

Nama : Kania Meliana Fityanti

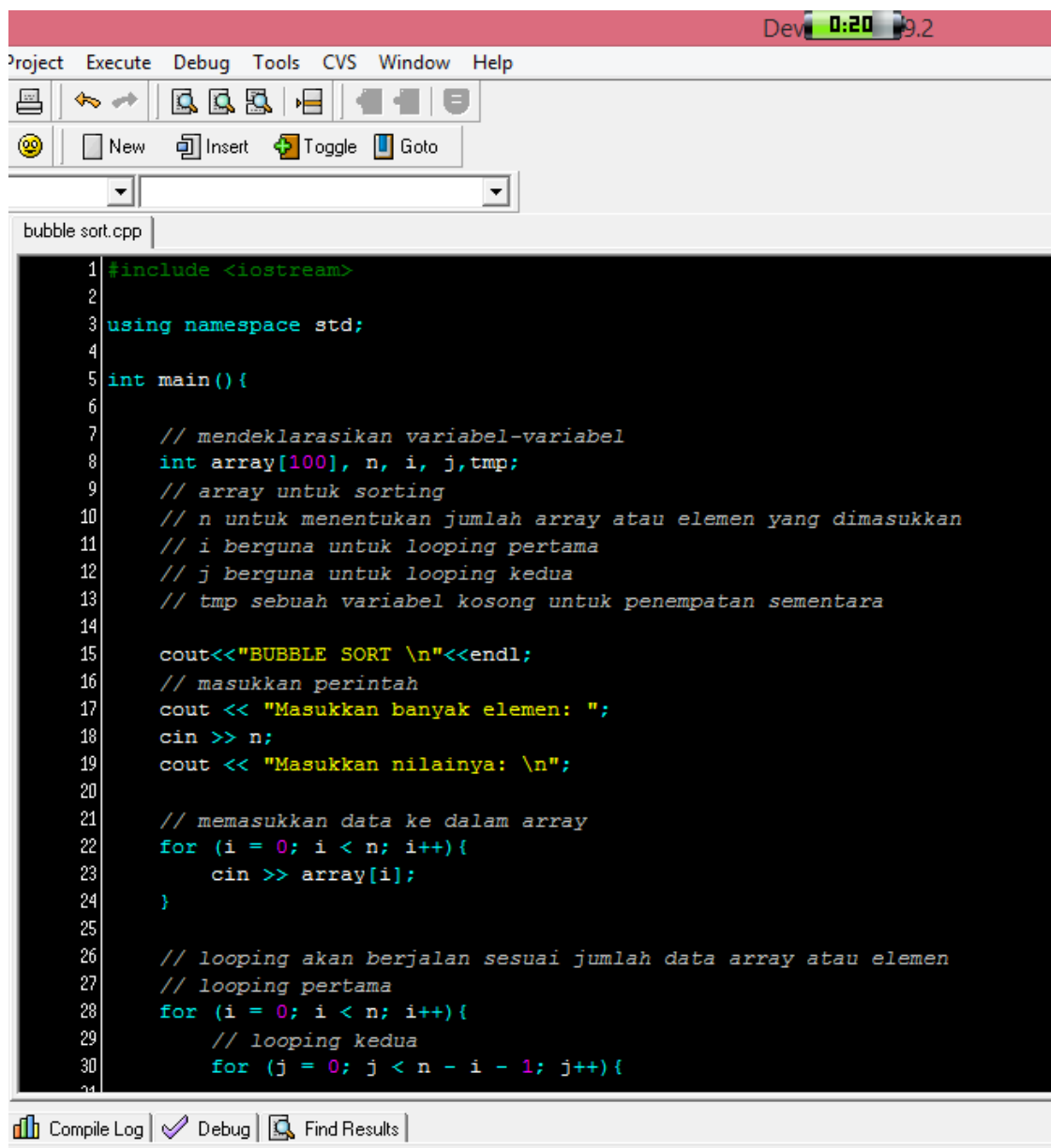
NIM : 21091397028

Kelas : 2021B

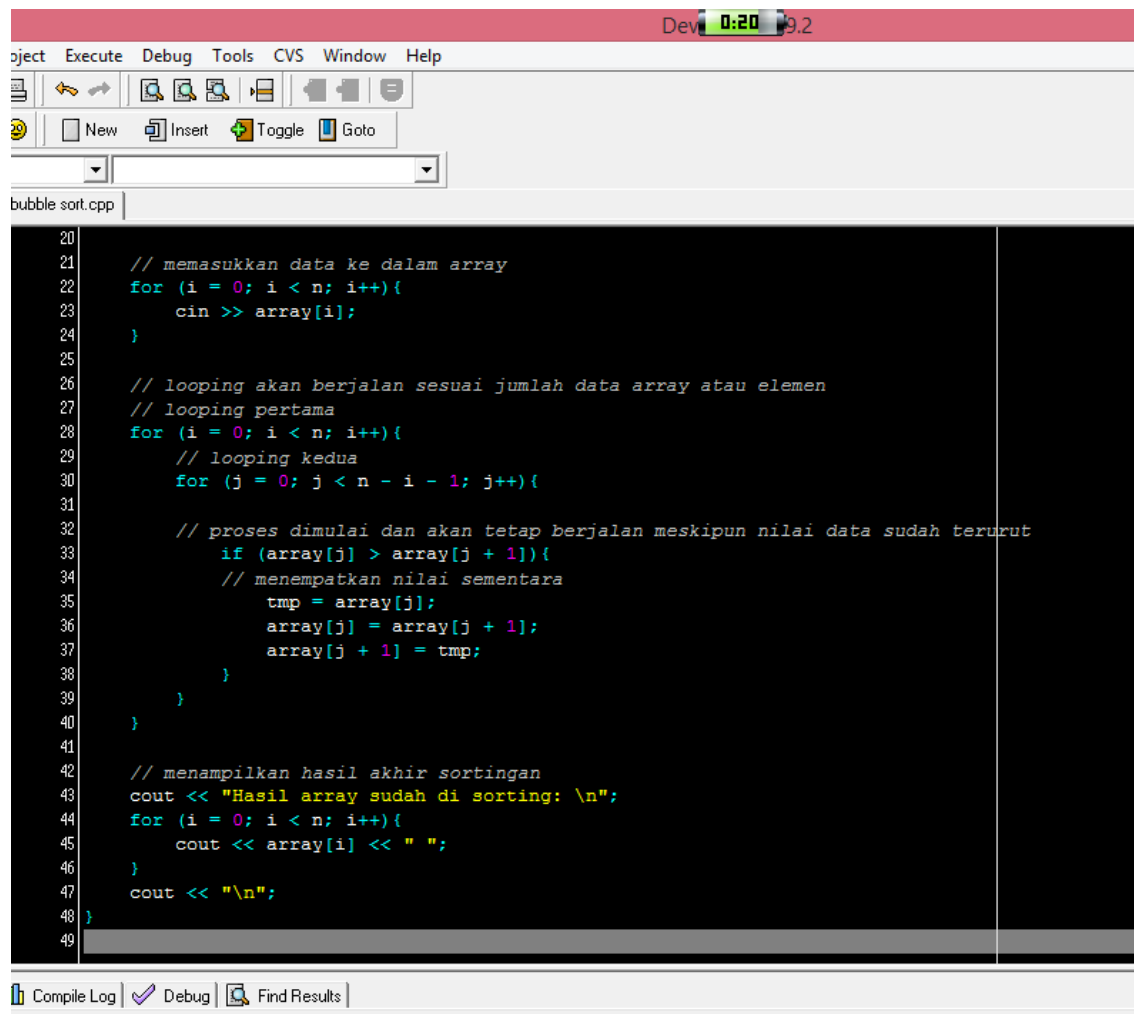
BUBBLE SORT

Bubble Sort adalah metode pengurutan data dengan cara melakukan penukaran data dari data pertama dengan data di sebelahnya secara terus menerus sampai bisa dipastikan dalam suatu iterasi tertentu tidak ada lagi perubahan atau penukaran.

