

Contoh Praktikum Algoritma dan Pemrograman



Nama : Agil Deriansyah Hasan
Nim : 4522210125

Dosen:

Dra.SRI REZEKI CANDRA NURSARI,M.Kom
Prak. Algoritma dan Pemrograman - B

S1-Teknik Informatika
Fakultas Teknik
Universitas Pancasila 2023/2024

Sort

The image shows a C++ program for sorting an array. The code is written in Notepad++ and the output is shown in the Command Prompt. The program uses a selection sort algorithm. It starts by including `<iostream>` and `<string.h>`, and defining `agil_n` as 9. It then uses `using namespace std;` and declares variables `agil_i`, `agil_j`, `agil_k`, and `agil_x`. The `agil1()` function initializes the array `agil_a` with the values {23, 17, 14, 6, 13, 10, 1, 5, 7}. The `agil2()` function prints the array before sorting. The `agil3()` function prints the array after sorting. The `main()` function calls `agil1()`, `agil2()`, and `agil3()`.

```
#include <iostream>
#include <string.h>
#define agil_n 9

using namespace std;

int agil_i, agil_j, agil_k, agil_x;

void agil1()
{
    int agil_a[agil_n] = {23, 17, 14, 6, 13, 10, 1, 5, 7};
    cout << "Sebelum Dilakukan Pengurutan" << endl;
    cout << "<=====>" << endl;
    cout << endl;
    for (agil_i=0; agil_i<=agil_n-1; agil_i++)
        cout << agil_a[agil_i] << " ";
    cout << endl;
}

void agil2()
{
    int agil_a[agil_n];
    cout << "Step By Step Dilakukan Pengurutan" << endl;
    cout << "<=====>" << endl;
    agil_k=0;
    while(agil_k<=agil_n-2){
        agil_j=agil_k;
        agil_i=agil_k+1;
        while(agil_i<=agil_n-1)
        {
            if(agil_a[agil_i] > agil_a[agil_j])
                agil_j=agil_i;
            cout << agil_a[agil_i] << " ";
            agil_i++;
        }
        cout << endl;
        agil_x = agil_a[agil_j];
        agil_a[agil_j] = agil_a[agil_k];
        agil_a[agil_k] = agil_x;
        agil_k++;
        cout << endl;
    }
}

void agil3()
{
    cout << "Sesudah Dilakukan Pengurutan" << endl;
    cout << "<=====>" << endl;
    cout << endl;
    for (agil_i=0; agil_i<=agil_n-1; agil_i++)
        cout << agil_a[agil_i] << " ";
    cout << endl;
}

int main()
{
    agil1();
    agil2();
    agil3();
    cout << endl;
    return 0; }
```

The Command Prompt shows the output of the program. It first displays the array before sorting: 23 17 14 6 13 10 1 5 7. Then it shows the step-by-step sorting process, with the array being rearranged in each iteration. Finally, it shows the array after sorting: 1 5 6 7 10 13 14 17 23.

```
E:\>g++ sort.cpp -o sort.exe
E:\>sort
Sebelum Dilakukan Pengurutan
<=====>
23 17 14 6 13 10 1 5 7

Step By Step Dilakukan Pengurutan
<=====>
17 14 6 13 10 1 5 7
14 6 13 10 1 5 7
6 13 10 1 5 7
13 10 1 5 7
10 1 5 7
1 5 7
5 6
1

Sesudah Dilakukan Pengurutan
<=====>
1 5 6 7 10 13 14 17 23
```

PSEUDOCODE

Mengurutkan data dengan metode Insertion Sort

KAMUS/DEKLARASI VARIABEL

agil_a[agil_n], agil_i, agil_k, agil_x, agil_n : int

Algoritma/Deskripsi

agil_n=9

agil_A[agil_n] = {23, 17, 14, 6, 13, 10, 1, 5, 7}

for(agil_i=0; agil_i<= agil_n-1; agil_i++)

 print(agil_A[agil_i], " ")

endfor

for(agil_k=0; agil_k<= agil_n; agil_k++)

 agil_i= agil_k

 agil_x= agil_A[agil_i]

 while(agil_i> 0 && agil_A[agil_i-1] > agil_x)

 agil_A[agil_i] = agil_A[agil_i-1]

 agil_i—

 endwhile

 agil_A[agil_i]= agil_x

endfor

for(agil_i= 0; agil_i<= agil_n-1; agil_i++)

 print(agil_A[agil_i], " ")

endfor

Algoritma/Bahasa Alami:

- agil_n=9
- Const agil_A[agil_n] = {23, 17, 14, 6, 13, 10, 1, 5, 7}
- agil_i=0
- Selama(agil_i <= agil_n-1) kerjakan baris 5 s.d.7 kalau tidak baris 8
- Mencetak/menampilkan isi variable agil_A[agil_i]
- Mencetak/menampilkan isi variable (" ")
- agil_i++
- agil_k=0
- Selama(agil_k <= agil_n), maka kerjakan baris 10 s.d. 16 kalau tidak baris 17
- agil_i = agil_k
- agil_x= agil_A[agil_i]
- Selama(agil_i>=0 dan agil_A[agil_i-1] > agil_x), maka kerjakan baris 13 s.d.14 kalau tidak baris 15
- agil_A[agil_i]= dhendi_A[dhendi_i-1]
- agil_i—
- agil_A[agil_i]= agil_x
- agil_k++
- Selama(agil_i <= agil_n-1) kerjakan baris 18 s.d.20 kalau tidak baris 21
- Mencetak/menampilkan isi variable agil_A[agil_i]
- Mencetak/menampilkan isi variable (" ")
- agil_i++
- Selesai

insert Sort = Ascending

agil_A:[23, 17, 14, 6, 13, 10, 1, 5, 7]									
Iterasi	Data[1]	Data[2]	Data[3]	Data[4]	Data[5]	Data[6]	Data[7]	Data[8]	Data[9]
Awal	23	17	14	6	13	10	1	5	7
agil_i=1	23	17	14	6	13	10	1	5	7
agil_i=2	1	17	14	6	13	10	23	5	7
agil_i=3	1	5	14	6	13	10	23	17	7
agil_i=4	1	5	6	14	13	10	23	17	7
agil_i=5	1	5	6	7	13	10	23	17	14
agil_i=6	1	5	6	7	10	13	23	17	14
agil_i=7	1	5	6	7	10	13	23	17	14
agil_i=8	1	5	6	7	10	13	14	17	23
Akhir	1	5	6	7	10	13	14	17	23