

PERTEMUAN 7

DEFINISI SORTING PADA C++

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan pertemuan ini, mahasiswa mampu mempraktekkan:

1. Konsep Sorting Pada C++
2. Bubble Sort
3. Quick Sort

B. URAIAN MATERI

1. Konsep Sorting Pada C++

Sorting adalah proses teknis membantu sekumpulan atau sekelompok objek dalam susunan. Barisan benda tersebut bisa menaik (ascending) yaitu barisan suatu benda mulai dari Nilai sampai terbesar, turun (descending) yaitu barisan suatu benda dari nilai terbesar ke urutannya. Jika N objek atau data disimpan dalam larik Nilai, sortir menaik mengatur elmen sedemikian rupa :

$$\text{NILAI}[0] \leq \text{NILAI}[1] \leq \text{NILAI}[2] \leq \dots \leq \text{NILAI}[N-1]$$

Sedangkan pengurutan dalam urutan menurun berarti elmen-elmen array diatur sedemikian rupa sehingga:

$$\text{NILAI}[0] \geq \text{NILAI}[1] \geq \dots \geq \text{NILAI}[N-1]$$

Data yang diurutkan berupa data dengan tipe data form. Jika data memiliki tipe struktur, maka harus dicantumkan berdasarkan bidang data yang akan disortir.

Karena akses ke memori utama lebih cepat daripada akses ke memori sekunder, jenis internal lebih cepat daripada jenis eksternal.

Banyak metode pengurutan menimbulkan pertanyaan: algoritma mana yang berkinerja terbaik? Performa pengurutan sangat menentukan performa sistem, karena pemilihan metod pemesanan yang sesuai akan berperan dalam aplikasi. Metode sortir yang akan dibahas adalah Bubble Sort, Quick Sort, Max / Minimum Sort, Merge Sort, dan Insert Sort.

2. Bubble Sort

Bubble Sort adalah metode yang membandingkan elmen saat ini dengan elmen berikut. Jika elmen saat ini lebih bsar (untuk urutan naik) atau lebih keccil (untuk urutan naik) dari elmen brikutnya, posiisinya berubah, tetapi jika tidak, posiisinya tetap.

Contoh: kita memiliki matriks nilai 8 elmen yang diurutkan scara bertahap menggunakan metode Bubble Sort: 25, 71, 30, 45, 20, 15, 6, 50. Pengurutan dimulai dari belakang sebagai berikut.

Tabel 7.2 Urutan Menaik Bubble Sort

Langkah -1:

25	72	30	45	20	15	6	50
25	72	30	45	20	6	15	50
25	72	30	45	6	20	15	50
25	72	30	6	45	20	15	50
25	72	6	30	45	20	15	50
25	6	72	30	45	20	15	50
6	25	72	30	45	20	15	50

Langkah -2:

6	25	72	30	45	20	15	50
6	25	72	30	45	15	20	50
6	25	72	30	15	45	20	50
6	25	72	15	30	45	20	50
6	25	15	72	30	45	20	50
6	15	25	72	30	45	20	50

Langkah -3:

6	15	25	72	30	45	20	50
6	15	25	72	30	20	45	50
6	15	25	72	20	30	45	50
6	15	25	20	72	30	45	50
6	15	20	25	72	30	45	50

Langkah -4:

6	15	20	25	72	30	45	50
6	15	20	25	72	30	45	50
6	15	20	25	30	72	45	50

Langkah -5:

6	15	20	25	30	72	45	50
6	15	20	25	30	45	72	50

Langkah -6:

6	15	20	25	30	45	59	72
---	----	----	----	----	----	----	----

Contoh: Kami memiliki array 8 elemen yang diurutkan dalam urutan menurun menggunakan metode Bubble Sort: 25, 72, 30, 45, 20, 15, 6, 50. Pengurutan dimulai dari belakang seperti berikut.

