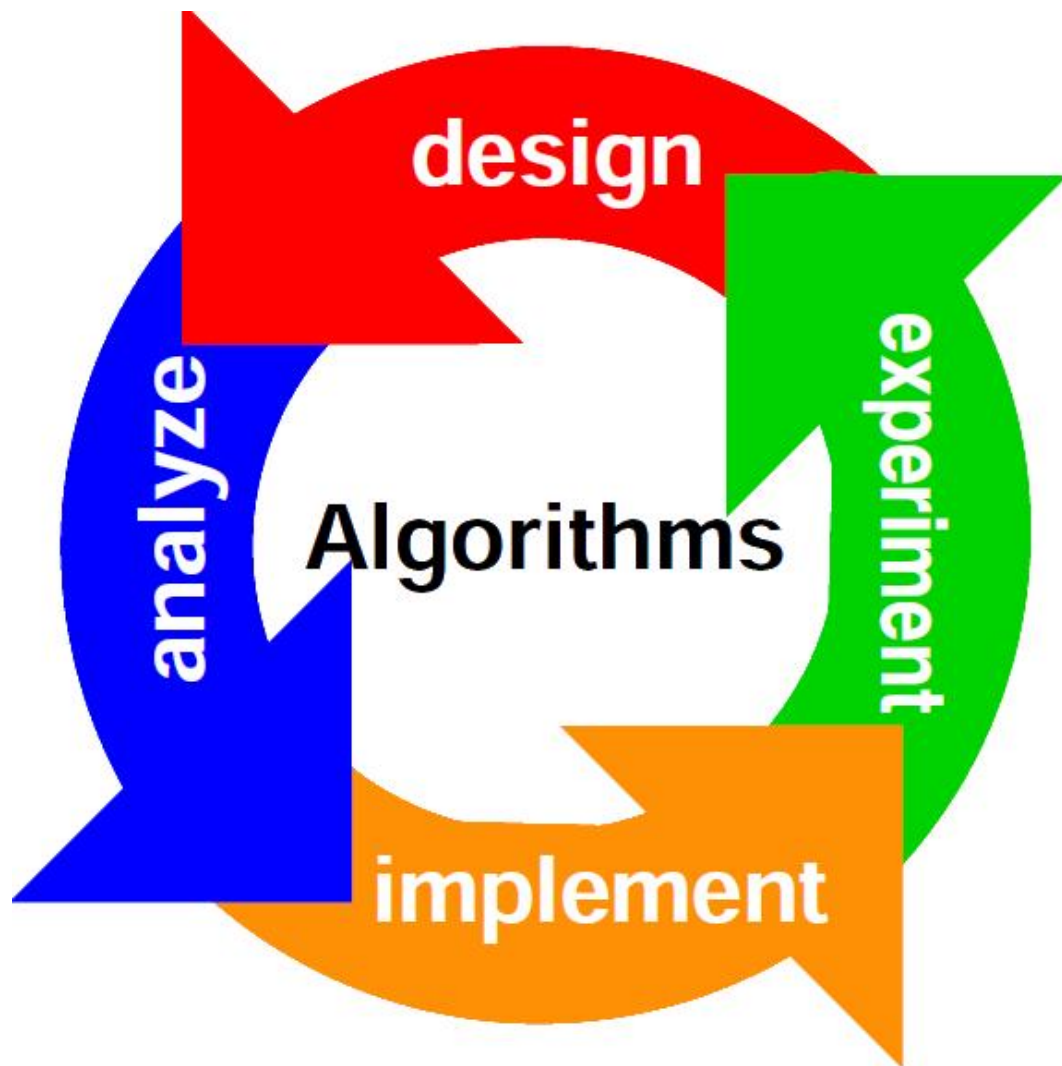


ALGORITMA DAN PEMROGRAMAN 2



TEKNIK PERANGKAT LUNAK
FT - UNPAM

3. PENGURUTAN (SORTING) lanjutan 1

II. METODE PENGURUTAN SELEKSI (SELECTION SORT)

Gagasan metode ini adalah memilih elemen maksimum/minimum dari array, lalu menempatkan elemen maksimum/minimum itu pada ujung (awal atau akhir) array. Selanjutnya elemen terujung tersebut diisolasi dan tidak disertakan pada proses selanjutnya. Proses yang sama diulang untuk elemen larik yang tersisa. Pemilihan nilai maksimum/ minimum dilakukan pada setiap pass.

