

BAB 11

PENGELOLAAN DATA PADA SINGLE & DOUBLE LINKEDLIST : PENGURUTAN (SORTING) DAN PENCARIAN (SEARCHING)

Tujuan

Mahasiswa dapat melakukan pengurutan terhadap suatu data yang terdapat di dalam linkedlist.

11.1. Pengurutan Data (Sorting) pada Linkedlist

Pada bab 5 kita yang lalu telah mempelajari bagaimana cara mengurutkan data yang tersimpan dalam sebuah sederet Larik/ array. Pada bab 11 ini kita akan mempelajari bagaimana melakukan cara mengurutkan data pada sebuah linkedlist.

Dalam prosesnya, pengurutan data dalam sebuah larik/array sangat mudah dilakukan. Hal ini karena pada struktur penyimpan larik terdapat indeks yang dapat membantu menandai data-data yang hendak dibandingkan/ ditukar.

Tidak seperti pada larik, pengurutan data dalam sebuah linkedlist lebih sulit dilakukan. Hal ini karena pada saat kita hendak membandingkan/ menukar data, tidak ada indeks yang dapat membantu untuk menandai data-data tersebut. Untuk itu diperlukan cara lain untuk menggantikan peran indeks tersebut.

Pada bab ini kita akan mempelajari bagaimana membuat program untuk mengurutkan data pada struktur penyimpan linkedlist. Metode pengurutan data yang akan kita gunakan adalah bubblesort, selection sort, dan insertion sort.

Sebelum kita mempelajari bagaimana program pengurutan pada linkedlist, perlu kita pahami terlebih dahulu bahwa di dalam banyak metode pengurutan data terdapat dua proses penting yang selalu ada, yaitu :

1. **membandingkan** dua buah data,
2. **menukar kedua data** tersebut jika diperlukan.

Pada proses membandingkan dua data, baik untuk struktur penyimpan array/larik maupun linkedlist, cara yang diterapkan cenderung sama, yaitu sama-sama harus ditemukan dahulu data yang dimaksud. (pada metode yang berbeda akan berbeda pula cara menemukannya).

Dalam proses menukar data, pada array/larik penukaran data hanya dapat dilakukan dengan menukar isi atau nilai (value) dari variabel yang menyimpan data-data

tersebut. Sementara pada linkedlist penukaran data dapat dilakukan dengan 2 pendekatan :

1. Menukar isi variabelnya (yaitu dengan cara menukar isi heap namun posisi heap tidak berubah)
2. Menukar posisi heap (tukar heap, yaitu dengan merubah posisi heap/ simpul-simpul dalam linkedlist)

Pada bab ini akan kita pelajari bagaimana melakukan penukaran data dalam linkedlist dengan kedua teknik di atas.

Metode Sorting	Single LinkedList		Double LinkedList	
	Tukar Nilai	Tukar Heap	Tukar Nilai	Tukar Heap
Bubble Sort	<i>ada</i>	<i>ada</i>	<i>ada</i>	<i>ada</i>
Selection Sort	<i>ada</i>	-	<i>ada</i>	<i>ada</i>
Insertion Sort	-	<i>ada</i>	-	<i>ada</i>

11.1.1. Pengurutan Bubble pada Single/Double LinkedList

Algoritma bubble sort akan membandingkan elemen yang saat ini dibaca dengan elemen yang berikutnya. Jika elemen yang saat ini dibaca lebih besar dari elemen berikutnya, maka tukarkan.

Berikut ini akan dibahas kedua pendekatan yang dimaksud untuk metode Bubble Sort.

