

LAPORAN PRAKTIKUM STRUKTUR DATA

PERTEMUAN KE-11



Disusun Oleh :

NAMA : Raden Isnawan Argi Aryasatya

NIM : 195410257

JURUSAN : Teknik Informatika

JENJANG : S1

Laboratorium Terpadu

Sekolah Tinggi Management Informatika Komputer

AKAKOM

YOGYAKARTA

2020

PERTEMUAN KE-11

(MENGURUTKAN DATA (SORTING) DAN PENCARIAN DATA (SEARCHING) PADA LINKEDLIST)

TUJUAN

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

DASAR TEORI

Pada modul 5 kita yang lalu telah mempelajari bagaimana cara mengurutkan data yang tersimpan dalam sebuah sederet Larik/ array. Pada modul 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 modul 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.

PRAKTIK 1

SORTING: bubbleshort

11.1: Bubblesort untuk single linkedlist dan double linkedlist dengan penukaran isi variabel

```
senaraiGanda.java x
277 public static void tukarNilai( simpul X, simpul Y)
278 {
279     simpul sementara = new simpul();
280     sementara.nama = X.nama;
281     sementara.alamat = X.alamat;
282     sementara.umur = X.umur;
283     sementara.jekel = X.jekel;
284     sementara.ipk = X.ipk;
285     X.nama = Y.nama;
286     X.alamat = Y.alamat;
287     X.umur = Y.umur;
288     X.jekel = Y.jekel;
289     X.ipk = Y.ipk;
290     Y.nama = sementara.nama;
291     Y.alamat = sementara.alamat;
292     Y.umur = sementara.umur;
293     Y.jekel = sementara.jekel;
294     Y.ipk = sementara.ipk;
295 }
296 //-----
297 // bisa untuk single LL dan double LL
298 //-----
299 public static void mengurutkanDataBubble_TeknikTukarNilai()
300 {
301     int N = hitungJumlahSimpul();
302     simpul A=null;
303     simpul B=null;
304     simpul berhenti = akhir.kanan;
305     System.out.println ("Banyaknya simpul = " + hitungJumlahSimpul());
306     for (int i=1; i<= hitungJumlahSimpul()-1; i++)
307     {
308         A = awal;
309         B = awal.kanan;
310         int nomor = 1;
311         while (B != berhenti)
312         {
```

```

313         if (A.nama.compareTo(B.nama) > 0)
314         {
315             //tukarkan elemen dari simpul A dan elemen dari simpul B
316             tukarNilai(A,B);
317         }
318         A = A.kanan;
319         B = B.kanan;
320         nomor++;
321     }
322     berhenti = A;
323 }
324 System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI=====");
325 }

```

```

C:\Windows\system32\cmd.exe
Silakan masukkan nama anda : Jake
Silakan masukkan alamat anda : Sleman
Silakan masukkan umur anda : 23
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : makan
hobi ke-1 : minum
hobi ke-2 : coding
Silakan masukkan IPK anda : 3,7
TAMBAH DEPAN :
Silakan masukkan nama anda : Beth
Silakan masukkan alamat anda : Texas
Silakan masukkan umur anda : 21
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : dansa
hobi ke-1 : nyanyi
hobi ke-2 : nari
Silakan masukkan IPK anda : 3,4
TAMBAH BELAKANG :
Silakan masukkan nama anda : McBoo
Silakan masukkan alamat anda : Ngawi
Silakan masukkan umur anda : 24
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : editing
hobi ke-1 : video
hobi ke-2 : foto
Silakan masukkan IPK anda : 3,2
TAMBAH BELAKANG :
Silakan masukkan nama anda : Pearl
Silakan masukkan alamat anda : Wates
Silakan masukkan umur anda : 22
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : dandan
hobi ke-1 : jogging
hobi ke-2 : lukis
Silakan masukkan IPK anda : 3,1
Silakan masukkan nama yang ingin dihapus : McBoo
menghapus McBoo dilakukan..
Banyaknya simpul = 3
===PROSES PENGURUTAN BUBBLE SELESAI=====
-----
NO NAMA ALAMAT UMUR JEKEL IPK
-----
Beth      Texas    21      P      dansa  nyanyi  nari    3.4
Jake      Sleman   23      L      makan  minum  coding  3.7
Pearl     Wates    22      P      dandan  jogging  lukis   3.1
-----
NO NAMA ALAMAT UMUR JEKEL IPK
-----
Pearl     Wates    22      P      dandan  jogging  lukis   3.1
Jake      Sleman   23      L      makan  minum  coding  3.7
Beth      Texas    21      P      dansa  nyanyi  nari    3.4
-----