Ditinjau dari pemilihan elemen maksimum/minimum, algoritma pengurutan seleksi dibagi menjadi dua:

1. Algoritma pengurutan seleksi-maksimum, yaitu memilih elemen maksimum sebagai basis pengurutan
2. Algoritma pengurutan seleksi-minimum, yaitu memilih elemen minimum sebagai basis pengurutan

Algoritma pengurutan seleksi-maksimum

Untuk memperoleh array dengan jumlah elemen N yang terurut menaik, algoritma pengurutan seleksi-maksimum dapat ditulis secara garis besar sebagai berikut:

Jumlah pass : $n = N - 1$

Jumlah perbandingan : $N(N + 1)/2$

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

- 1) Cari elemen terbesar (maks) mulai dari elemen pertama (ke-0) sampai elemen terakhir (ke-n)
- 2) Pertukarkan maks dengan elemen ke-n
- 3) Kurangi n dengan satu (karena elemen ke-n sudah terurut)

procedure SelectionMax(input/output L : LarikInt, input n : integer)

DEKLARASI

i : integer {pencacah pass}
j : integer {pencacah untuk mencari nilai maksimum}
imaks : integer {indeks yg berisi nilai maksimum sementara}
tmp : integer {variabel bantu untuk pertukaran}

DESKRIPSI:

```
for i ← n downto 1 do  
    imaks ← 0  
    for j ← 1 to i do  
        if L[j] > L[imaks] then  
            imaks ← j  
        endif  
    endfor  
    tmp ← L[i]  
    L[i] ← L[imaks]  
    L[imaks] ← tmp  
endfor
```

Contoh:

Urutkan elemen larik berikut secara menaik:

Elemen :	25	27	10	8	76	21
Index :	0	1	2	3	4	5

Pass 1 :

Cari elemen maksimum didalam array L[0...5] → hasilnya : maks = L[4] = 76. Pertukarkan maks dengan L[n], diperoleh:

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

Pass 2 (berdasarkan susunan larik hasil pass 1) :

Cari elemen maksimum didalam array $L[0...4] \rightarrow$ hasilnya : maks = $L[1] = 27$. Pertukarkan maks dengan $L[4]$, diperoleh:

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

Pass 3(berdasarkan susunan larik hasil pass 2) :

Cari elemen maksimum didalam array $L[0...3] \rightarrow$ hasilnya : maks = $L[0] = 25$. Pertukarkan maks dengan $L[3]$, diperoleh:

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

Pass 4 (berdasarkan susunan larik hasil pass 3) :

Cari elemen maksimum didalam array $L[0...2] \rightarrow$ hasilnya : maks = $L[1] = 21$. Pertukarkan maks dengan $L[2]$, diperoleh:

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

Pass 5 (berdasarkan susunan larik hasil pass 4) :

Cari elemen maksimum didalam array $L[0...1] \rightarrow$ hasilnya : maks = $L[1] = 10$. Pertukarkan maks dengan $L[1]$, diperoleh:

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

Tersisa satu elemen (yaitu 8), maka pengurutan selesai. Array sudah terurut naik.

Translasi algoritma pengurutan seleksi-maksimum kedalam bahasa C :

<pre> /*Mengurutkan data dengan metode Seleksi Maksimum*/ #include<iostream.h> void SeleksiMaksimum(int data[],int n); /*prototipe fungsi*/ void main(void) { int i; int n=9;//Index terbesar </pre>	<pre> void SeleksiMaksimum(int array1[],int n) { int i,j,tmp,imaks; for(i=n;i>=1;i--) { imaks=0; for(j=1;j<=i;j++) { </pre>
--	---

<pre> 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; SeleksiMaksimum(data,n); cout<<"Setelah diurutkan"<<endl; for(i=0;i<=n;i++) cout<<data[i]<<" "; cout<<endl; } </pre>	<pre> if(array1[j]>array1[imaks]) imaks=j; } tmp=array1[imaks]; array1[imaks]=array1[i]; array1[i]=tmp; } } </pre>
---	---

Algoritma pengurutan-minimum menaik :

Basis pencarian adalah elemen minimum (terkecil). Elemen minimum ditempatkan di awal larik.

