

LAPORAN PRAKTIKUM STRUKTUR DATA

PERTEMUAN KE-8



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-8 **(SINGLE LINKEDLIST (SENARAI BERANTAI TUNGGAL))**

TUJUAN

Mahasiswa dapat mengimplementasikan penggunaan Single Linked List

DASAR TEORI

Linked list (daftar yang terhubung) atau disebut juga Senarai Berantai, adalah pengalokasian memori secara dinamis agar dapat digunakan untuk menyimpan deretan data (list). Mengapa disebut dinamis? Karena alokasi memori ini dilakukan dengan membuat simpul (heap) di memori. Jika pada larik/ array kita hanya dapat menyimpan data hanya dalam jumlah tertentu saja dan tidak dapat diubah (statis), maka pada linked list kita dapat menyimpan data secara lebih dinamis karena pengalokasian simpul baru di memori baru akan dilakukan pada saat diperlukan.

Single Linked List (Senarai Berantai Tunggal) adalah kumpulan heap/ obyek/ simpul/ node yang saling terhubung satu sama lain (linked) yang dimanfaatkan untuk menyimpan sederet data (list) dimana pada setiap heap yang ada terdapat satu buah pointer anggota (single) yang bertugas sebagai pointer pengait yang digunakan untuk mengkaitkan diri dengan simpul sejenis yang ada di sebelah kanannya. Kebanyakan orang menyingkat Single Linked List (Senarai Berantai Tunggal) hanya dengan sebutan Linked List (Senarai Berantai).

PRAKTIK 1

```
singleLinkedList.java x
1  import java.util.Scanner;
2  class simpul
3  {
4      //bagian deklarasi struktur record -----
5      String nama;
6      String alamat;
7      int umur;
8      char jekel;
9      String hobi[] = new String[3];
10     float ipk;
11     simpul kanan;
12 }
13 class singleLinkedList
14 {
15     public static simpul awal;
16     public static simpul akhir;
17     public static void inisialisasiSenaraiKosong()
18     {
19         awal = null;
20         akhir = null;
21     }
22     public static void tambahDepan()
23     {
24         //-----bagian entri data dari keyboard-----
25         String NAMA;
26         String ALAMAT;
27         int UMUR;
28         char JEKEL;
29         String HOBI[] = new String[3];
30         float IPK;
31         Scanner masukan = new Scanner(System.in);
32         int bacaTombol=0;
33         System.out.println("TAMBAH DEPAN : ");
34         System.out.print("Silakan masukkan nama anda : ");
35         NAMA = masukan.nextLine();
36         System.out.print("Silakan masukkan alamat anda : ");
```

```

37     ALAMAT = masukan.nextLine();
38     System.out.print("Silakan masukkan umur anda : ");
39     UMUR = masukan.nextInt();
40     System.out.print("Silakan masukkan Jenis Kelamin anda : ");
41     try
42     {
43         bacaTombol = System.in.read();
44     }
45         catch(java.io.IOException e)
46     {
47     }
48     JEKEL = (char)bacaTombol;
49     System.out.println("Silakan masukkan hobi (maks 3) : ");
50     System.out.print("hobi ke-0 : ");
51         HOBI[0] = masukan.next();
52     System.out.print("hobi ke-1 : ");
53         HOBI[1] = masukan.next();
54     System.out.print("hobi ke-2 : ");
55         HOBI[2] = masukan.next();
56     System.out.print("Silakan masukkan IPK anda : ");
57     IPK = masukan.nextFloat();
58     //-----bagian menciptakan & mengisi simpul baru-----
59     simpul baru;
60         baru = new simpul();
61         baru.nama = NAMA;
62         baru.alamat = ALAMAT;
63         baru.umur = UMUR;
64         baru.jekel = JEKEL;
65         baru.hobi[0] = HOBI[0];
66     baru.hobi[1] = HOBI[1];
67     baru.hobi[2] = HOBI[2];
68     baru.ipk = IPK;
69     //-----bagian mencangkokkan simpul baru ke dalam simpul lama-----
70     if (awal == null) // jika senarai masih kosong
71     {
72         awal = baru;
73     akhir = baru;
74     baru.kanan = null;
75     }
76     else // jika senarai tidak kosong
77     {
78         baru.kanan = awal;
79         awal = baru;
80     }
81     }
82     public static void tambahBelakang()
83     {
84         //-----bagian entri data dari keyboard-----
85         String NAMA;
86         String ALAMAT;
87         int UMUR;
88         char JEKEL;
89         String HOBI[] = new String[3];
90         float IPK;
91         Scanner masukan = new Scanner(System.in);
92         int bacaTombol=0;
93         System.out.println("TAMBAH BELAKANG : ");
94         System.out.print("Silakan masukkan nama anda : ");
95             NAMA = masukan.nextLine();
96         System.out.print("Silakan masukkan alamat anda : ");
97             ALAMAT = masukan.nextLine();
98         System.out.print("Silakan masukkan umur anda : ");
99             UMUR = masukan.nextInt();
100        System.out.print("Silakan masukkan Jenis Kelamin anda : ");
101        try
102        {
103            bacaTombol = System.in.read();
104        }
105            catch(java.io.IOException e)
106        {
107        }
108        JEKEL = (char)bacaTombol;

