Contoh Praktikum Algortima dan Pemograman

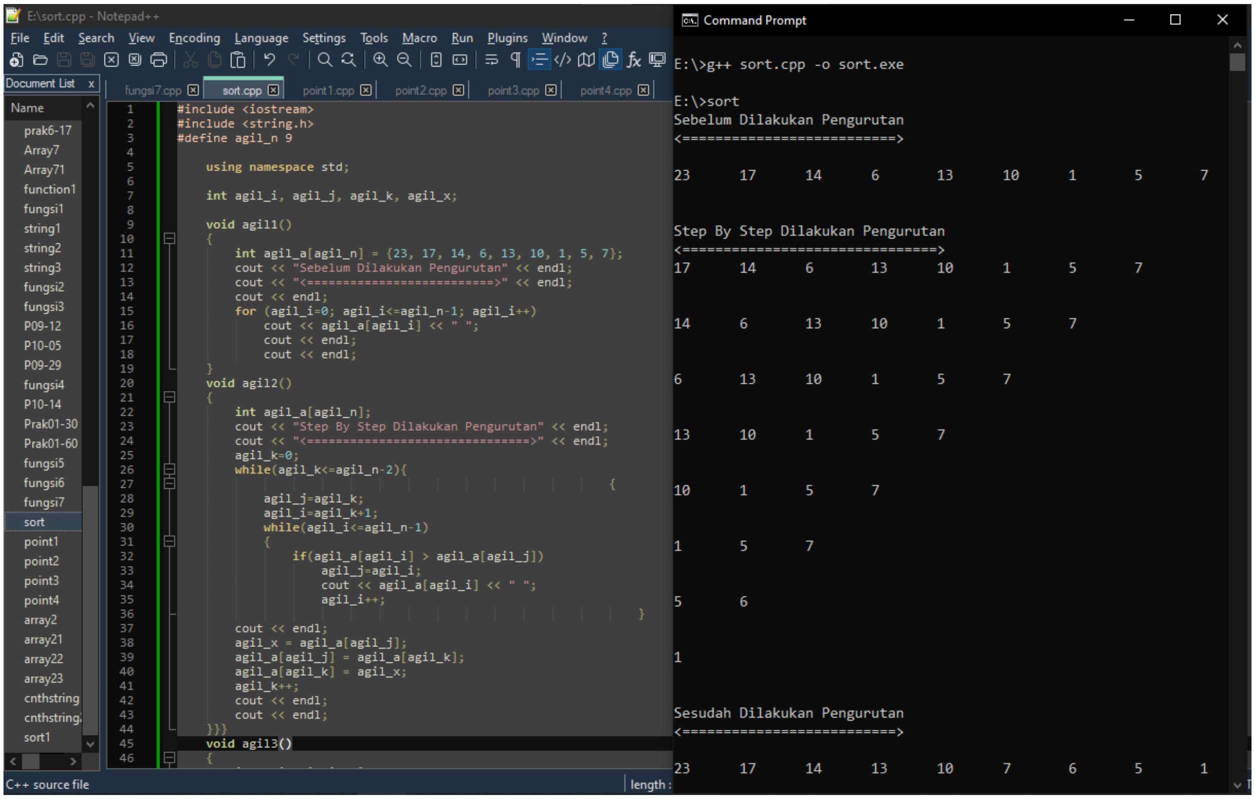
Nama : Agil Deriansyah Hasan
Nim : 4522210125

Dosen:

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

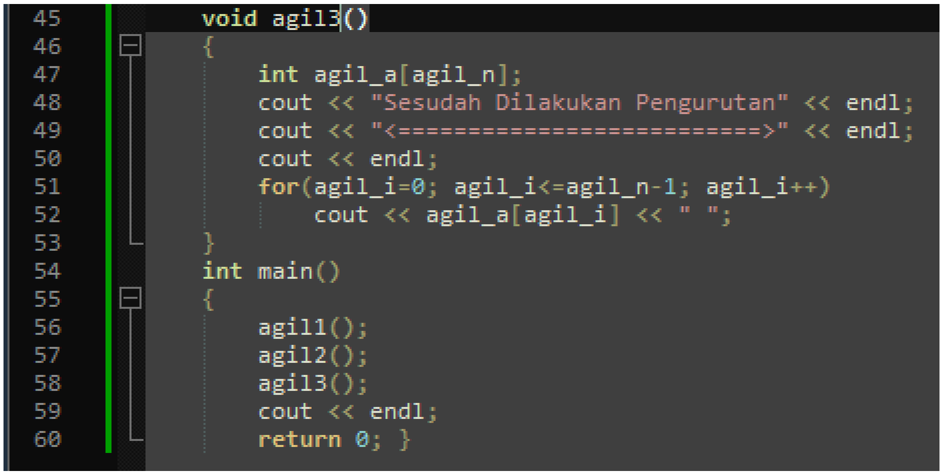
S1-Teknik Informatika
Fakultas Teknik
Universitas Pancasila 2023/2024

Sort



```
#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;
            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;
        cout << endl;
    }}}
    void agil3()
    {
```

```
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
<==========================>

23      17      14      13      10      7       6       5       1
```



```
    void agil3()
    {
        int agil_a[agil_n];
        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] << " ";
    }
    int main()
    {
        agil1();
        agil2();
        agil3();
        cout << endl;
        return 0; }
```

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]= dhandi_A[dhandi_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