

SORTING ALGORITMA DAN STRKTUR DATA II



Institut Teknologi Sumatera

TUJUAN PERKULIAHAN

- Mahasiswa mampu memahami algoritma pengurutan
- Mahasiswa mampu menerapkan algoritma pengurutan dengan bahasa pemrograman C++
- Mahasiwa mampu menyelesaikan masalah yang memerlukan pengurutan sebagai algoritmanya

PRE TEST

Gunakan perulangan bersarang untuk menggambarkan segita bintang berikut

- Sebutkan yang dimaksud dengan pointer! Apa kegunaannya dalam suatu algoritma!
- Sebutkan teknik-teknik passing pada sub program, sebutkan perbedaannya!

PENGURUTAN

- Suatu proses (operasi) yang mengurutkan data sehingga tersusun teratur menurut aturan tertentu.
- Terdapat dua jenis pengurutan, yaitu: increasing (naik) atau decreasing (turun).

PENGURUTAN

Data Acak

Ascending

Descending

5	6	8	1	3	25	10
1	3	5	6	8	10	25
25	10	8	6	5	3	1

KEGUNAAN PENGURUTAN

- Membantu proses pencarian (searching)
- Menyelesaikan masalah-masalah kompleks seperti penjadwalan (scheduling), pengolahan basis data, riset operasi, dsb.

JENIS PENGURUTAN

- Pengurutan data internal
 - Data sedikit sehingga proses dapat dilakukan dalam main memory.
- Pengurutan data eksternal
 - Data besar, koleksi data tersebut ada dalam auxiliary memory device seperti magnetic tape atau disk.

ALGORITMA PENGURUTAN

Terdapat banyak metode untuk melakukan pengurutan, diantaranya:

- Bubble Sort
- Selection Sort
- Insertion Sort
- Shell Sort
- Merge Sort
- Quick Sort
- Dsb.

 Disebut pengurutan gelembung, menggunakan terminologi gelembung karena berat jenis nya yg lebih ringan dari air.

- IDE UTAMA dari metode ini yaitu melakukan pemeriksaan dari dua buah elemen yang letaknya bersebelahan, dan menukarnya sampai dengan kondisi urut yang diinginkan.
- Bubble sort mudah dipahami, tapi kurang efisien.
- Kompleksitas n data acak adalah $\mathcal{O}(n^2)$

- Bubble sort memiliki n-1 fase. Setiap fase melakukan pemeriksaan dan pertukaran.
- Berikut ini terdapat array dengan 7 elemen belum terurut. Array tersebut akan diurut menaik.

8	6	5	1	თ	25	10
0	1	2	3	4	5	6

Fase i=1, perulangan pertama. Perulangan dari n-1 s.d n > i;

j	Elemen dibandingkan	Nilai dibandingkan	Tukar?	Hasil sementara
6	A[6] < A[5]	10 < 25	Ya	8 6 5 1 3 10 25
5	A[5] < A[4]	10 < 3	Tidak	8 6 5 1 3 10 25
4	A[4] < A[3]	3 < 1?	Tidak	8 6 5 1 3 10 25
3	A[3] < A[2]	1 < 5?	Ya	8 6 1 5 3 10 25
2	A[2] < A[1]	1 < 6?	Ya	8 1 6 5 3 10 25
1	A[1] < A[0]	1 < 8?	Ya	1 8 6 5 3 10 25

State Akhir Fase ke-1

1	8	6	5	3	10	25
0	1	2	3	4	5	6
			Δ			

• Fase i=2, perulangan kedua. Perulangan dari n-1 s.d n > i;

j	Elemen dibandingkan	Nilai dibandingkan	Tukar?	Hasil sementara
6	A[6] < A[5]	25 < 10	Tidak	1 8 6 5 3 10 25
5	A[5] < A[4]	10 < 3	Tidak	1 8 6 5 3 10 25
4	A[4] < A[3]	3 < 5	Ya	1 8 6 3 5 10 25
3	A[3] < A[2]	3 < 6	Ya	1 8 3 6 5 10 25
2	A[2] < A[1]	3 < 8	Ya	1 3 8 6 5 10 25

State Akhir Fase ke-2

1	3	8	6	5	10	25
0	1	2	3	4	5	6
			A			

Fase i=3, perulangan ketiga. Perulangan dari n-1 s.d n > i;

j	Elemen dibandingkan	Nilai dibandingkan	Tukar?	Hasil sementara
6	A[6] < A[5]	25 < 10	Tidak	1 3 8 6 5 10 25
5	A[5] < A[4]	10 < 5	Tidak	1 3 8 6 5 10 25
4	A[4] < A[3]	5 < 6	Ya	1 3 8 5 6 10 25
3	A[3] < A[2]	5 < 8	Ya	1 3 5 8 6 10 25

