

LAPORAN PRAKTIKUM STRUKTUR DATA
LAPORAN AWAL PRAKTIKUM

Pertemuan ke-08

Sorting Lanjut-2



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

Sorting adalah proses mengatur atau menyusun elemen-elemen dalam satu kumpulan data dengan aturan tertentu. Dalam program C++, terdapat beberapa algoritma sorting yang digunakan untuk mengurutkan array atau struktur data lainnya, seperti Bubble Sort, Selection Sort, dan Insertion Sort. Keuntungan menggunakan algoritma sorting yaitu menghemat waktu dan memudahkan dalam mencari nilai tertentu pada suatu array yang sudah terurut.

Shell Sort adalah algoritma sorting yang menggunakan konsep pembagian data menjadi beberapa bagian yang lebih kecil untuk dilakukan sorting secara terpisah. Algoritma ini menggunakan pendekatan incremental, di mana elemen-elemen yang berjarak jauh dibandingkan dan ditukar jika diperlukan. Proses ini terus berlanjut dengan mengurangi jarak antar elemen hingga mencapai jarak 1. Setelah itu, algoritma ini beralih ke penggunaan Insertion Sort untuk menyelesaikan proses sorting. Shell Sort memiliki kompleksitas waktu yang lebih baik daripada Insertion Sort, tetapi tidak secepat algoritma sorting yang lebih efisien seperti Quick Sort atau Merge Sort.

Insertion Sort adalah algoritma sorting sederhana yang bekerja dengan membagi data menjadi dua bagian: bagian yang sudah diurutkan dan bagian yang belum diurutkan. Algoritma ini memilih satu elemen dari bagian belum diurutkan dan memasukkannya ke posisi yang tepat di bagian yang sudah diurutkan. Proses ini diulang untuk setiap elemen dalam bagian belum diurutkan hingga seluruh data terurut. Insertion Sort memiliki kompleksitas waktu yang lebih baik daripada algoritma sorting seperti Bubble Sort atau Selection Sort untuk jumlah data yang kecil atau hampir terurut.

B. TUGAS PENDAHULUAN

1. Jelaskan kekurangan menggunakan metode Shell Sort dan Insertion Sort dengan metode-metode Sorting lainnya!

Jawab :

Kekurangan metode Shell Sort adalah membutuhkan method tambahan dan sulit untuk membagi masalah . Sedangkan kekurangan metode Insertion Sort adalah untuk larik yang jumlahnya besar tidak praktis.

2. Jelaskan perbedaan program Sorting dengan menggunakan antara metode Shell Sort dan Insertion Sort

Jawab :

Metode Shell Sort ini mengurutkan data dengan cara membandingkan suatu data dengan data lain yang memiliki jarak tertentu, kemudian dilakukan penukaran bila diperlukan. Perbedaannya dengan metode Insertion Sort ini memilih elemen dengan nilai paling rendah dan menukar dengan elemen terpilih dengan elemen ke-i. Nilai dari i dimulai dari 1 ke n, dimana n adalah jumlah total elemen yang dikurangi 1.

3. Jelaskan tahapan-tahapan Sorting menggunakan metode Shell Sort!

Jawab :

Jarak ditentukan dengan $n \div 2$, dimana n adalah banyaknya elemen array. Lakukan pertukaran tempat jika setiap kali perbandingan dipenuhi (lebih besar untuk menaik dan lebih kecil untuk urut menurun). Setiap kali perbandingan terhadap keseluruhan elemen selesai dilakukan, maka perbandingan yang baru dilakukan kembali dimana jarak diperoleh dengan jarak $\div 2$ (jarak diperoleh dari nilai jarak sebelumnya).

4. Jelaskan tahapan-tahapan Sorting menggunakan metode Insertion Sort!

Jawab :

Jarak ditentukan dengan $n \div 2$, dimana n adalah banyaknya elemen array. Lakukan pertukaran tempat jika setiap kali perbandingan dipenuhi (lebih besar untuk menaik dan lebih kecil untuk urut menurun). Setiap kali perbandingan terhadap keseluruhan elemen selesai dilakukan, maka perbandingan yang baru dilakukan kembali dimana jarak diperoleh dengan jarak $\div 2$ (jarak diperoleh dari nilai jarak sebelumnya).

C. TUGAS PRAKTIKUM

- Lat9_1

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[20];
    int i, k, N, l;
    int temp, jarak, s;

    cout << "Masukkan Banyak Bilangan : ";
    cin >> N;

    for (i = 0; i < N; i++)
    {
        cout << "Elemen Ke-" << i << " : ";
        cin >> Nilai[i];
    }

    cout << "\nData Sebelum diurutkan : ";
    for (i = 0; i < N; i++)
    {
        cout << Nilai[i] << " ";
    }

    cout << "\n\nMetode Shell Sort (Menaik) : ";

    jarak = N / 2;
    cout << "\nJarak = " << jarak;

    while (jarak >= 1) {
        do {
            s = 0;

            for (i = 0; i <= (N - jarak) - 1; i++) {
                k = i + jarak;

                if (Nilai[i] > Nilai[k]) {
                    temp = Nilai[i];
                    Nilai[i] = Nilai[k];
                    Nilai[k] = temp;
                    s = 1;
                }

                for (l = 0; l < N; l++) {
                    cout << Nilai[l] << " ";
                }
            }
        }
    }

    while (s != 0);
    jarak /= 2;
    cout << "\nJarak = " << jarak << "\n";
}
```

```

cout << "\n\nData Setelah diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}

return 0;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====
Masukkan Banyak Bilangan : 3
Elemen Ke-0 : 23
Elemen Ke-1 : 12
Elemen Ke-2 : 32