Tabel 7.3 Urutan Menurun Bubble Sort

Langkah -1:

25	72	30	45	20	15	50	6
25	72	30	45	20	50	15	6
25	72	30	45	50	20	15	6
25	72	30	50	45	20	15	6
25	72	50	30	45	20	15	6
25	72	50	30	45	20	15	6
72	25	50	30	45	20	15	6

Langkah -2:

72	25	50	30	45	20	15	6
72	25	50	30	45	20	15	6
72	25	50	30	45	20	15	6
72	25	50	45	30	20	15	6
72	25	50	45	30	20	15	6
72	50	25	45	30	20	15	6

Langkah -3:

72	50	25	45	30	20	15	6
72	50	25	45	30	20	15	6
72	50	25	45	30	20	15	6
72	50	25	45	30	20	15	6
72	50	45	25	30	20	15	6

Langkah -4:

72	50	45	25	30	20	15	6
72	50	45	25	30	20	15	6
72	50	45	25	30	20	15	6
72	50	45	30	25	20	15	6

Langkah -5:

72	50	45	30	25	20	15	6
72	50	45	30	25	20	15	6
72	50	45	30	25	20	15	6

Langkah -6:

72	50	45	30	25	20	15	6
72	50	45	30	25	20	15	6

Langkah -7:

72	50	45	30	25	20	15	6
----	----	----	----	----	----	----	---

Perhatikan contoh pengurutan yang ditunjukkan di atas yang telah diurutkan pada langkah 4 sehingga sebenarnya prosesnya dapat dipilah. Untuk mengatasi masalah ini, variabel tipe Boolean digunakan. Demikian juga, metode sortir

menaik masih menggunakan variabel tipe Boolean untuk menyelesaikan hal yang sama.

3. Quick Sort

Quick Sort mengurutkan data menggunakan prinsip rekursif. Cara ini menggunakan strategi "split" dengan mekanisme sebagai berikut.

Jika kita memiliki array Value $[k..l]$. Larik dibagi menjadi dua bagian yaitu Nilai larik kiri $[k..m]$ dan Nilai larik kanan $[m + 1..l]$. Partisi larik menjadi dua menggunakan elemen pertama sebagai elemen pivot. Elemen di sebelah kiri elemen pivot adalah elemen larik. Nilai elemen larik $[k..m]$. Nilai $[m + 2..l]$ adalah setiap elemen yang lebih besar dari pivot. Lakukan hal yang sama seperti di atas dengan nilai array $[k..m]$ dan Value $[m + 1..l]$ hingga tidak dapat lagi dipartisi.

Contoh: Kami memiliki matriks nilai dari 8 elemen yang diurutkan dalam urutan menaik menggunakan metode Sortir Maksimum: 25, 72, 30, 45, 20, 15, 6, 50. Urutannya adalah sebagai berikut .

- a. Terima sebuah elemn pertama sebagai elemn pivot, dan letak kan elemn aray yang lebih kcil dari pivot di sbelah kiri elemn pivot, dan tmpatkan semeua elemn aray yang lebih besar dari elemn pivot di sebelah kanan elemn pivot.

20	15	6		25		25	72	30	45	50
Nilai[0..2]				Nilai[3]		Nilai[4..7]				

- b. Manfaat kan matriks nlai [0.2]. Ambil elemn pertma sebagai titik pivot elemen, letakkan semua elemn peletakan yang kecil dari titik pivot elemn di sebelah kiri titik pivot elemen, dan tempatkan semua elmen peletakan dari larik yang lebih besar dari titik pvot di sebelah knan titik pvot elemen.

15	6		20	25	25	72	30	45	50
Nilai[0..1]			Nilai[2]	Nilai[3]	Nilai[4..7]				

Prhatikan bahwa nilai array [2] tidak lagi dapat dipartsi, sehingga brhenti di situ.

- c. Manfaat kan matriks nilai [0.1]. Ambil elemn petama sbagai elemn pvot, ltakan semua elemn aray yang lebi kecil di sbelah kiri elemen pivot, dan

tempatkan semua elemen array yang lebih besar dari elemen pivot di sebelah kanan elemen pivot.

6	15	20	25	25	72	30	45	50
Nilai[0]	Nilai[1]	Nilai[2]	Nilai[3]	Nilai[4..7]				

Perhatikan bahwa nilai array [0] dan kemudian Nilai [1] tidak lagi dapat dipartisi, sehingga berhenti di situ.

