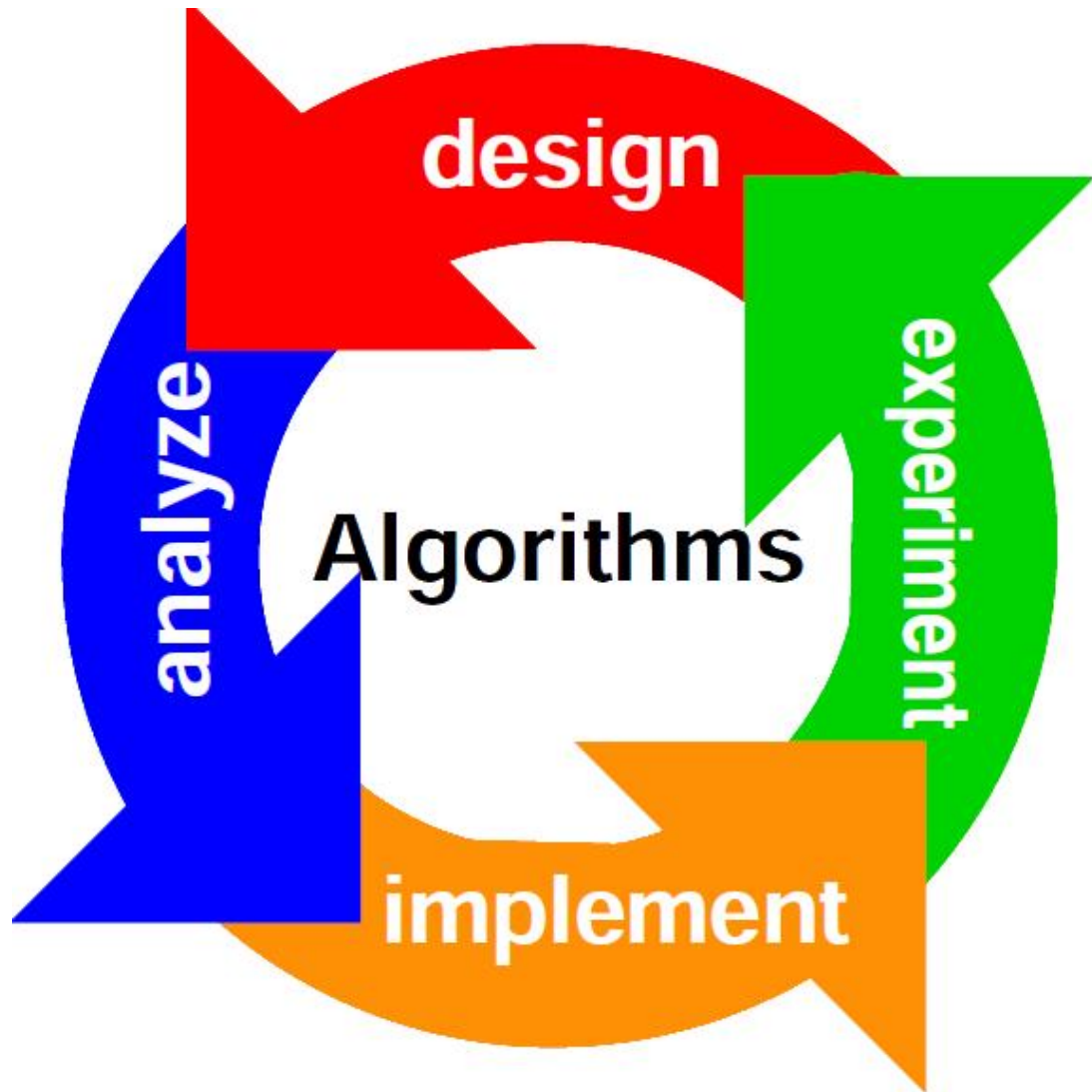


ALGORITMA DAN PEMROGRAMAN 2



TEKNIK PERANGKAT LUNAK
FT - UNPAM

3. PENGURUTAN (SORTING)

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pengurutan : proses mengatur sekumpulan obyek meurut urutan atau susunan tertentu. Urutan menaik(ascending) : $L[0] \leq L[1] \leq L[2] \leq \dots \leq L[n]$ Urutan menurun(descending) : $L[0] \geq L[1] \geq L[2] \geq \dots \geq L[n]$ | Keuntungan dari data yang terurut: <ul style="list-style-type: none"> - mempercepat proses pencarian - langsung diperoleh nilai maksimum dan minimum |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Contoh :

- a. data bertipe integer terurut naik:

| | | | | |
|----|----|----|----|----|
| 23 | 30 | 40 | 45 | 60 |
|----|----|----|----|----|