Codingan Bubble Sort dan penjelasannya



```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6
7     // mendeklarasikan variabel-variabel
8     int array[100], n, i, j, tmp;
9     // array untuk sorting
10    // n untuk menentukan jumlah array atau elemen yang dimasukkan
11    // i berguna untuk looping pertama
12    // j berguna untuk looping kedua
13    // tmp sebuah variabel kosong untuk penempatan sementara
14
15    cout<<"BUBBLE SORT \n"<<endl;
16    // masukkan perintah
17    cout << "Masukkan banyak elemen: ";
18    cin >> n;
19    cout << "Masukkan nilainya: \n";
20
21    // memasukkan data ke dalam array
22    for (i = 0; i < n; i++){
23        cin >> array[i];
24    }
25
26    // looping akan berjalan sesuai jumlah data array atau elemen
27    // looping pertama
28    for (i = 0; i < n; i++){
29        // looping kedua
30        for (j = 0; j < n - i - 1; j++){
```

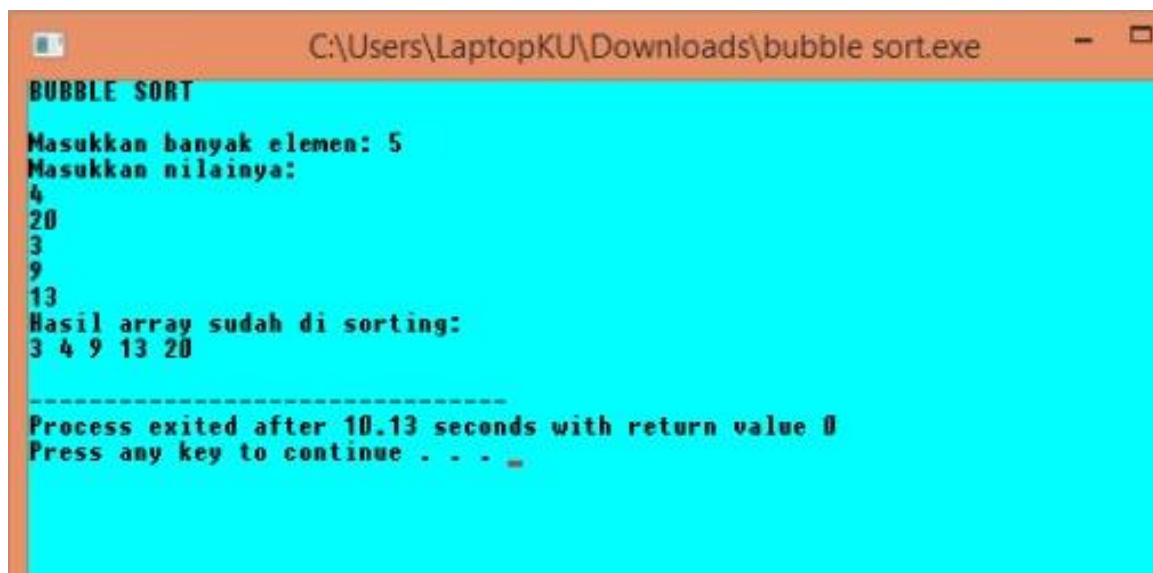


The image shows a C++ IDE window titled 'Dev 0.20 9.2'. The menu bar includes 'Project', 'Execute', 'Debug', 'Tools', 'CVS', 'Window', and 'Help'. The toolbar contains icons for 'New', 'Insert', 'Toggle', and 'Goto'. The file name 'bubble sort.cpp' is displayed in the editor's title bar. The code is as follows:

```
20
21 // memasukkan data ke dalam array
22 for (i = 0; i < n; i++){
23     cin >> array[i];
24 }
25
26 // looping akan berjalan sesuai jumlah data array atau elemen
27 // looping pertama
28 for (i = 0; i < n; i++){
29     // looping kedua
30     for (j = 0; j < n - i - 1; j++){
31
32         // proses dimulai dan akan tetap berjalan meskipun nilai data sudah terurut
33         if (array[j] > array[j + 1]){
34             // menempatkan nilai sementara
35             tmp = array[j];
36             array[j] = array[j + 1];
37             array[j + 1] = tmp;
38         }
39     }
40 }
41
42 // menampilkan hasil akhir sortiran
43 cout << "Hasil array sudah di sorting: \n";
44 for (i = 0; i < n; i++){
45     cout << array[i] << " ";
46 }
47 cout << "\n";
48 }
49
```

The IDE status bar at the bottom shows 'Compile Log', 'Debug', and 'Find Results'.