11.1.1.1. Pengurutan Bubble pada Single/Double LinkedList dengan Pendekatan Tukar Nilai

Pendekatan pertama dimana proses penukaran data hanya dilakukan dengan cara menukar isi variabelnya saja. Program 11.1 berikut adalah fungsi pengurutan bubblesort untuk single dan double linkedlist dengan cara menukar isi variabel.

```
public static void tukarNilai( simpul X, simpul Y)
{
    simpul sementara = new simpul();

    sementara.nama    = X.nama;
    sementara.alamat  = X.alamat;
    sementara.umur    = X.umur;
    sementara.jekel   = X.jekel;
    sementara.ipk     = X.ipk;

    X.nama    = Y.nama;
    X.alamat  = Y.alamat;
    X.umur    = Y.umur;
    X.jekel   = Y.jekel;
    X.ipk     = Y.ipk;
}
```

```

        Y.nama    = sementara.nama;
        Y.alamat  = sementara.alamat;
        Y.umur    = sementara.umur;
        Y.jekel   = sementara.jekel;
        Y.ipk     = sementara.ipk;
    }

    //-----
    // bisa untuk single LL dan double LL
    //-----
    public static void mengurutkanDataBubble_TeknikTukarNilai()
    {
        int N = hitungJumlahSimpul();
        simpul A=null;
        simpul B=null;
        simpul berhenti = akhir.kanan;

        System.out.println ("Banyaknya simpul = " + hitungJumlahSimpul());
        for (int i=1; i<= hitungJumlahSimpul()-1; i++)
        {
            A = awal;
            B = awal.kanan;
            int nomor = 1;
            while (B != berhenti)
            {
                if (A.nama.compareTo(B.nama)>0)
                {
                    //tukarkan elemen dari simpul A dan elemen dari simpul B
                    tukarNilai(A,B);
                }
                A = A.kanan;
                B = B.kanan;
                nomor++;
            }
            berhenti = A;
        }
        System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI=====");
    }
}

```

Program 11.1. Fungsi Bubblesort untuk single linkedlist dan double linkedlist dengan penukaran isi variabel

11.1.1.2. Pengurutan Bubble pada Single Linkedlist dengan Pendekatan Tukar Heap

Pendekatan kedua dimana proses penukaran data dilakukan dengan cara menukar posisi heapnya. Program 11.2 berikut adalah fungsi pengurutan bubblesort untuk single linkedlist dengan cara menukar posisi heapnya.

```

public static void mengurutkanDataBubble_TeknikTukarHeap()
{
    int N = hitungJumlahSimpul();
    simpul A=null;
    simpul B=null;
    simpul bantu=null;
    simpul berhenti = akhir.kanan;
    int nomor;
}

```

```

System.out.println ("Banyaknya simpul = " + hitungJumlahSimpul());
for (int i=1; i<= hitungJumlahSimpul()-1; i++)
//for (int i=1; i<= 4; i++)
{
    A = awal;
    B = awal.kanan;
    nomor = 1;

    //proses banding-tukar, khusus simpul pertama dgn sebelahnya
    if (A.nama.compareTo(B.nama)> 0)
    {
        A.kanan = B.kanan;
        B.kanan = A;
        awal = B;
    }

    //proses banding-tukar, simpul kedua dgn sebelahnya, dst
    nomor++;
    bantu = awal;
    while (bantu.kanan.kanan!=berhenti)
    {
        A = bantu.kanan;
        B = bantu.kanan.kanan;
        if (A.nama.compareTo(B.nama)>0)
        {
            //tukarkan simpul A dan simpul B
            A.kanan = B.kanan;
            B.kanan = A;
            bantu.kanan = B;
            if (B==akhir) akhir = A;
        }
        bantu = bantu.kanan;
        nomor++;
    }
    berhenti = bantu.kanan;;
    System.out.println ("");
}
System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI=====");
}

```

Program 11.2. Fungsi Bubblesort untuk **single linkedlist** dengan penukaran posisi heap

Pendekatan kedua dimana proses penukaran data dilakukan dengan cara menukar posisi heapnya.

11.1.1.3. Pengurutan Bubble pada Double Linkedlist dengan Pendekatan Tukar Heap

Program 11.3 berikut adalah fungsi pengurutan bubblesort untuk **double linkedlist** dengan cara menukar posisi heapnya.

```

public static void mengurutkanDataBubble_TeknikTukarHeap()
{
    int N = hitungJumlahSimpul();
    simpul bantu = awal;

    System.out.println ("Banyaknya simpul = " + hitungJumlahSimpul());
}