```

penjelasan:

tujuan BubbleSort disini adalah untuk mengurutkan setiap data yang dimasukkan. Di program di atas, saya memasukkan 4 data lalu menghapus satu data bernama “McBoo”. Setelah itu program Bubble Sort menghitung data yang tersisa yaitu 3 data

11.2: Fungsi Bubblesort untuk single linkedlist dengan penukaran posisi heap

```
singleLinkedList.java x
296 public static void mengurutkanDataBubble_TeknikTukarHeap()
297 {
298     int N = hitungJumlahSimpul();
299     simpul A=null;
300     simpul B=null;
301     simpul bantu=null;
302     simpul berhenti = akhir.kanan;
303     int nomor;
304     System.out.println ("Banyaknya simpul = " + hitungJumlahSimpul());
305     for (int i=1; i<= hitungJumlahSimpul()-1; i++)
306         //for (int i=1; i<= 4; i++)
307         {
308             A = awal;
309             B = awal.kanan;
310             nomor = 1;
311             //proses banding-tukar, khusus simpul pertama dgn sebelahnya
312             if (A.nama.compareTo(B.nama)> 0)
313             {
314                 A.kanan = B.kanan;
315                 B.kanan = A;
316                 awal = B;
317             }
318             //proses banding-tukar, simpul kedua dgn sebelahnya, dst
319             nomor++;
320             bantu = awal;
321             while (bantu.kanan.kanan!=berhenti)
322             {
323                 A = bantu.kanan;
324                 B = bantu.kanan.kanan;
325                 if (A.nama.compareTo(B.nama)>0)
326                 {
327                     //tukarkan simpul A dan simpul B
328                     A.kanan = B.kanan;
329                     B.kanan = A;
330                     bantu.kanan = B;
331                     if (B==akhir) akhir = A;
332                 }
333                 bantu = bantu.kanan;
334                 nomor++;
335             }
336             berhenti = bantu.kanan;;
337             System.out.println ("");
338         }
339         System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI=====");
340     }
341     //-----bagian program utama-----
342     public static void main(String[] args)
343     {
344         inisialisasiSenaraiKosong();
345         tambahDepan();
346         tambahDepan();
347         tambahBelakang();
348         tambahBelakang();
349         hapus();
350         mengurutkanDataBubble_TeknikTukarHeap();
351         cetakSenarai();
352     }
353 }
```

```

C:\Windows\system32\cmd.exe
TAMBAH DEPAN :
Silakan masukkan nama anda : Blake
Silakan masukkan alamat anda : Munich
Silakan masukkan umur anda : 23
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : renang
hobi ke-1 : voli
hobi ke-2 : bola
Silakan masukkan IPK anda : 3,4
TAMBAH DEPAN :
Silakan masukkan nama anda : Emma
Silakan masukkan alamat anda : London
Silakan masukkan umur anda : 22
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : akting
hobi ke-1 : nyanyi
hobi ke-2 : model
Silakan masukkan IPK anda : 3,2
TAMBAH BELAKANG :
Silakan masukkan nama anda : Rhino
Silakan masukkan alamat anda : Uganda
Silakan masukkan umur anda : 30
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : foto
hobi ke-1 : video
hobi ke-2 : musik
Silakan masukkan IPK anda : 3,1
TAMBAH BELAKANG :
Silakan masukkan nama anda : Sheila
Silakan masukkan alamat anda : Jogja
Silakan masukkan umur anda : 20
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : lukis
hobi ke-1 : gambar
hobi ke-2 : desain
Silakan masukkan IPK anda : 3,2
Silakan masukkan nama yang ingin dihapus : Emma
menghapus Emma dilakukan..
Banyaknya simpul = 3

===PROSES PENGURUTAN BUBBLE SELESAI=====
NO NAMA ALAMAT UMUR JEREL IPK
-----
Blake Munich 23 L renang voli bola 3.4
Rhino Uganda 30 L foto video musik 3.1
Sheila Jogja 20 P lukis gambar desain 3.2
Press any key to continue . . . _

