

MODUL 8

SINGLE LINKEDLIST (SENARAI BERANTAI TUNGGAL)



CAPAIAN PEMBELAJARAN

Mahasiswa dapat mengimplementasikan penggunaan Single Linked List



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. TextPad



DASAR TEORI

Untuk mempelajari bagaimana sebuah linkedlist terbentuk anda dapat mempelajari kembali pelajaran pada modul 7 mengenai Single Linkedlist.

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.

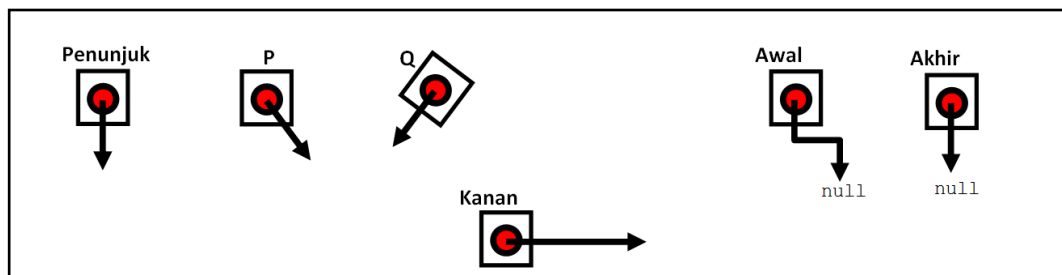
Single Linked List (Senarai Berantai Tunggal) adalah kumpulan heap/ obyek/ simpul/ node yang saling terhubung satu sama lain (*linked*) yang dimanfaatkan untuk menyimpan sederet data (*list*) dimana pada setiap heap yang ada terdapat **satu buah pointer anggota (*single*)** yang bertugas sebagai pointer 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)

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.

Penunjuk (pointer)

Penunjuk atau pointer yang digunakan di sini sama dengan yang telah anda pelajari pada modul 7, Pada linkedlist, pointer akan dipergunakan untuk menunjuk sebuah simpul atau heap. Ilustrasi dari sebuah pointer dapat dilihat pada gambar 8.1 berikut.

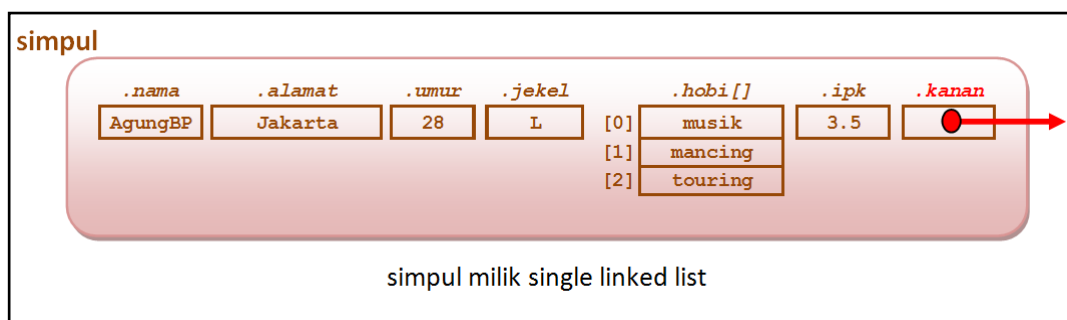


Gambar 8.1

Penunjuk dapat menunjuk ke sebuah simpul, ataupun ke sebuah tempat kosong (*null*).

Simpul (Node)

Simpul yang digunakan dalam bab ini adalah *simpul tunggal* yang diilustrasikan seperti gambar 8.2. (Gambar 8.2 merupakan dimodifikasi dari Gambar 2.2 modul 2, dari yang sebelumnya bertipe /kelas “*biodataMahasiswa*” diganti menjadi tipe/ kelas “*simpul*”, dengan tambahan anggota bernama “*kanan*”)



Gambar 8.2

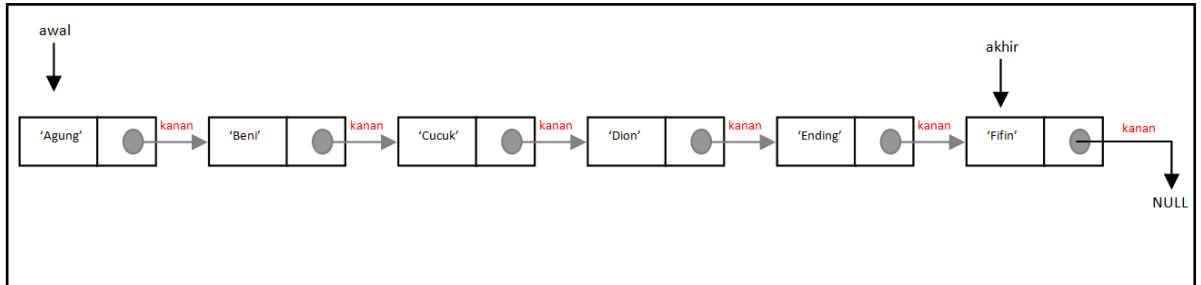
Simpul yang demikian disebut ‘simpul tunggal’ karena simpul ini hanya mempunyai sebuah penunjuk, yaitu ‘kanan’. Artinya simpul ini hanya dapat menunjuk ke simpul yang serupa dengan dirinya yang letaknya disebelah kanannya.