State Akhir Fase ke-3

1	3	5	8	6	10	25
0	1	2	3	4	5	6
			A			

• Fase i=4, perulangan keempat. Perulangan dari n-1 s.d n > i;

j	Elemen dibandingkan	Nilai dibandingkan	Tukar?	Hasil sementara
6	A[6] < A[5]	25 < 10	Tidak	1 3 5 8 6 10 25
5	A[5] < A[4]	10 < 6	Tidak	1 3 5 8 6 10 25
4	A[4] < A[3]	6 < 8	Ya	1 3 5 6 8 10 25

State Akhir Fase ke-4

1	3	5	6	8	10	25
0	1	2	3	4	5	6
			A			

• Fase i=5, perulangan kelima. Perulangan dari n-1 s.d n > i;

j	Elemen dibandingkan	Nilai dibandingkan	Tukar?	Hasil sementara
6	A[6] < A[5]	25 < 10	Tidak	1 3 5 6 8 10 25
5	A[5] < A[4]	10 < 8	Tidak	1 3 5 6 8 10 25

State Akhir Fase ke-5

1	3	5	6	8	10	25
0	1	2	3	4	5	6
			A			

• Fase i=6, perulangan keenam. Perulangan dari n-1 s.d n > i;

j	Elemen dibandingkan	Nilai dibandingkan	Tukar?	Hasil sementara
6	A[6] < A[5]	25 < 10	Tidak	1 3 5 6 8 10 25

State Akhir Fase ke-6

1	3	5	6	8	10	25
0	1	2	3	4	5	6
			A			

LATIHAN SIMULASI BUBBLE SORT

 Menggunakan algoritma Bubble Sort, simulasikan pengurutan menaik pada deret berikut.

45	89	5	1	6	25	2
0	1	2	3	4	5	6

ALGORITMA BUBBLE SORT

```
#include <iostream>
using namespace std;

void tukar(int *a, int *b){
   int tmp;
   tmp = *a;
   *a = *b;
   *b = tmp;
}
```

```
int i, j, n=7;
int A[n] = \{5,6,8,1,3,25,10\};
for(i=0; i < n; i++) {
   for(j=n-1; j > i; j--){
      if(A[j] < A[j-1]){
         tukar(&A[j-1], &A[j]);
cout << endl << "Setelah pengurutan : ";</pre>
for(i=0; i < n; i++) {
   cout << A[i] << " ";</pre>
```

int main(){

LATIHAN SIMULASI BUBBLE SORT

 Menggunakan algoritma Bubble Sort, simulasikan pengurutan menurun pada deret berikut.

6	33	5	1	26	10	4
0	1	2	3	4	5	6

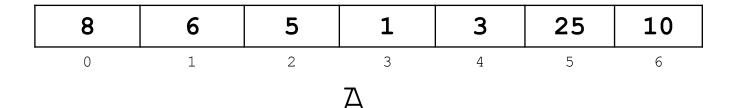
X

LATIHAN SIMULASI BUBBLE SORT

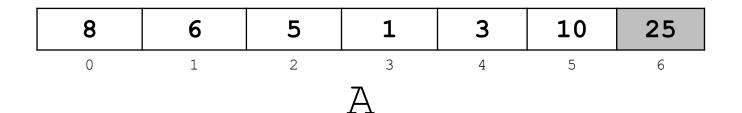
Menggunakan deret pada slide sebelumnya, buatlah algoritma dan program pengurutan Bubble Sort menurun!

- IDE UTAMA dari algoritma pengurutan ini adalah memilih elemen ekstrem (maksimum atau minimum) dari sebuah deret. Kemudian menempatkan elemen ekstrem tersebut pada bagian awal atau akhir deret.
- Elemen ekstrem tersebut kemudian diisolasi dan tidak disertakan pada proses selanjutnya.
- Pencarian elemen ekstrem dilanjutkan pada deret yang tersisa.

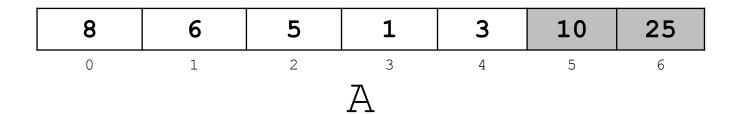
- Lakukan pengurutan menaik pada array A berikut.
- Elemen dengan nilai terbesar akan ada disebelah kanan.
 Sedangkan elemen dengan nilai terkecil akan ada disebelah kiri.