```

penjelasan:

tujuan BubbleSort disini adalah untuk mengurutkan setiap data yang dimasukkan. Di program di atas, saya memasukkan 4 data lalu menghapus satu data bernama “Emma”. Setelah itu program Bubble Sort menghitung data yang tersisa yaitu 3 data. Heap digunakan untuk menyimpan data tambahan sesuai permintaan (request data). Karena memori yang dipakai oleh stack bersifat sementara, maka informasi yang terdapat di alamat stack akan langsung didealokasi secara otomatis ketika scope sebuah program berakhir.

11.3: Fungsi Bubblesort untuk double linkedlist dengan penukaran posisi heap

```

senaraiGanda.java X
277 public static void mengurutkanDataBubble_TeknikTukarHeap()
278 {
279     int N = hitungJumlahSimpul();
280     simpul bantu = awal;
281     System.out.println ("Banyaknya simpul = " + hitungJumlahSimpul());
282     for (int i=1; i<= hitungJumlahSimpul(); i++)
283     {
284         //khusus menguji simpul pertama dgn sebelahny
285         if (awal.nama.compareTo(awal.kanan.nama)> 0)
286         {
287             bantu = awal.kanan;
288             awal.kanan = bantu.kanan;
289             bantu.kanan.kiri = awal;
290             bantu.kanan = awal;
291             bantu.kiri = null;
292             awal.kiri = bantu;
293             awal = bantu;
294         }
295         //khusus menguji simpul kedua dgn sebelahny, simpul ketiga
296         bantu = awal;
297         while (bantu.kanan != akhir)
298         {

```

```

senaraiGanda.java x
299     simpul A = bantu.kanan;
300     simpul B = bantu.kanan.kanan;
301     if (A.nama.compareTo(B.nama)>0)
302     {
303         //tukarkan simpul A dan simpul B
304         A.kanan = B.kanan;
305         if (B!=akhir) A.kanan.kiri = A;
306         B.kanan.kiri = A;
307         B.kanan = A;
308         A.kiri = B;
309         bantu.kanan = B;
310         B.kiri = bantu;
311         if (B==akhir) akhir = A;
312     }
313     bantu = bantu.kanan;
314 }
315     System.out.println("");
316 }
317 System.out.println("===PROSES PENGURUTAN BUBBLE SELESAI====");
318 }

```

penjelasan:

heap disini adalah area memori yang digunakan untuk alokasi secara dinamis. Bagian-bagian memori yang dialokasikan dilakukan secara sembarang (no pattern). Maksudnya adalah kode-kode yang akan dieksekusi akan diletakkan pada lokasi penyimpanan dalam memori, namun lokasi ini tidak memiliki pattern atau tidak berpola (acak). Lokasi memori yang ditempati ini tidak akan diketahui sebelum runtime (saat dijalankan). Heap seringkali digunakan program untuk berbagai keperluan. Tapi intinya adalah heap dialokasikan untuk mensuplai memori tambahan yang tidak dialokasikan saat kompilasi. Alokasi ini dilakukan saat runtime, seiring berjalannya program.\

