

ALGORITMA DAN PEMROGRAMAN

MATERI 6

- ALGORITMA PENGURUTAN (*SORTING*)
- ALGORITMA PENGURUTAN APUNG (*BUBBLE SORT*)
- ALGORITMA PENGURUTAN SELEKSI (*SELECTION SORT*)

Oleh : Roni Sapto P., S. Kom.
E : ronisapto [at] gmail [dot] com
T : +62 821 16 75 93 57
W : mycampus.dezignwebster.com

ALGORITMA PENGURUTAN (*SORTING*)

- ◎ **Pengurutan (*sorting*)** adalah proses mengatur sekumpulan objek menurut urutan atau susunan tertentu.
- ◎ Bila dilihat dari urutan atau susunannya, bisa dikategorikan menjadi 2 :
 - ◎ *Ascending* (tersusun secara menaik)
 - ◎ *Descending* (tersusun secara menurun)

ALGORITMA PENGURUTAN (*SORTING*)

- Algoritma Pengurutan Apung (*Bubble Sort*)
- Algoritma Pengurutan Seleksi (*Selection Sort*)

4

ALGORITMA PENGURUTAN APUNG (*BUBBLE SORT*)

- ⊙ Terinspirasi gelembung sabun pada permukaan air.
- ⊙ Data yang diinginkan (misal data yang kecil pada pengurutan *ascending*) akan 'diapungkan' melalui proses pertukaran.
- ⊙ Proses pengapungan dilakukan sebanyak $n-1$ langkah (n adalah jumlah data).
- ⊙ Untuk setiap langkah dilakukan pertukaran 2 elemen bersebelahan (bila diperlukan) pada bagian elemen-elemen yang belum terurut.

ALGORITMA PENGURUTAN APUNG

LANGKAH KERJA

Langkah kerja :

Untuk setiap langkah $i = 1, 2, \dots, n-1$, lakukan :

Mulai dari elemen $k = n, n-1, \dots, i+1$, lakukan :

1. Bandingkan $D[k]$ dengan $D[k-1]$.
2. Pertukarkan $D[k]$ dengan $D[k-1]$ jika :
 - ⊙ $D[k] < D[k-1]$, bila tujuannya adalah menyusun secara menaik, atau
 - ⊙ $D[k] > D[k-1]$, bila tujuannya adalah menyusun secara menurun.

ALGORITMA PENGURUTAN APUNG

PENGURUTAN TERSUSUN NAIK (*ASCENDING*)

ALGORITMA Apung_Naik
 {I.S. : data pada larik belum terurut, asumsi jumlah data adalah n }
 {F.S. : mendapatkan larik yang telah terurut menaik}

DEKLARASI :
 $i, n, k, temp$: integer
 D : array [1..100] of integer

ALGORITMA :

```

INPUT(n)
FOR i ← 1 TO n-1 DO
  FOR k ← n DOWNTO i+1 DO
    IF (D[k] < D[k-1]) THEN
      temp ← D[k]
      D[k] ← D[k-1]
      D[k-1] ← temp
    ENDIF
  ENDFOR
ENDFOR
  
```

ALGORITMA PENGURUTAN APUNG

PENGURUTAN TERSUSUN TURUN (*DESCENDING*)

ALGORITMA Apung_Turun
 {I.S. : data pada larik belum terurut, asumsi jumlah data adalah n}
 {F.S. : mendapatkan larik yang telah terurut menurun}

DEKLARASI :
 i, n, k, temp : integer
 D : array [1..100] of integer

ALGORITMA :

```

INPUT(n)
FOR i ← 1 TO n-1 DO
  FOR k ← n DOWNTO i+1 DO
    IF (D[k] > D[k-1]) THEN
      temp ← D[k]
      D[k] ← D[k-1]
      D[k-1] ← temp
    ENDIF
  ENDFOR
ENDFOR
  
```

ALGORITMA PENGURUTAN SELEKSI

(*SELECTION SORT*)

- ⊙ Prinsip kerjanya adalah memilih elemen maksimum/minimum lalu ditempatkan pada awal/akhir larik (elemen terujung) dengan cara dipertukarkan secara langsung.
- ⊙ Elemen yang telah dipilih dan ditempatkan “diisolasi”, tidak lagi diikuti-sertakan pada pemilihan berikutnya, karena telah menempati posisi yang tepat.
- ⊙ Area pemilihan berkurang satu elemen, karena setelah penempatan elemen yang telah terpilih tidak lagi diikuti-sertakan.

ALGORITMA PENGURUTAN SELEKSI

(*SELECTION SORT*)

- **Algoritma Pengurutan Seleksi-Maksimum**
 - Elemen yang dipilih adalah elemen bernilai maksimum.
- **Algoritma Pengurutan Seleksi-Minimum**
 - Elemen yang dipilih adalah elemen bernilai minimum.