Simpul serta penunjuknya dideklarasikan dengan cara :

```
class simpul
{ //bagian deklarasi struktur record -----
-
    String nama;
    String alamat;
    int umur;
    char j_kel;
    String hobi[] = new String[3];
    float ipk;
    simpul kanan; //bagian yang ditambahkan dari program 2.2
}
```

Senarai Berantai Tunggal (Single Linked List)

Senarai Berantai Tunggal adalah kumpulan simpul-simpul yang terhubung satu dengan yang lain yang digambarkan sebagai berikut :



Gambar 8.3

Senarai Berantai Tunggal di deklarasikan dengan cara sebagai berikut :

```
class senarai
{ protected
    simpul awal;
    simpul akhir;
  public
    public static void inisialisasiSenarai()
    { awal = null;
      akhir = null;
    }
    public static void tambahDepan()
    { ..... }
    public static void tambahBelakang()
    { ..... }
    public static void tambahTengah()
    { ..... }
    public static void cetakSenarai()
    { ..... }
    public static hapus()
    { ..... }
    public static void main()
    { ..... }
}
```

Di dalam pengelolaan data menggunakan Single LinkedList terdapat beberapa operasi yang dapat dilakukan antara lain :

- | | |
|--|-------------------------------|
| a. Operasi Tambah Data di Depan | (dipelajari pada Modul 8 ini) |
| b. Operasi Tambah Data di Belakang | (dipelajari pada Modul 8 ini) |
| c. Operasi Menampilkan (mencetak) semua Data | (dipelajari pada Modul 8 ini) |
| d. Operasi Tambah Data di Tengah | (dipelajari pada Modul 9) |
| e. Operasi Menghapus Data | (dipelajari pada Modul 9) |
| f. Operasi Pencarian Data | (dipelajari pada Modul 11) |
| g. Operasi Pengeditan Data | (dipelajari pada Modul 11) |
| h. Operasi Pengurutan Data | (dipelajari pada Modul 11) |

Pada modul 8 ini kita akan mempelajari bersama operasi Tambah data di depan, belakang dan bagaimana menampilkan isi single linkedlist



PRAKTIK

1. Praktek 1

Tuliskan program berikut ini.

```
import java.util.Scanner;
class simpul
{ //bagian deklarasi struktur record -----
    String nama;
    String alamat;
    int    umur;
    char   jekel;
    String hobi[] = new String[3];
    float  ipk;
    simpul kanan;
}

class singleLinkedList
{
    public static simpul awal;
    public static simpul akhir;

    public static void inisialisasiSenaraiKosong()
    {
        awal = null;
        akhir = null;
    }

    public static void tambahDepan()
    {
        //-----bagian entri data dari keyboard-----
        String NAMA;
        String ALAMAT;
        int    UMUR;
        char   JEKEL;
        String HOBI[] = new String[3];
        float  IPK;
        Scanner masukan = new Scanner(System.in);
        int bacaTombol=0;

        System.out.println("TAMBAH DEPAN : ");
        System.out.print("Silakan masukkan nama anda : ");
        NAMA = masukan.nextLine();
        System.out.print("Silakan masukkan alamat anda : ");
        ALAMAT = masukan.nextLine();
        System.out.print("Silakan masukkan umur anda : ");
        UMUR = masukan.nextInt();
        System.out.print("Silakan masukkan Jenis Kelamin anda : ");
        try
        { bacaTombol = System.in.read();
        }
        catch(java.io.IOException e)
        {
        }
        JEKEL = (char)bacaTombol;
        System.out.println("Silakan masukkan hobi (maks 3) : ");
        System.out.print("hobi ke-0 : ");
        HOBI[0] = masukan.next();
        System.out.print("hobi ke-1 : ");
        HOBI[1] = masukan.next();
        System.out.print("hobi ke-2 : ");
```

```

HOBI[2] = masukan.next();
System.out.print("Silakan masukkan IPK anda : ");
IPK = masukan.nextFloat();

//-----bagian menciptakan & mengisi simpul baru-----
simpul baru;
baru = new simpul();
baru.nama = NAMA;
baru.alamat = ALAMAT;
baru.umur = UMUR;
baru.jekel = JEKEL;
baru.hobi[0] = HOBI[0];
baru.hobi[1] = HOBI[1];
baru.hobi[2] = HOBI[2];
baru.ipk = IPK;

//-----bagian mencangkokkan simpul baru ke dalam simpul lama-----
if (awal == null) // jika senarai masih kosong
{ awal = baru;
  akhir = baru;
  baru.kanan = null;
}
else // jika senarai tidak kosong
{ baru.kanan = awal;
  awal = baru;
}
}

public static void tambahBelakang()
{
  //-----bagian entri data dari keyboard-----
  String NAMA;
  String ALAMAT;
  int UMUR;
  char JEKEL;
  String HOBI[] = new String[3];
  float IPK;
  Scanner masukan = new Scanner(System.in);
  int bacaTombol=0;

  System.out.println("TAMBAH BELAKANG : ");
  System.out.print("Silakan masukkan nama anda : ");
  NAMA = masukan.nextLine();
  System.out.print("Silakan masukkan alamat anda : ");
  ALAMAT = masukan.nextLine();
  System.out.print("Silakan masukkan umur anda : ");
  UMUR = masukan.nextInt();
  System.out.print("Silakan masukkan Jenis Kelamin anda : ");
  try
  { bacaTombol = System.in.read();
  }
  catch(java.io.IOException e)
  {
  }
  JEKEL = (char)bacaTombol;
  System.out.println("Silakan masukkan hobi (maks 3) : ");
  System.out.print("hobi ke-0 : ");
  HOBI[0] = masukan.next();
  System.out.print("hobi ke-1 : ");
  HOBI[1] = masukan.next();
  System.out.print("hobi ke-2 : ");
  HOBI[2] = masukan.next();
  System.out.print("Silakan masukkan IPK anda : ");
  IPK = masukan.nextFloat();

  //-----bagian menciptakan & mengisi simpul baru-----
  simpul baru;
  baru = new simpul();
  baru.nama = NAMA;
  baru.alamat = ALAMAT;

