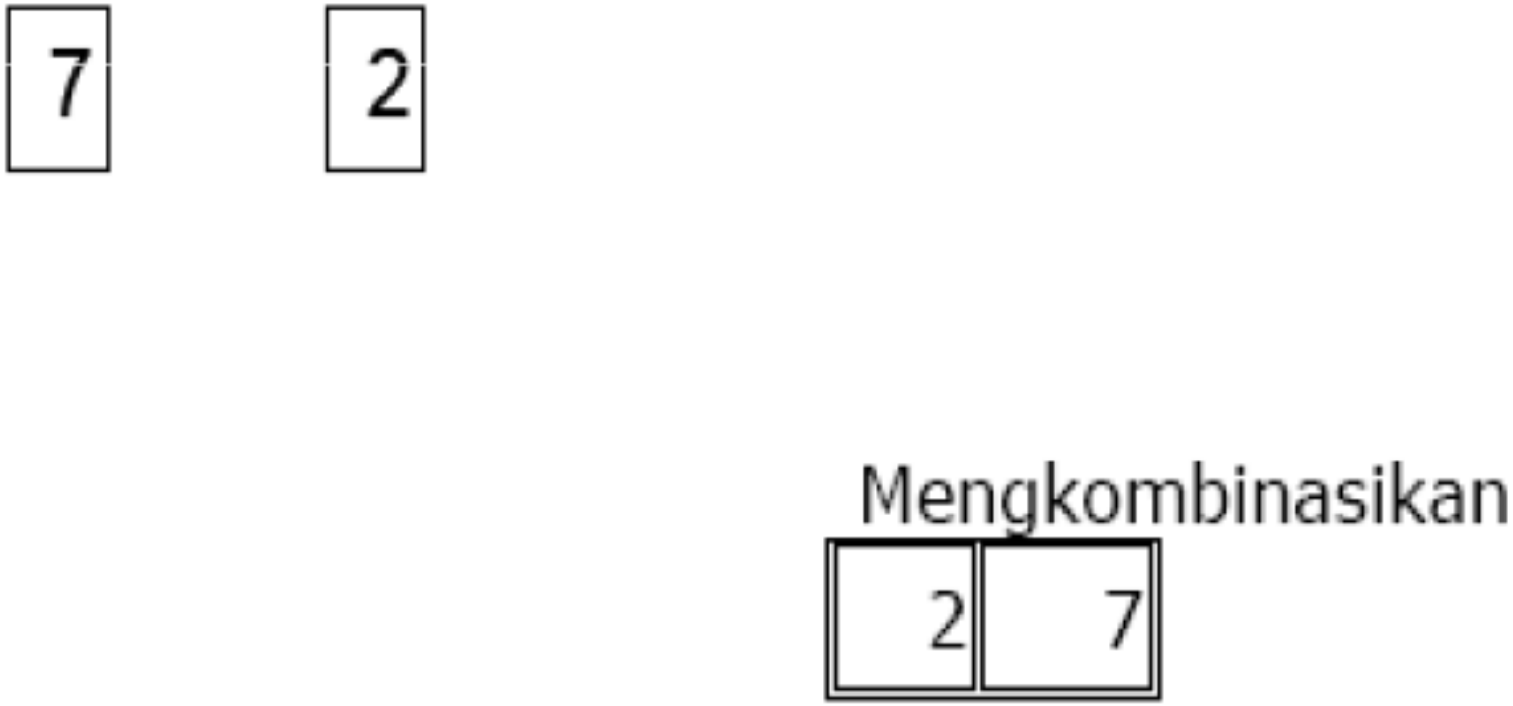
7	2

5	6

fppt.com

Ilustrasi Merge Sort 3

Membagi *LeftArr* menjadi dua bagian:
LeftArr RightArr



fppt.com

Ilustrasi Merge Sort 3

Membagi *RightArr* menjadi dua bagian:
LeftArr RightArr

Mengkombinasikan

5	6

Mengkombinasikan *LeftArr* dan *RightArr*.

2	5	6	7

fppt.com

Algoritma Merge Sort

```
static void mergeSort(int l,int r)
   {

      if(l==r) return;
         else{
              int mid = (l+r)/2;
                   mergeSort(l, mid);
              mergeSort(mid+1,r);
              Merging(l,mid+1,r);
         }
   }
```

fppt.com

```
static void Merging(int kiri,int tengah, int kanan)
  {
     int j=0;

     int batasBawah = kiri;
     int mid = tengah-1;
     int n=kanan-batasBawah+1;
     int tampung[] = new int[X.length];

     while(kiri <=mid && tengah <= kanan){
        if(X[kiri] < X[tengah])
           tampung[j++]=X[kiri++];
        else
           tampung[j+
           +]=X[tenga
           h++];
     }

     while(kiri<=mid)
        tampung[j++]=X[kiri++];
     while(tengah<=kanan)
        tampung[j++]=X[tengah++];

     for(int i=0;i<n;i++)
     {
       X[batasBawah+i]=tampung[i];
     }
  }
```

fppt.com

Tugas Latihan

- Urutkan data berikut dengan insertion sort:

  2, 10, 3, 1, 17, 25, 16, 9, 18

- Buatlah ilustrasi proses pengurutannya secara ascending maupun descending!

fppt.com

Tugas Latihan

- Urutkan data berikut dengan merge sort :

  2, 10, 3, 1, 17, 25, 16, 9, 18

- Buatlah ilustrasi proses pengurutannya secara ascending maupun descending!

fppt.com