```

```

109 System.out.println("Silakan masukkan hobi (maks 3) : ");
110 System.out.print("hobi ke-0 : ");
111         HOBI[0] = masukan.next();
112 System.out.print("hobi ke-1 : ");
113         HOBI[1] = masukan.next();
114 System.out.print("hobi ke-2 : ");
115         HOBI[2] = masukan.next();
116 System.out.print("Silakan masukkan IPK anda : ");
117 IPK = masukan.nextFloat();
118 //-----bagian menciptakan & mengisi simpul baru-----
119 simpul baru;
120 baru = new simpul();
121 baru.nama = NAMA;
122 baru.alamat = ALAMAT;
123 baru.umur = UMUR;
124 baru.jekel = JEKEL;
125 baru.hobi[0] = HOBI[0];
126 baru.hobi[1] = HOBI[1];
127 baru.hobi[2] = HOBI[2];
128 baru.ipk = IPK;
129 //-----bagian mencangkokkan simpul baru ke dalam simpul lama-----
130 if (awal == null) // jika senarai kosong
131 {
132     awal = baru;
133     akhir = baru;
134     baru.kanan = null; }
135 else // jika senarai tidak kosong
136 {
137     akhir.kanan = baru;
138     akhir = baru;
139     baru.kanan = null;
140 }
141 }
142 public static void cetakSenarai()
143 {
144     if (awal==null) // jika senarai masih kosong
145         System.out.print("....MAAF SENARAI KOSONG....");
146     else // jika senarai tidak kosong
147     {
148         System.out.println("-----");
149         System.out.println("NO NAMA ALAMAT UMUR JEKEL IPK ");
150         System.out.println("-----");
151         simpul bantu;
152         bantu = awal;
153         while (bantu != null)
154         {
155             System.out.print (bantu.nama + "\t ");
156             System.out.print (bantu.alamat + "\t ");
157             System.out.print (bantu.umur + "\t ");
158             System.out.print (bantu.jekel + "\t ");
159             System.out.print (bantu.hobi[0] + "\t ");
160             System.out.print (bantu.hobi[1] + "\t ");
161             System.out.print (bantu.hobi[2] + "\t ");
162             System.out.println(bantu.ipk);
163             bantu = bantu.kanan;
164         }
165         System.out.println("-----");
166     }
167 }
168 //-----bagian program utama-----
169 public static void main(String[] args)
170 {
171     inisialisasiSenaraiKosong();
172     tambahDepan();
173     tambahDepan();
174     tambahDepan();
175     tambahBelakang();
176     tambahBelakang();
177     tambahBelakang();
178     cetakSenarai();
179 }
180 }