- d. Manfaatkan larik dengan Nilai [4.7]. Ambil elemen pertama sebagai elemen pivot, letakkan semua elemen array yang lebih kecil di sebelah kiri elemen pivot, dan tempatkan semua elemen array yang lebih besar dari elemen pivot di sebelah kanan elemen pivot.

6	15	20	25	30	45	50	72
Nilai[0]	Nilai[1]	Nilai[2]	Nilai[3]	Nilai[4..6]			Nilai[7]

- e. Manfaatkan larik dengan Nilai [4.6]. Ambil elemen pertama sebagai elemen pivot, letakkan semua elemen array yang lebih kecil di sebelah kiri elemen pivot, dan tempatkan semua elemen array yang lebih besar dari elemen pivot di sebelah kanan elemen pivot.

6	15	20	25	30	45	50	72
Nilai[0]	Nilai[1]	Nilai[2]	Nilai[3]	Nilai[4]	Nilai[5..6]		Nilai[7]

- f. Gunakan matriks nilai [5.6]. Ambil elemen pertama sebagai elemen pivot, letakkan semua elemen array yang engselnya lebih kecil di sebelah kiri elemen pivot, dan tempatkan semua elemen dari array yang lebih besar dari elemen pivot di sebelah kanan elemen pivot.

6	15	20	25	30	45	50	72
Nilai[0]	Nilai[1]	Nilai[2]	Nilai[3]	Nilai[4]	Nilai[5]	Nilai[6]	Nilai[7]

Karna tidak semua elemen array dapat dipartisi, proses pengurutan berakhir dan hasilnya diperoleh sebagai berikut (gabungkan dari Value [0] menjadi Value [7]):

6	15	20	25	30	45	50	
---	----	----	----	----	----	----	--

Untuk melakukan proses descending pengurutan data menggunakan metod Quick Sort dilakukan dengan menempatkan semua elemen larik yang lebih

kecil dari titik pivot di sebelah kanan pivot point dan semua elemen larik lebih besar dari titik pivot di sebelah kiri pivot point. titik sumbu.

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan 7	<ol style="list-style-type: none"> 1. Apa yang di maksud dengan Sorting ? 2. Algoritma mana yang memiliki kinerja yang baik ? 3. Terdapat urutan data berikut : 12 7 9 10 13 15 16 1 tulislah urutan proses untuk mengurutkan data Dengan menggunakan “Buble Sort” ? 4. Buatlah program sorting berdasarkan inputan user dengan menghasilkan output data dengan bilangan genap saja menggunakan bubble sort 5. Terdapat urutan data berikut : 12 7 9 10 13 15 16 1 tulislah urutan proses untuk mengurutkan data Dengan menggunakan “Quick Sort”?

D. REFERENS

C and Data Structures by Practice by Ramesh Vasappanavara

Data Structures Program Design in C++ by KruseDordal, P. L. (2020). *An Introduction to Computer Network*. Chicago: Loyola University Chicago.

Forouzan, B. A. (2013). *Data Communications and Networking*. New York: McGraw-Hill.

Goralski, W. (2017). *The Illustrated Network*. Cambridge: Morgan Kaufmann.

Kurose, J. F., & Ross, K. W. (2017). *Computer Networking: A Top-down Approach*. Pearson.

Lowe, D. (2018). *Networking All-In-One*. Hoboken: John Wiley & Sons, Inc.

- Peterson, L. L., & Davie, B. S. (2010). *Computer Networks*. Burlington: Kaufmann.
- Sudiendro, H. (2013). *Teknik Dasar Telekomunikasi*. Jakarta: Kementrian Pendidikan & Kebudayaan.
- Sukaridhoto, S. (2014). *Buku Jaringan Komputer I*. Surabaya: Politeknik Elektronika Negeri Surabaya (PENS).
- Tanenbaum, A. S., & Wetherall, D. J. (2011). *Computer Networks*. Pearson Prentice Hall.