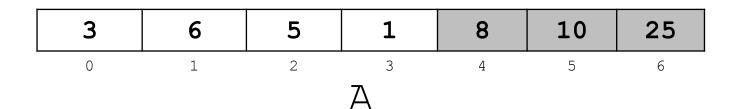
- Cari elemen maksimum pada array A.
- Hasilnya idx maks = Array A index ke-5 = 25.
- Tukar $A[n-1] \leftrightarrow A[idx maks]$



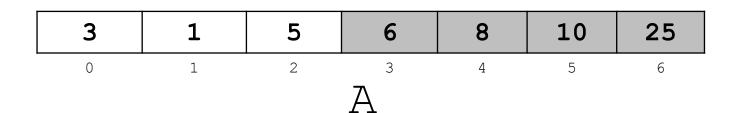
- Cari elemen maksimum pada array A.
- Hasilnya idx maks = Array A index ke-5 = 10.
- Tukar $A[n-2] \leftrightarrow A[idx maks]$



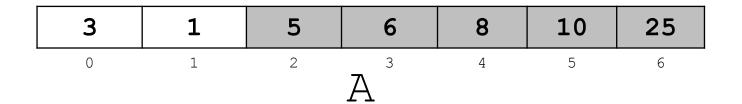
- Cari elemen maksimum pada array A.
- Hasilnya idx maks = Array A index ke-0 = 8.
- Tukar $A[n-3] \leftrightarrow A[idx maks]$



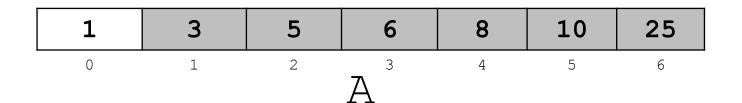
- Cari elemen maksimum pada array A.
- Hasilnya idx maks = Array A index ke-1 = 6.
- Tukar $A[n-4] \leftrightarrow A[idx maks]$



- Cari elemen maksimum pada array A.
- Hasilnya idx maks = Array A index ke-2 = 5.
- Tukar $A[n-5] \leftrightarrow A[idx maks]$

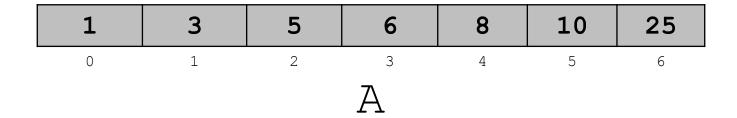


- Cari elemen maksimum pada array A.
- Hasilnya idx maks = Array A index ke-0 = 3.
- Tukar $A[n-6] \leftrightarrow A[idx maks]$



Fase 7

• Tersisa satu buah elemen, maka pengurutan selesai.



ALGORITMA SELECTION SORT

```
#include <iostream>
using namespace std;

void tukar(int *a, int *b){
   int tmp;
   tmp = *a;
   *a = *b;
   *b = tmp;
}
```

```
int main(){
   int i, j, n=7;
   int A[n] = \{5,6,8,1,3,25,10\};
   for (i=n-1; i > 0; i--) {
      idx maks = 0;
      for(j=1; j <= i; j++){</pre>
          if(A[j] > A[idx maks]){
             idx maks = j;
      tukar(&A[i],&A[idx maks]);
   cout << endl << "Setelah pengurutan : ";</pre>
   for(i=0; i < n; i++) {
      cout << A[i] << " ";</pre>
```

LATIHAN SIMULASI SELECTION SORT

 Menggunakan algoritma Selection Sort, simulasikan pengurutan menaik pada deret berikut.

6	33	5	1	26	10	4
0	1	2	3	4	5	6

X

LATIHAN SIMULASI BUBBLE SORT

Menggunakan deret pada slide sebelumnya, buatlah algoritma dan program pengurutan Selection Sort menurun!

PENGURUTAN PADA RECORD

Dimiliki record sebagai berikut.

```
struct buku {
    string isbn;
    string judul;
    int hal;
};
```

- Apabila dalam record tersebut terdapat N buah data, urutkan data yang ada berdasarkan jumlah halamannya secara menaik.
- Gunakan algoritma favorit anda.

Terima Kasih

PR

- Menggunakan algoritma pengurutan Insertion Sort, lakukan pengurutan terhadap N buah record yang berasal dari pengguna.
- Gunakan satu buah prosedur saja untuk melakukan pengurutan naik, atau menurun.