- b. data bertipe riil terurut menurun:

| | | | | |
|-------|------|------|------|------|
| 100,1 | 60,3 | 50,4 | 40,5 | 10,7 |
|-------|------|------|------|------|

- c. data bertipe string terurut naik:

| | | | | |
|------|------|-------|------|------|
| Amir | Badu | Cecep | Dudi | Rudi |
|------|------|-------|------|------|

- d. data bertipe karakter terurut naik:

| | | | | |
|---|---|---|---|---|
| d | e | g | l | m |
|---|---|---|---|---|

- e. data terstruktur terurut naik berdasarkan field NIM

| | | | |
|----------------|----------------|-----------------|-----------------|
| 135901, Eko, A | 135902, Edo, A | 135903, Dudi, B | 135904, Amir, A |
|----------------|----------------|-----------------|-----------------|

Metode pengurutan diklasifikasikan menjadi dua:

- Internal : metode pengurutan untuk data yang disimpan di dalam memori komputer. Umumnya struktur internal yang dipakai untuk mengurutkan internal adalah larik, sehingga pengurutan internal disebut juga pengurutan larik.
- Eksternal : metode pengurutan untuk data yang disimpan didalam disk storage, disebut juga pengurutan arsip (file), karena struktur eksternal yang digunakan adalah arsip.

I. METODE PENGURUTAN APUNG (BUBBLE SORT)

Metode ini diinspirasi oleh gelembung sabun yang berada diatas permukaan air. Gelembung sabun selalu terapung ke atas permukaan air karena berat jenisnya lebih ringan daripada berat jenis air. Jika sebuah array dengan jumlah elemen (ukuran array) sebesar N maka proses pengapungan dilakukan sebanyak N-1 langkah (satu langkah disebut satu pass). Proses pengapungan dilakukan dengan cara perbandingan.

Jumlah langkah = $N - 1$

Jumlah perbandingan = $N(N - 1)/2$

Algoritma Pengurutan Apung :

| | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|
| L = | 30 | 60 | 35 | 70 | 50 | 25 | 20 | 40 | 43 |
| k = | 0 | 1 | .. | .. | .. | .. | .. | .. | n |

Untuk mendapatkan larik yang terurut menaik, algoritma pengurutan apung secara global sbb:

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

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

1) Bandingkan $L[k]$ dengan $L[k-1]$

2) Pertukarkan $L[k]$ dengan $L[k-1]$ jika $L[k] < L[k-1]$

Contoh :

Tinjau L dengan $N = 6$ buah elemen belum terurut ($n = 5$) :

| | | | | | | |
|-----|----|----|----|---|----|----|
| L = | 25 | 27 | 10 | 8 | 76 | 21 |
| k = | 0 | 1 | 2 | 3 | 4 | 5 |

Pass 1:

| k | Elem. yg dibandingkan | Pertukarkan ? | Hasil Sementara |
|---|-----------------------|---------------|----------------------------------------------------------------------------------------------------------------------------------|
| n | | | a |
| 5 | $L[5] < L[4]$ | Ya | 25, 27, 10, |
| 4 | $L[4] < L[3]$ | Tidak | 8, <u>21</u> , 76 |
| 3 | $L[3] < L[2]$ | Ya | 25, 27, 10, |
| 2 | $L[2] < L[1]$ | Ya | <u>8</u> , 21, 76 |
| 1 | $L[1] < L[0]$ | Ya | 25, 27, <u>8</u> , <u>10</u> , 21, 76 25, <u>8</u> , <u>27</u> , 10, 21, 76 <u>8</u> , <u>25</u> , 27, 10, 21, 76 |

Hasil akhir pass-1:

| | | | | | |
|---|----|----|----|----|----|
| 8 | 25 | 27 | 10 | 21 | 76 |
|---|----|----|----|----|----|

Pass 2 (berdasarkan hasil akhir pass 1) :

| k | Elem. yg dibandingkan | Pertukarkan ? | Hasil Sementara |
|---|-----------------------|---------------|-----------------|
|---|-----------------------|---------------|-----------------|

| n | | | a |
|---|---------------|-------|--------------------------------------------------------------------------------------------------------------------|
| 5 | $L[5] < L[4]$ | Tidak | 8, 25, 27, |
| 4 | $L[4] < L[3]$ | Tidak | 10, <u>21</u> , 76 |
| 3 | $L[3] < L[2]$ | Ya | 8, 25, 27, |
| 2 | $L[2] < L[1]$ | Ya | <u>10</u> , <u>21</u> , 76 8, 25, <u>10</u> , <u>27</u> , 21, 76 8, <u>10</u> , <u>25</u> , 27, 21, 76 |

