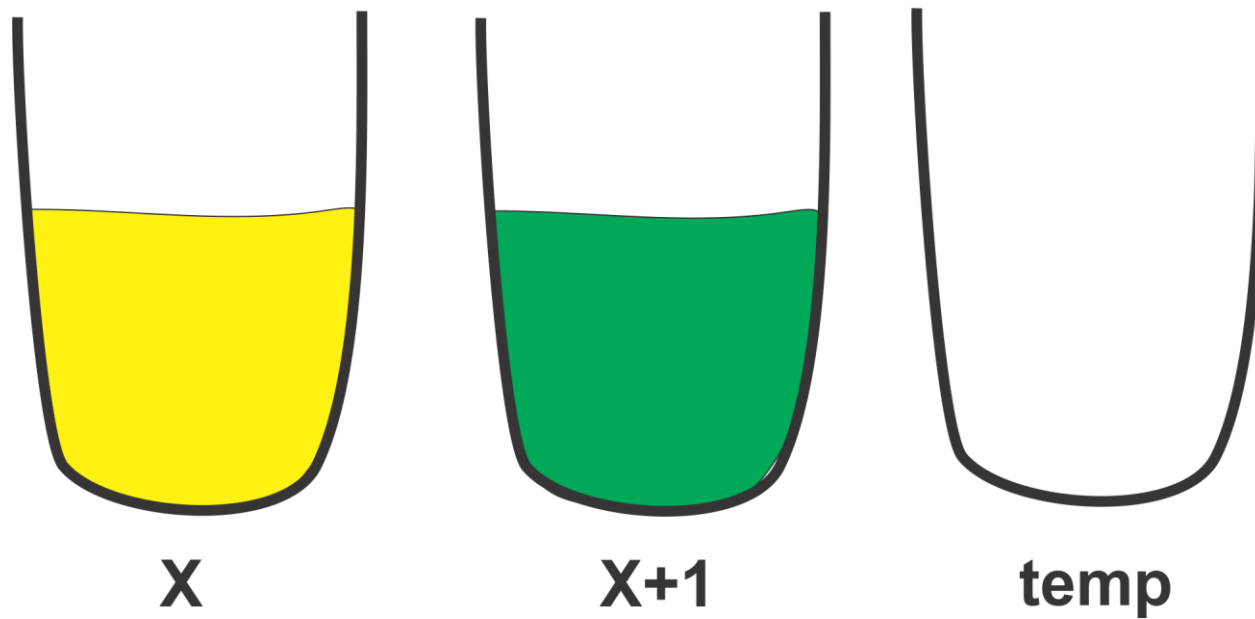


PENGURUTAN DATA

OLEH : AKHI HARUNI

A solid green horizontal bar at the bottom of the slide.

Bagaimana cara menukar **gelas x** dengan **gelas x+1**
agar isinya tidak tertukar?



Jawabannya adalah menggunakan bantuan gelas kosong (yang biasa kita namai temp)

Jadi proses pemindahannya seperti ini ...

temp atau gelas kosong dituangi isi salah satu gelas (misal X). Gelas yang kosong (X) dituangi isi gelas yang lain (X+1). Terakhir, gelas yang lain (X+1) dituangi isi dari gelas temp.

(1) temp = x;

(2) x = x + 1;

(3) x + 1 = temp;

Konsep di atas akan kita gunakan dalam pengurutan data

Mengapa harus diurutkan?

Data diurutkan tak lain agar lebih mudah diproses. Apabila data sudah urut, kita dapat menggunakannya untuk berbagai keperluan misal perankingan, penentuan remedial, dan lain-lain.

Apa saja tahapannya?

- Pertama, buat sekumpulan data. Tidak akan pernah ada proses pengurutan data jika datanya hanya satu sehingga kita perlu membuat sekumpulan data terlebih dahulu.
- Kedua, melakukan proses penukaran data dengan konsep pada slide sebelumnya
- Menampilkan hasil pengurutan data

Metode Bubble Sort

Bubble artinya gelembung, makin ringan makin naik. Demikian yang terjadi pada algoritma bubble sort. Akan ada pergeseran-pergeseran sesuai perintah (ascending/descending). Apabila descending, maka data akan diurutkan dari besar ke kecil. Apabila ascending, maka data akan diurutkan dari kecil ke besar.

Tidak hanya metode Bubble Sort saja dalam algoritma pengurutan data. Coba kamu cari tau tentang metode **Selection Sort** dalam pengurutan data. Apa perbedaannya dengan Bubble Sort?

1. Membuat sekumpulan data

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     //membuat sekumpulan data
6     int nilaipemdas[5], a, b, temp;
7     for(a=0; a<5; a++) {
8         cout << "Masukkan nilai pemdas : ";
9         cin >> nilaipemdas[a];
10    }
11
```

2. Mengurutkan data

```
11
12 //mengurutkan data dengan menukar data
13 = for(a=0; a<5; a++) {
14 =     for(b=0; b<5; b++) {
15 =         if (nilaipemdas[b] > nilaipemdas[b+1]) {
16             temp = nilaipemdas[b];
17             nilaipemdas[b] = nilaipemdas[b+1];
18             nilaipemdas[b+1] = temp;
19         }
20     }
21 }
22
```

Penjelasan kode program pada proses kedua

- ❑ `for (a=0; a<5; a++)` merupakan perulangan pertama → untuk jumlah data
- ❑ `for (b=0; b<5; a++)` merupakan perulangan kedua → untuk jumlah iterasi (tahapan)
- ❑ `if (nilaipemdas[b] > nilaipemdas[b+1])` maksudnya adalah..
 - ❑ apabila index 0 lebih besar dari index 1, maka data ditukar
 - ❑ misal index 0 (data pertama) diisi 9, index 1 (data kedua) diisi 8, maka dalam pengurutan ascending (kecil ke besar), data akan ditukar menjadi 8 pada index ke-0 dan 9 pada index ke-1
- ❑ `temp = nilaipemdas[b];` dan seterusnya merupakan proses penukaran data. Dalam kurung siku nilai pemdas, bukan lagi 5 melainkan **b**. Mengapa **b**? karena variabel perulangan terdekat yang digunakan adalah **b** → `for (b)`

3. Menampilkan hasil pengurutan

```
22
23 //menampilkan hasil pengurutan data
24 cout << "Hasil pengurutannya adalah : " << endl;
25 for(a=0; a<5; a++) {
26     cout << nilaipemdas[a] << endl;
27 }
28 }
29
```

compile and run

ASCENDING

DESCENDING



```
Masukkan nilai pendas : 8
Masukkan nilai pendas : 6
Masukkan nilai pendas : 10
Masukkan nilai pendas : 7
Masukkan nilai pendas : 9
Hasil pengurutannya adalah :
6
7
8
9
10
```



```
Masukkan nilai pendas : 9
Masukkan nilai pendas : 10
Masukkan nilai pendas : 7
Masukkan nilai pendas : 8
Masukkan nilai pendas : 6
Hasil pengurutannya adalah :
10
9
8
7
6
```