Data Sebelum diurutkan : 23 12 32

Metode Shell Sort (Menaik) :
Jarak = 112 23 32
Jarak = 0

Data Setelah diurutkan : 12 23 32
=====
Process exited after 7.287 seconds with return value 0
Press any key to continue . . .

```

- Lat9_2

```

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[20];
    int i, k, N, l;
    int temp, jarak, s;

    cout << "Masukkan Banyak Bilangan : ";
    cin >> N;

    for (i = 0; i < N; i++)
    {
        cout << "Elemen Ke-" << i << " : ";
        cin >> Nilai[i];
    }

    cout << "\nData Sebelum diurutkan : ";
    for (i = 0; i < N; i++)
    {
        cout << Nilai[i] << " ";
    }

    cout << "\n\nMetode Shell Sort (Menurun) : ";

    jarak = N / 2;
    cout << "\nJarak = " << jarak;

    while (jarak >= 1) {
        do {
            s = 0;

            for (i = 0; i <= (N - jarak) - 1; i++) {

```

```

        k = i + jarak;

        if (Nilai[i] < Nilai[k]) {
            temp = Nilai[i];
            Nilai[i] = Nilai[k];
            Nilai[k] = temp;
            s = 1;

            for (l = 0; l < N; l++) {
                cout << Nilai[l] << " ";
            }
        }
    }
}

while (s != 0);
jarak /= 2;
cout << "\nJarak = " << jarak << "\n";
}

cout << "\n\nData Setelah diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}

return 0;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan Banyak Bilangan : 3
Elemen Ke-0 : 43
Elemen Ke-1 : 23
Elemen Ke-2 : 45

Data Sebelum diurutkan : 43 23 45

Metode Shell Sort (Menaik) :
Jarak = 23 43 45
Jarak = 0

Data Setelah diurutkan : 23 43 45
=====
Process exited after 12.29 seconds with return value 0
Press any key to continue . . .

```

- Lat9_3

```

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[20];
    int i, j, k, N;
    int temp;

    cout << "Masukkan Banyak Bilangan : ";
    cin >> N;

    for (i = 0; i < N; i++)
    {
        cout << "Elemen Ke-" << i << " : ";
    }
}

```

```

    cin >> Nilai[i];
}

cout << "\nData Sebelum diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}

cout << "\n\nMetode Insertion Sort (Menaik) : ";

for (i = 0; i < N; i++) {
    temp = Nilai[i];
    j = i - 1;

    while ((temp <= Nilai[j]) && (j >= 1)) {
        Nilai[j + 1] = Nilai[j];
        j--;
    }

    if (temp >= Nilai[j]) {
        Nilai[j + 1] = temp;
    } else {
        Nilai[j + 1] = Nilai[j];
        Nilai[j] = temp;
    }
}

cout << "\n\nData Setelah diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}

return 0;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====
Masukkan Banyak Bilangan : 3
Elemen Ke-0 : 43
Elemen Ke-1 : 12
Elemen Ke-2 : 5

Data Sebelum diurutkan : 43 12 5
Metode Insertion Sort (Menaik) :
Data Setelah diurutkan : 5 12 43
=====
Process exited after 6.35 seconds with return value 0
Press any key to continue . . .

```

- Lat9_4

```

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

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[20];
    int i, j, k, N;
    int temp;

    cout << "Masukkan Banyak Bilangan : ";
}

```

```

cin >> N;

for (i = 0; i < N; i++)
{
    cout << "Elemen Ke-" << i << " : ";
    cin >> Nilai[i];
}

cout << "\nData Sebelum diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}

cout << "\n\nMetode Insertion Sort (Menurun) : ";

for (i = 0; i < N; i++) {
    temp = Nilai[i];
    j = i - 1;

    while ((temp > Nilai[j]) && (j >= 1)) {
        Nilai[j + 1] = Nilai[j];
        j--;
    }

    if (temp <= Nilai[j]) {
        Nilai[j + 1] = temp;
    } else {
        Nilai[j + 1] = Nilai[j];
        Nilai[j] = temp;
    }
}

cout << "\n\nData Setelah diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}

return 0;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 2110114013009
=====
Masukkan Banyak Bilangan : 3
Elemen Ke-0 : 0
Elemen Ke-1 : 21
Elemen Ke-2 : 10

Data Sebelum diurutkan : 0 21 10

Metode Insertion Sort (Menurun) :