Hasil akhir pass-2:

| | | | | | |
|---|----|----|----|----|----|
| 8 | 10 | 25 | 27 | 21 | 76 |
|---|----|----|----|----|----|

Pass 3 (berdasarkan hasil akhir pass-2)

| k | Elem yg dibandingkan | Pertukarkan? | Hasil S |
|---|----------------------|--------------|--------------------|
| 5 | $L[5] < L[4]$ | Tidak | 8, 10, 25, |
| 4 | $L[4] < L[3]$ | Ya | 8, 10, 25, |
| 3 | $L[3] < L[2]$ | Ya | 8, 10, <u>21</u> , |

Hasil akhir pass-3:

| | | | | | |
|---|----|----|----|----|----|
| 8 | 10 | 21 | 25 | 27 | 76 |
|---|----|----|----|----|----|

Pass 4 (berdasarkan hasil akhir pass-3)

| k | Elem. yg dibandingkan | Pertukarkan ? | Hasil Sementara |
|---|-----------------------|---------------|--------------------|
| n | | | a |
| 5 | $L[5] < L[4]$ | Tidak | 8, 10, 21, |
| 4 | $L[4] < L[3]$ | Tidak | 25, <u>27</u> , 76 |
| | | | 8, 10, 21, |
| | | | <u>25, 27</u> , 76 |

Hasil akhir pass-4:

| | | | | | |
|---|----|----|----|----|----|
| 8 | 10 | 21 | 25 | 27 | 76 |
|---|----|----|----|----|----|

Pass 5 (berdasarkan hasil akhir pass-4)

| k | Elem. yg dibandingkan | Pertukarkan ? | Hasil Sementara |
|---|-----------------------|---------------|-------------------|
| n | | | a |
| 5 | $L[5] < L[4]$ | Tidak | 8, 10, 21, |
| | | | 25, 27, 76 |

Hasil akhir pass-5:

| | | | | | |
|---|----|----|----|----|----|
| 8 | 10 | 21 | 25 | 27 | 76 |
|---|----|----|----|----|----|

Hasil akhir pass-5 menyisakan satu elemen (yaitu 76) yang tidak perlu diurutkan, maka pengurutan selesai.

Algoritma pengurutan apung menaik:

procedure BubbleSort(input/output L : LarikInt, input n : integer)
 {Keadaan Awal : Elemen larik L sudah terdefinisi nilai-nilainya}
 {Keadaan Akhir : Elemen larik L terurut menaik}

DEKLARASI

i : integer {pencacah untuk jumlah langkah}
 k : integer {pencacah untuk pengapungan pada setiap langkah}
 tmp : integer {variabel bantu untuk pertukaran}

DESKRIPSI

```
for i ← 1 to n do
    for k ← n downto i do
        if L(k) < L(k - 1) then
            tmp ← L[k]
            L[k] ← L[k - 1]
            L[k - 1] ← tmp
        endif
    endfor
endfor
```

Metode pengurutan ini merupakan metode yang tidak efisien untuk data yang besar, disebabkan oleh banyaknya operasi pertukaran yang dilakukan pada setiap langkah pengapungan.

Translasi algoritma pengurutan dengan metode bubble sort kedalam bahasa C sbb:

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <pre>/*Mengurutkan data dengan metode Bubble Sort*/ #include<iostream.h> void BubbleSort(int data[],int n); //prototipe fungsi void main(void) { int i;</pre> | <pre>void BubbleSort(int array1[],int n) { int i,k,tmp; for(i=1;i<=n;i++) {</pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> int n=9;//Jml Data = 10 maka n = 9 int data[]={20,10,32,100,60,12,70,25,45,65}; cout<<"Sebelum diurutkan :"<<endl; for(i=0;i<=n;i++) cout<<data[i]<<" "; cout<<endl; cout<<"_____ " <<endl; BubbleSort(data,n); /*Pemanggilan fungsi BubbleSort*/ cout<<"Setelah diurutkan"<<endl; for(i=0;i<=n;i++) cout<<data[i]<<" "; cout<<endl; } </pre> | <pre> for(k=n;k>=i;k--) { if(array1[k]<array1 [k-1]) { tmp=array1[k]; array1[k]=array1[k -1]; array1[k-1]=tmp; } } } } </pre> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

TUGAS

1. Tulislah algoritma untuk pengurutan secara menurun menggunakan metode bubble sort (min dan max).
2. Buatlah programnya menggunakan C++ untuk array sbb:

| | | | | | | | | | |
|----|----|----|----|----|---|---|-----|-----|----|
| 10 | 40 | 20 | 60 | 15 | 4 | 8 | 100 | 200 | 25 |
|----|----|----|----|----|---|---|-----|-----|----|