```

```

C:\Windows\system32\cmd.exe
TAMBAH DEPAN :
Silakan masukkan nama anda : Jacob
Silakan masukkan alamat anda : Norwich
Silakan masukkan umur anda : 19
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : bola
hobi ke-1 : futsal
hobi ke-2 : renang
Silakan masukkan IPK anda : 2,8
TAMBAH DEPAN :
Silakan masukkan nama anda : Emma
Silakan masukkan alamat anda : Stoke
Silakan masukkan umur anda : 18
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : belajar
hobi ke-1 : baca
hobi ke-2 : puisi
Silakan masukkan IPK anda : 3,9
TAMBAH DEPAN :
Silakan masukkan nama anda : Rooney
Silakan masukkan alamat anda : Everton
Silakan masukkan umur anda : 22
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : voli
hobi ke-1 : golf
hobi ke-2 : karate
Silakan masukkan IPK anda : 3,2
TAMBAH BELAKANG :
Silakan masukkan nama anda : Lisa
Silakan masukkan alamat anda : Phuket
Silakan masukkan umur anda : 20
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : dance
hobi ke-1 : nyanyi
hobi ke-2 : nonton
Silakan masukkan IPK anda : 3,1
TAMBAH BELAKANG :
Silakan masukkan nama anda : James
Silakan masukkan alamat anda : Sleman
Silakan masukkan umur anda : 21
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi (maks 3) :
hobi ke-0 : masak
hobi ke-1 : gitar
hobi ke-2 : drum
Silakan masukkan IPK anda : 3,3
TAMBAH BELAKANG :
Silakan masukkan nama anda : Lynn
Silakan masukkan alamat anda : Texas
Silakan masukkan umur anda : 23
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi (maks 3) :
hobi ke-0 : jurnalis
hobi ke-1 : nulis
hobi ke-2 : jajan
Silakan masukkan IPK anda : 3,5
=====
NO NAMA ALAMAT UMUR JEKEL IPK
=====
Rooney Everton 22 L bola voli golf karate 3.2
Emma Stoke 18 P belajar baca puisi 3.9
Jacob Norwich 19 L bola futsal renang 2.8
Lisa Phuket 20 P dance nyanyi nonton 3.1
James Sleman 21 L masak gitar drum 3.3
Lynn Texas 23 P jurnalis nulis jajan 3.5
=====
Press any key to continue . . . _

```

penjelasan:

Deretan nama yang muncul terlebih dahulu dari atas ke bawah adalah Rooney, Emma, Jacob, Lisa, James, dan Lynn. Urutannya berbeda dengan data yang saya input tadi. Mengapa? Coba perhatikan hal yang terjadi pada Tambah Depan. Tambah depan akan mengisi posisi yang bisa disebut dengan Head. Data pertama yang masuk lewat head akan mengisi ruang kosong di paling bawah. Contohnya seperti kasus program diatas: Tambah depan pertama adalah Jacob yang masuk duluan sehingga ada di paling bawah, lalu diikuti Emma yang nanti menempati urutan tengah, Lalu diikuti dengan Rooney yang akan menempati urutan paling atas. Hal ini bisa diibaratkan saat kita menaruh barang di box, barang pertama yang di input akan menjadi barang pertama yang paling bawah. Dengan perandaian itu bisa dibilang Jacob adalah barang pertama yang dimasukkan sehingga dia ada di posisi paling bawah.

Beda halnya dengan Tambah Belakang. Tambah Belakang berarti sebuah tempat yang memasukkan data nya lewat belakang atau bisa disebut dengan Tail. Untuk kasus ini, data pertama yang diinput akan menjadi data di paling atas (karena memasukkannya dari bawah ke atas). Dalam program ini kasusnya: Lisa dimasukkan dari bawah (tail) sehingga masuk ke paling atas Tambah Belakang (di bawah Jacob yaitu posisi paling bawah Tambah Depan. Lalu diikuti data-data selanjutnya.

PRAKTIK 2

di program utama, menambah perintah:

tambahBelakang();

tambahBelakang();