Data Setelah diurutkan : 21 10 0
=====
Process exited after 8.179 seconds with return value 0
Press any key to continue . . .

```

- Lat9_5

```

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

void shellSortAsc()
{
    int Nilai[20];
    int i, k, N, l;
    int temp, jarak, s;

```



```

cout << "Masukkan Banyak Bilangan : ";
cin >> N;

for (i = 0; i < N; i++)
{
    cout << "Elemen Ke-" << i << " : ";
    cin >> Nilai[i];
}

cout << "\nData Sebelum diurutkan : ";
for (i = 0; i < N; i++)
{
    cout << Nilai[i] << " ";
}

cout << "\n\nMetode Shell Sort (Menaik) : ";

jarak = N / 2;
cout << "\nJarak = " << jarak;

while (jarak >= 1) {
    do {
        s = 0;

        for (i = 0; i <= (N - jarak) - 1; i++) {
            k = i + jarak;

            if (Nilai[i] > Nilai[k]) {
                temp = Nilai[i];
                Nilai[i] = Nilai[k];
                Nilai[k] = temp;
                s = 1;
            }

            for (l = 0; l < N; l++) {
                cout << Nilai[l] << " ";
            }
        }
    }
}

while (s != 0);
jarak /= 2;
cout << "\nJarak = " << jarak << "\n";
}

cout << "\n\nData Setelah diurutkan : ";
for (i = 0; i < N; i++) {
    cout << Nilai[i] << " ";
}
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int pilihan;

    cout << "Pilih Metode Pengurutan : \n";
    cout << "1. Metode Shell Sort (Menaik) :\n";
    cout << "2. Metode Shell Sort (Menurun) :\n";
    cout << "3. Metode Insertion Sort (Menaik) :\n";
}

```

```

cout << "4. Metode Insertion Sort (Menurun) :\n";

cout << "\nPilihan : ";
cin >> pilihan;

switch (pilihan)
{
    case 1:
        shellSortAsc();
        break;
    default:
        cout << "Pilihan tidak tersedia.";
        break;
}

return 0;
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====
Pilih Metode Pengurutan :
1. Metode Shell Sort (Menaik) :
2. Metode Shell Sort (Menurun) :
3. Metode Insertion Sort (Menaik) :
4. Metode Insertion Sort (Menurun) :

Pilihan : 1
Masukkan Banyak Bilangan :
3
Elemen Ke-0 : 10
Elemen Ke-1 : 6
Elemen Ke-2 : 2

Data Sebelum diurutkan : 10 6 2

Metode Shell Sort (Menaik) :
Jarak = 16 10 2 6 2 10 2 6 10
Jarak = 0

Data Setelah diurutkan : 2 6 10
=====
Process exited after 14.44 seconds with return value 0
Press any key to continue . . .

```

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-08

Sorting Lanjut-2



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

1. Buatlah program untuk mengurutkan sederetan data: suka, aku, sama, kamu, dulu, sampai, dari, sekarang. Dengan menggunakan salah satu metode Shell Sort dan Insertion Sort!

Jawab :

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

void shellSort(string arr[], int n) {
    for (int gap = n / 2; gap > 0; gap /= 2) {
        for (int i = gap; i < n; i++) {
            string temp = arr[i];
            int j;
            for (j = i; j >= gap && arr[j - gap] > temp; j -= gap) {
                arr[j] = arr[j - gap];
            }
            arr[j] = temp;
        }
    }
}

void printArray(string arr[], int n) {
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main() {
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    const int n = 8;
    string data[n] = {"suka", "aku", "sama", "kamu", "dulu", "sampai", "dari", "sekarang"};

    cout << "Data sebelum diurutkan: ";
    printArray(data, n);

    shellSort(data, n);
    cout << "Data setelah diurutkan dengan Shell Sort: ";
    printArray(data, n);

    return 0;
}
```

```
C:\xampp\htdocs\project\teknik-informatika-s1\se...
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Data sebelum diurutkan: suka aku sama kamu dulu sampai
dari sekarang
Data setelah diurutkan dengan Shell Sort: aku dari dulu
kamu sama sampai sekarang suka

-----
Process exited after 3.109 seconds with return value 0
Press any key to continue . . .
```

2. KESIMPULAN

Secara singkatnya sorting adalah metode untuk pengurutan data. Secara garis besarnya, Sorting (Pengurutan) adalah suatu proses penyusunan kembali kumpulan objek menggunakan tata aturan tertentu. Sorting disebut juga sebagai suatu algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu berdasarkan satu atau beberapa kunci dalam tiap-tiap elemen. Pengurutan atau sorting merupakan proses dasar yang ada dalam sebuah algoritma dan struktur data. Penggunaan algoritma sorting dapat pula diaplikasikan pada algoritma Python. Tujuan utama dari proses pengurutan atau sorting adalah untuk mengurutkan data berdasarkan keinginan baik itu dari yang terendah maupun yang tertinggi, sehingga data yang dihasilkan akan lebih terstruktur, teratur dan sesuai dengan kebutuhan.