Algoritma seleksi-minimum untuk memperoleh larik yang terurut menaik sbb :

Untuk setiap pass $i = 0, 1, 2, \dots, n-1$, lakukan

- 1) Cari elemen terkecil (min) mulai dari elemen ke i sampai elemen ke n .
- 2) Pertukarkan min dengan elemen ke i .

procedure SelectionMin(input/output L : LarikInt, input n : integer)

DEKLARASI

i : integer {pencacah pass}
 j : integer {pencacah untuk mencari nilai minimum}
 $imin$: integer {indeks yg berisi nilai minimum sementara}
 tmp : integer {variabel bantu untuk pertukaran}

DESKRIPSI:

for $i \leftarrow 0$ to $n-1$ do
 $imin \leftarrow 0$
 for $j \leftarrow i+1$ to n do

```

    if L[j] < L[imin] then
        imin ← j
    endif
endfor
tmp ← L[i]
L[i] ← L[imin]
L[imin] ← tmp
endfor

```

Contoh:

Urutkan elemen larik berikut secara menaik:

Elemen :

29	27	10	8	76	21
----	----	----	---	----	----

 Index : 0 1 2 3 4 5

Pass 0 :

Cari elemen terkecil didalam array L[0...5] → hasilnya : min = L[3] = 8.
 Pertukarkan min dengan L[0], diperoleh:

8	27	10	29	76	21
---	----	----	----	----	----

Pass 1 (berdasarkan susunan larik hasil pass 0) :

Cari elemen terkecil didalam array L[1...5] → hasilnya : min = L[2] = 10.
 Pertukarkan min dengan L[1], diperoleh:

8	10	27	29	76	21
---	----	----	----	----	----

Pass 2 (berdasarkan susunan larik hasil pass 1) :

Cari elemen terkecil didalam array L[2...5] → hasilnya : min = L[5] = 21.
 Pertukarkan min dengan L[2], diperoleh:

8	10	21	29	76	27
---	----	----	----	----	----

Pass 3 (berdasarkan susunan larik hasil pass 2) :

Cari elemen terkecil didalam array L[3...5] → hasilnya : min = L[5] = 27.
 Pertukarkan min dengan L[3], diperoleh:

8	10	21	27	76	29
---	----	----	----	----	----

Pass 4 (berdasarkan susunan larik hasil pass 3) :

Cari elemen terkecil didalam array L[4...5] → hasilnya : min = L[5] = 29.

Pertukarkan min dengan L[4], diperoleh:

8	10	21	27	29	76
---	----	----	----	----	----

Tersisa satu elemen (yaitu 76), maka pengurutan selesai. Array sudah terurut naik.

Translasi algoritma pengurutan seleksi-minimum kedalam bahasa C :

```
/*Mengurutkan data dengan metode Seleksi
Minimum*/
```

```
#include<iostream.h>
```

```
void SeleksiMinimum(int data[],int n);
```

```
/*prototipe fungsi*/
```

```
void main(void)
```

```
{
```

```
    int i;
```

```
    int n=9;//Index terbesar
```

```
    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;
```

```
    SeleksiMinimum(data,n);
```

```
    cout<<"Setelah diurutkan"<<endl;
```

```
    for(i=0;i<=n;i++)
```

```
        cout<<data[i]<<" ";
```

```
void
```

```
SeleksiMinimum(int
```

```
array1[],int n)
```

```
{
```

```
    int i,j,tmp,imin;
```

```
    for(i=0;i<=n-1;i++)
```

```
    {
```

```
        imin=i;
```

```
        for(j=i+1;j<=n;j++)
```

```
        {
```

```
            if(array1[j]<array
1[imin])
```

```
                imin=j;
```

```
        }
```

```
        tmp=array1[imin];
```

```
        array1[imin]=arra
```

<pre> cout<<endl; } </pre>	<pre> y1[i]; array1[i]=tmp; } } </pre>
----------------------------------	---

TUGAS

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

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