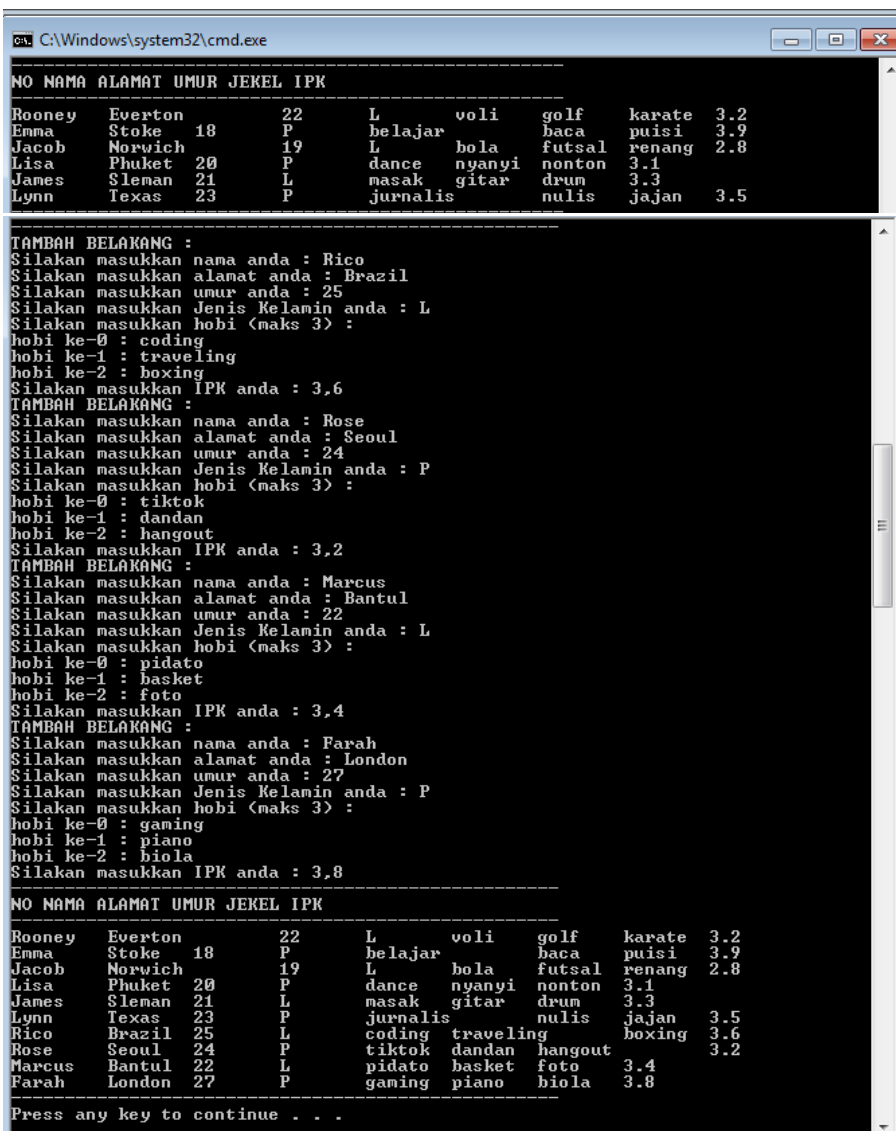
tambahBelakang();

tambahBelakang();

cetakSenarai();

```
167 }
168 //-----bagian program utama-----
169 public static void main(String[] args)
170 {
171     inisialisasiSenaraiKosong();
172     tambahDepan();
173     tambahDepan();
174     tambahDepan();
175     tambahBelakang();
176     tambahBelakang();
177     tambahBelakang();
178     cetakSenarai();
179     tambahBelakang();
180     tambahBelakang();
181     tambahBelakang();
182     tambahBelakang();
183     cetakSenarai();
184 }
185 }
```