ALGORITMA PENGURUTAN SELEKSI

PENGURUTAN SELEKSI-MAKSIMUM (*ASCENDING*)

ALGORITMA *Seleksi_Max_Naik*
 {I.S. : data pada larik belum terurut, asumsi jumlah data adalah n}
 {F.S. : mendapatkan larik yang telah terurut menaik}

DEKLARASI :
 imax, i, n, k, temp : integer
 D : array [1..100] of integer

ALGORITMA :

```

INPUT(n)
FOR i ← n DOWNTO 2 DO
  imax ← i
  FOR k ← i-1 DOWNTO 1 DO
    IF (D[k] > D[imax]) THEN
      imax ← k
    ENDIF
  ENDFOR
  temp ← D[i]
  D[i] ← D[imax]
  D[imax] ← temp
ENDFOR
  
```

ALGORITMA PENGURUTAN SELEKSI

PENGURUTAN SELEKSI-MAKSIMUM (*DESCENDING*)

ALGORITMA Seleksi_Max_Turun
 {I.S. : data pada larik belum terurut, asumsi jumlah data adalah n}
 {F.S. : mendapatkan larik yang telah terurut menurun}

DEKLARASI :
 imax, i, n, k, temp : integer
 D : array [1..100] of integer

ALGORITMA :
 INPUT(n)
 FOR i ← 1 TO n-1 DO
 imax ← i
 FOR k ← i+1 TO n DO
 IF (D[k] > D[imax]) THEN
 imax ← k
 ENDIF
 ENDFOR
 temp ← D[i]
 D[i] ← D[imax]
 D[imax] ← temp
 ENDFOR

ALGORITMA PENGURUTAN SELEKSI

PENGURUTAN SELEKSI-MINIMUM (*ASCENDING*)

ALGORITMA Seleksi_Min_Naik
 {I.S. : data pada larik belum terurut, asumsi jumlah data adalah n}
 {F.S. : mendapatkan larik yang telah terurut menaik}

DEKLARASI :
 imax, i, n, k, temp : integer
 D : array [1..100] of integer

ALGORITMA :
 INPUT(n)
 FOR i ← 1 TO n-1 DO
 imax ← i
 FOR k ← i+1 TO n DO
 IF (D[k] < D[imax]) THEN
 imax ← k
 ENDIF
 ENDFOR
 temp ← D[i]
 D[i] ← D[imax]
 D[imax] ← temp
 ENDFOR

ALGORITMA PENGURUTAN SELEKSI

PENGURUTAN SELEKSI-MINIMUM (*DESCENDING*)

ALGORITMA Seleksi_Max_Naik

{I.S. : data pada larik belum terurut, asumsi jumlah data adalah n}

{F.S. : mendapatkan larik yang telah terurut menurun}

DEKLARASI :

imax, i, n, k, temp : integer

D : array [1..100] of integer

ALGORITMA :

```

INPUT(n)
FOR i ← n DOWNTO 2 DO
  imax ← i
  FOR k ← i-1 DOWNTO 1 DO
    IF (D[k] < D[imax]) THEN
      imax ← k
    ENDIF
  ENDFOR
  temp ← D[i]
  D[i] ← D[imax]
  D[imax] ← temp
ENDFOR

```

ALGORITMA PENGURUTAN SELEKSI

PERBEDAAN *MAX-MIN* DAN *ASC-DESC*

	Seleksi-Maksimum	Seleksi-Minimum
Ascending	<pre> FOR i ← n DOWNTO 2 DO imax ← i FOR k ← i-1 DOWNTO 1 DO IF (D[k] > D[imax]) THEN imax ← k ENDIF ENDFOR temp ← D[i] D[i] ← D[imax] D[imax] ← temp ENDFOR </pre>	<pre> FOR i ← 1 TO n-1 DO imax ← i FOR k ← i+1 TO n DO IF (D[k] < D[imax]) THEN imax ← k ENDIF ENDFOR temp ← D[i] D[i] ← D[imax] D[imax] ← temp ENDFOR </pre>
Descending	<pre> FOR i ← 1 TO n-1 DO imax ← i FOR k ← i+1 TO n DO IF (D[k] > D[imax]) THEN imax ← k ENDIF ENDFOR temp ← D[i] D[i] ← D[imax] D[imax] ← temp ENDFOR </pre>	<pre> FOR i ← n DOWNTO 2 DO imax ← i FOR k ← i-1 DOWNTO 1 DO IF (D[k] < D[imax]) THEN imax ← k ENDIF ENDFOR temp ← D[i] D[i] ← D[imax] D[imax] ← temp ENDFOR </pre>