SEARCHING: Linear Search

11.4: Fungsi Linear Search pada single linkedlist

```

singleLinkedList.java x
341 public static void cariLinear()
342 {
343     if (awal==null) // jika senarai masih kosong
344         System.out.print("...MAAF SENARAI KOSONG...");
345     else // jika senarai tidak kosong
346     {
347         Scanner masukan = new Scanner(System.in);
348         System.out.print("Silakan masukkan nama yang anda cari : ");
349         String NAMACARI = masukan.nextLine();
350         boolean statusKetemu = false;
351         int i = 0;
352         int posisiKetemu=-1;
353         simpul bantu;
354         bantu = awal;
355         while (bantu != null)
356         {
357             if (NAMACARI.equals(bantu.nama))
358             {
359                 statusKetemu = true;
360                 posisiKetemu = i;
361             }
362             bantu = bantu.kanan;
363             i++;
364         }
365         System.out.println("Status Ketemu = " +statusKetemu + " di posisi ke" +posisiKetemu);
366     }
367 }
368 //-----bagian program utama-----
369 public static void main(String[] args)
370 {
371     inisialisasiSenaraiKosong();
372     tambahDepan();
373     tambahDepan();
374     tambahBelakang();
375     tambahBelakang();
376     hapus();

```

```

377 mengurutkanDataBubble_TeknikTukarHeap();
378 cariLinear();
379 cetakSenarai();
380 }
381 }
382

```

```

C:\Windows\system32\cmd.exe
TAMBAH DEPAN :
Silakan masukkan nama anda : Rocky
Silakan masukkan alamat anda : Jungle
Silakan masukkan umur anda : 35
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : akting
hobi ke-1 : boxing
hobi ke-2 : silat
Silakan masukkan IPK anda : 3,9
TAMBAH DEPAN :
Silakan masukkan nama anda : Jim
Silakan masukkan alamat anda : Bogor
Silakan masukkan umur anda : 24
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : nyetir
hobi ke-1 : game
hobi ke-2 : makan
Silakan masukkan IPK anda : 3,2
TAMBAH BELAKANG :
Silakan masukkan nama anda : Hailee
Silakan masukkan alamat anda : Queens
Silakan masukkan umur anda : 19
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : balet
hobi ke-1 : akting
hobi ke-2 : nyanyi
Silakan masukkan IPK anda : 3,6
TAMBAH BELAKANG :
Silakan masukkan nama anda : Salsa
Silakan masukkan alamat anda : Berbah
Silakan masukkan umur anda : 18
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : makan
hobi ke-1 : kpop
hobi ke-2 : chatting
Silakan masukkan IPK anda : 3,8
Silakan masukkan nama yang ingin dihapus : Jim
menghapus Jim dilakukan..
Banyaknya simpul = 3

===PROSES PENGURUTAN BUBBLE SELESAI=====
Silakan masukkan nama yang anda cari : Salsa
Status Ketemu = true di posisi ke2

NO NAMA ALAMAT UMUR JEKEL IPK
-----
Hailee Queens 19 P balet akting nyanyi 3.6
Rocky Jungle 35 L akting boxing silat 3.9
Salsa Berbah 18 P makan kpop chatting 3.8
-----
Press any key to continue . . .

```

penjelasan:

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 metode ini sulit diimplementasikan pada linkedlist.

Di situ saya mencari Salsa, program mencari satu per satu hingga Salsa ditemukan. Setelah ditemukan, maka muncul output “Status Ketemu: true di posisi ke2

KESIMPULAN

di pertemuan kali ini kita belajar tentang Bubble Sort dan Linear Search. Bubble sort adalah sebuah teknik pengurutan data dengan cara menukar dua data yang bersebelahan jika urutan dari data tersebut salah. Algoritma ini dapat mengurutkan data dari besar ke kecil (Ascending) dan kecil ke besar (Descending). Algoritma ini tidak cocok untuk set data dengan jumlah besar karena kompleksitas dari algoritma ini adalah $O(n^2)$ di mana n adalah jumlah item.

Linear Search merupakan sebuah teknik pencarian data dengan menelusuri semua data satu per satu. Apabila ditemukan kecocokan data maka program akan mengembalikan output, jika tidak pencarian akan terus berlanjut hingga akhir dari array tersebut. Algoritma ini tidak cocok untuk set data dengan jumlah besar karena kompleksitas dari algoritma ini adalah $O(n)$ di mana n adalah jumlah item. Jika data yang dicari berada pada paling akhir dari array, maka program harus menelusuri semua array terlebih dahulu.