output:



```
C:\Windows\system32\cmd.exe

NO NAMA ALAMAT UMUR JEKEL IPK
-----
Rooney Everton 22 L voli golf karate 3.2
Emma Stoke 18 P belajar baca puisi 3.9
Jacob Norwich 19 L bola futsal renang 2.8
Lisa Phuket 20 P dance nyanyi nonton 3.1
James Sleman 21 L masak gitar drum 3.3
Lynn Texas 23 P jurnalis nulis jajan 3.5

TAMBAH BELAKANG :
Silakan masukkan nama anda : Rico
Silakan masukkan alamat anda : Brazil
Silakan masukkan umur anda : 25
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi <maks 3> :
hobi ke-0 : coding
hobi ke-1 : traveling
hobi ke-2 : boxing
Silakan masukkan IPK anda : 3,6
TAMBAH BELAKANG :
Silakan masukkan nama anda : Rose
Silakan masukkan alamat anda : Seoul
Silakan masukkan umur anda : 24
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi <maks 3> :
hobi ke-0 : tiktok
hobi ke-1 : dandan
hobi ke-2 : hangout
Silakan masukkan IPK anda : 3,2
TAMBAH BELAKANG :
Silakan masukkan nama anda : Marcus
Silakan masukkan alamat anda : Bantul
Silakan masukkan umur anda : 22
Silakan masukkan Jenis Kelamin anda : L
Silakan masukkan hobi <maks 3> :
hobi ke-0 : pidato
hobi ke-1 : basket
hobi ke-2 : foto
Silakan masukkan IPK anda : 3,4
TAMBAH BELAKANG :
Silakan masukkan nama anda : Farah
Silakan masukkan alamat anda : London
Silakan masukkan umur anda : 27
Silakan masukkan Jenis Kelamin anda : P
Silakan masukkan hobi <maks 3> :
hobi ke-0 : gaming
hobi ke-1 : piano
hobi ke-2 : biola
Silakan masukkan IPK anda : 3,8

NO NAMA ALAMAT UMUR JEKEL IPK
-----
Rooney Everton 22 L voli golf karate 3.2
Emma Stoke 18 P belajar baca puisi 3.9
Jacob Norwich 19 L bola futsal renang 2.8
Lisa Phuket 20 P dance nyanyi nonton 3.1
James Sleman 21 L masak gitar drum 3.3
Lynn Texas 23 P jurnalis nulis jajan 3.5
Rico Brazil 25 L coding traveling boxing 3.6
Rose Seoul 24 P tiktok dandan hangout 3.2
Marcus Bantul 22 L pidato basket foto 3.4
Farah London 27 P gaming piano biola 3.8

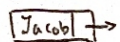
Press any key to continue . . .
```


penjelasan:

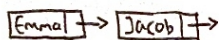
deretan nama yang muncul dari atas (Head) ke bawah (Tail) adalah Rooney, Emma, Jacob, Lisa, James, Lynn, Rico, Rose, Marcus, Farah. Setelah Senarai pertama muncul, ada perintah yang baru yang melakukan penambahan 4 data Tambah Belakang yaitu Rico, Rose, Marcus, Farah. Karena ditambahkan dari bawah (Tail), maka Rico menjadi teratas diantara 4 data terbaru tersebut. Rico langsung masuk dari bawah ke atas menempati posisi dibawah Lynn. Lalu diikuti Rose, Marcus, dan Farah. Lalu Senarai ke-2 muncul untuk menggabungkan dan menampilkan data di senarai 1 dan data-data Tambah Belakang yang baru dimasukkan

TUGAS

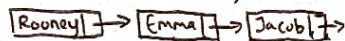
Tambah depan: Jacob



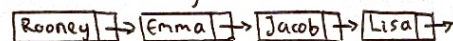
Tambah depan: Emma



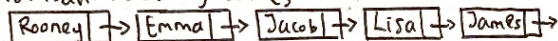
Tambah depan: Rooney



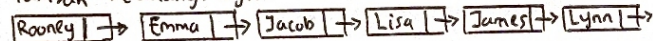
Tambah belakang: Lisa



Tambah belakang: James



Tambah belakang: Lynn



→ Senarai 1 muncul

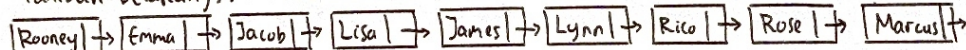
Tambah belakang: Rico



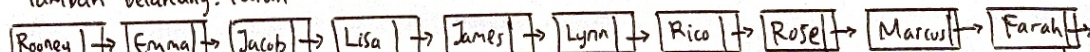
Tambah belakang: Rose



Tambah belakang: Marcus



Tambah belakang: Farah



head

tail

Senarai 2 muncul

KESIMPULAN

Di dalam konsep Single Linked List ada 3 unsur pendukung yang penting, yaitu :

1. **Penunjuk** (disebut juga dengan *pointer*)
2. **Simpul** (disebut juga dengan *list* atau *node* atau *heap*)
3. **Senarai Berantai Tunggal** atau Single Linked List.

Senarai Berantai Tunggal adalah kumpulan simpul-simpul yang terhubung satu dengan yang lain