```

```

for (int i=1; i<= hitungJumlahSimpul(); i++)
{
    //khusus menguji simpul pertama dgn sebelahny
    if (awal.nama.compareTo(awal.kanan.nama)> 0)
    {
        bantu = awal.kanan;
        awal.kanan = bantu.kanan;
        bantu.kanan.kiri = awal;
        bantu.kanan = awal;
        bantu.kiri = null;
        awal.kiri = bantu;
        awal = bantu;
    }

    //khusus menguji simpul kedua dgn sebelahny, simpul ketiga dgn
    sebelahny,dst,
    bantu = awal;
    while (bantu.kanan != akhir)
    {
        simpul A = bantu.kanan;
        simpul B = bantu.kanan.kanan;
        if (A.nama.compareTo(B.nama)>0)
        {
            //tukarkan simpul A dan simpul B
            A.kanan = B.kanan;
            if (B!=akhir) A.kanan.kiri = A;

            B.kanan.kiri = A;
            B.kanan = A;
            A.kiri = B;

            bantu.kanan = B;
            B.kiri = bantu;

            if (B==akhir) akhir = A;
        }
        bantu = bantu.kanan;
    }
    System.out.println ("");
}
System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI====");
}

```

Program 11.3. Fungsi Bubblesort untuk **double linkedlist** dengan penukaran posisi heap

11.1.2. Pengurutan Selection pada Single/ Double Linkedlist

Pengurutan data pada linkedlist dengan menggunakan metode Selection Sort akan dibahas pada buku yang lain

11.1.3. Pengurutan Insertion pada Single/ Double Linkedlist

Pengurutan data pada linkedlist dengan menggunakan metode Insertion Sort akan dibahas pada buku yang lain

11.2. Pencarian Data (Searching) pada Linkedlist

Pada bab 4 kita telah mempelajari topik tentang pencarian data dalam sebuah sederet Larik/ array. Pada bab 11 ini kita juga akan mempelajari bagaimana melakukan pencarian data yang tersimpan dalam sebuah linkedlist.

Pada searching linkedlist, pencarian yang dapat dilakukan adalah jenis pencarian Linear Search atau Sequential Search. Hal ini karena pencarian linear search dilakukan dengan menyisir data dari posisi data pertama hingga data terakhir. Sementara pencarian biner (binary search) cenderung sulit dilakukan karena metode bagi dua yang menjadi ciri khas metoden ini sulit diimplementasikan pada linkedlist.

Pada bagian ini akan kita pelajari bagaimana melakukan pencarian data dalam linkedlist metode linear search. Program 11.4 menjelaskan langkah-langkah yang dilakukan untuk pencarian linear menggunakan single linkedlist.

```
public static void cariLinear()
{
    if (awal==null) // jika senarai masih kosong
        System.out.print("....MAAF SENARAI KOSONG....");
    else // jika senarai tidak kosong
    {
        Scanner masukan = new Scanner(System.in);
        System.out.print("Silakan masukkan nama yang anda cari : ");
        String NAMACARI = masukan.nextLine();

        boolean statusKetemu = false;
        int i = 0;
        int posisiKetemu=-1;

        simpul bantu;
        bantu = awal;
        while (bantu != null)
        {
            if (NAMACARI.equals(bantu.nama))
            {
                statusKetemu = true;
                posisiKetemu = i;
            }
            bantu = bantu.kanan;
            i++;
        }
        System.out.println("Status Ketemu = "+statusKetemu +" di posisi
            ke "+posisiKetemu);
    }
}
```

Program 11.4. Fungsi Linear Search pada **single linkedlist**

11.3. Latihan

Latihan 1 :

Tambahkan program 11.1 hingga 11.4 ke dalam master program yang telah anda buat pada bab yang lalu. Perhatikan mana program yang digunakan untuk single linklist dan mana yang untuk double linklist (jangan tertukar).

Eksekusi master program anda untuk melakukan pengurutan data secara **Bubblesort**. Ujilah program dengan mengentri data secara acak (nama mahasiswa dientri tidak dalam keadaan urut) kemudian lakukan pengurutan terhadap data tersebut, kemudian tampilkan data kembali. Bagaimana hasilnya? Catat dan simpulkan dalam laporan anda.

Eksekusi master program anda untuk melakukan pencarian data secara linear search. Ujilah juga dengan mengentri data secara acak kemudian lakukan pencarian sembarang terhadap data tersebut, kemudian tampilkan hasilnya.

Catat dan simpulkan percobaan anda hari ini dalam laporan anda.

Latihan

Tugas