```

```

baru.umur    = UMUR;
baru.jekel   = JEKEL;
baru.hobi[0] = HOBI[0];
baru.hobi[1] = HOBI[1];
baru.hobi[2] = HOBI[2];
baru.ipk     = IPK;

//-----bagian mencangkokkan simpul baru ke dalam simpul lama-----
if (awal == null)          // jika senarai kosong
{
    awal = baru;
    akhir = baru;
    baru.kanan = null;
}
else // jika senarai tidak kosong
{
    akhir.kanan = baru;
    akhir = baru;
    baru.kanan = null;
}
}

public static void cetakSenarai()
{
    if (awal==null) // jika senarai masih kosong
        System.out.print("...MAAF SENARAI KOSONG...");
    else // jika senarai tidak kosong
    {
        System.out.println("-----");
        System.out.println("NO NAMA          ALAMAT          UMUR      JEKEL      IPK ");
        System.out.println("-----");
        simpul bantu;
        bantu = awal;
        while (bantu != null)
        {
            System.out.print (bantu.nama + "\t ");
            System.out.print (bantu.alamat + "\t ");
            System.out.print (bantu.umur + "\t ");
            System.out.print (bantu.jekel + "\t ");
            System.out.print (bantu.hobi[0] + "\t ");
            System.out.print (bantu.hobi[1] + "\t ");
            System.out.print (bantu.hobi[2] + "\t ");
            System.out.println(bantu.ipk);
            bantu = bantu.kanan;
        }
        System.out.println("-----");
    }
}

//-----bagian program utama-----
public static void main(String[] args)
{
    inisialisasiSenaraiKosong();
    tambahDepan();
    tambahDepan();
    tambahDepan();
    tambahBelakang();
    tambahBelakang();
    tambahBelakang();
    cetakSenarai();
}
}

```

Operasi-operasi pada **Senarai Berantai Tunggal** dapat dilakukan di sini.
(tambah depan, tengah, belakang, hapus, cetak, dll)

Program 8.1

Setelah anda tulis program di atas, tambahkanlah pada program utama (pada bagian yang dititik-titik) perintah-perintah berikut ini :

```

    tambahDepan();
    tambahDepan();
    tambahDepan();

```

dengan mengisi sembarang data mahasiswa. Catatlah urutan data yang anda entri. Kemudian tambahkanlah di bawahnya perintah untuk mencetak senarai berikut :

```
cetakSenarai();
```

Jalankan program dan amati apa yang terjadi pada hasil runningnya? Deretan *nama* siapa sajakah yang muncul lebih dahulu dan yang terkemudian? Bandingkan dengan urutan data yang anda entri sebelumnya. Samakah urutannya? Catatlah dan simpulkan dalam laporan anda.

2. Praktek 2

Sekarang, tambahkanlah perintah berikut ini dan letakkanlah tepat setelah perintah untuk mencetak senarai di atas.

```
tambahBelakang();  
tambahBelakang();  
tambahBelakang();  
tambahBelakang();
```

lalu tambahkanlah lagi di bawahnya perintah untuk mencetak senarai sebagai berikut :

```
cetakSenarai();
```

kemudian jalankan program dan amati apa yang terjadi pada hasil runningnya? Deretan *nama* siapa sajakah yang muncul? Catatlah dan simpulkan dalam laporan anda.



LATIHAN

1.
2.



TUGAS

1. Dengan menggunakan data yang ada isikan, ilustrasikan/ Gambarkan proses menambah di depan dan menambah data di belakang. Lakukan hal tersebut pada kertas tersendiri dan ditulis tangan (bukan diketik/print).
2.



REFERENSI

Disadur dari Buku Ajar Struktur Data Menggunakan Java, Agung Budi Prasetyo, 2017, hal: 81-87, <http://agungbudiprasetyo.atspace.com/buku/index.html>, diakses pada 12:07 PM 8/06/2019.