Output



The image shows the output of the bubble sort program in a window titled 'C:\Users\LaptopKU\Downloads\bubble sort.exe'. The output is as follows:

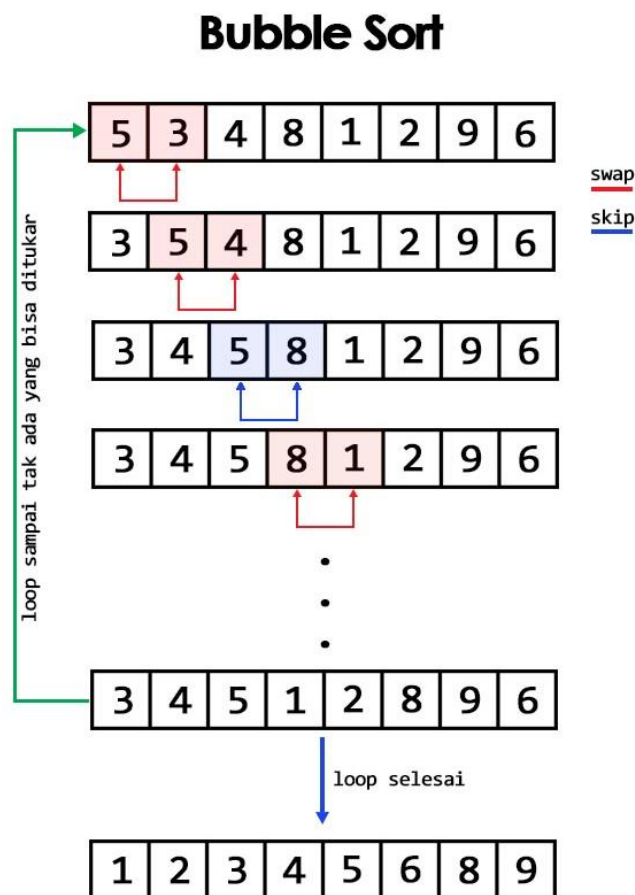
```
BUBBLE SORT
Masukkan banyak elemen: 5
Masukkan nilainya:
4
20
3
9
13
Hasil array sudah di sorting:
3 4 9 13 20

-----
Process exited after 10.13 seconds with return value 0
Press any key to continue . . .
```

Prinsip kerja dari bubble sort adalah :

1. Pengecekan mulai dari data ke satu sampai data ke-n
2. Bandingkan data ke-n dengan data sebelumnya (n-1)
3. Jika lebih kecil maka pindahkan bilang tersebut dengan bilang yang ada didepannya (sebelumnya) satu persatu (n-1,n-2,n-3,...dts)
4. Jika lebih besar maka tidak terjadi permindahan
5. Ulang langkah 2 dan 3 s/d sort optimal.

Contohnya,



Kelebihan Bubble Sort :

- Proses penghitungan Bubble sort merupakan metode yang paling sederhana
- Algoritma Bubble Sort mudah dipahami
- Langkah atau tahapan dalam pengurutan data sangat sederhana.

Kekurangan Bubble Sort :

- Proses penghitungan Bubble Sort menggunakan metode pengurutan termasuk paling tidak efisien walaupun dianggap sederhana. Karena proses

pengurutan data dilakukan dengan tahapan satu - satu, mulai dari data paling awal sebelah kiri, sampai data terakhir

- Ketika data yang kita punya banyak atau dalam jumlah yang besar, maka proses penghitungan akan semakin lama dan lambat. Karena proses pengurutan data secara tunggal (satu - satu).
- Jumlah pengulangan akan tetap sama sampai ke data yang terakhir, walaupun sebagian data yang ada telah terurut.