

MODUL 9

SINGLE LINKEDLIST (LANJUTAN)



CAPAIAN PEMBELAJARAN

Mahasiswa dapat mengimplementasikan penggunaan Single Linked List



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. TextPad



DASAR TEORI

Pada pertemuan terdahulu kita telah mempraktekkan penggunaan single linkedlist untuk mengelola data, khususnya untuk menginisialisasi linkedlist, menambah data di depan, menambah data di belakang, dan menampilkan isi linkedlist.

Pada pertemuan ini kita akan melanjutkan pengelolaan data dengan single linkedlist khususnya untuk menambah data di tengah dan menghapus data.



PRAKTIK

1. Praktek 1

Tuliskan program berikut ini menggunakan TextPad

```
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()
    {
        .....
    }

    public static void tambahDepan()
    {
        .....
    }

    public static void tambahBelakang()
    {
        .....
    }

    public static void cetakSenarai()
    {
        .....
    }

    public static int hitungJumlahSimpul()
    {
        int N = 0;
        simpul bantu;
        bantu = awal;
        while (bantu!=null)
        {
            N++;
            bantu = bantu.kanan;
        }
        return(N);
    }

    public static void tambahTengah()
    {
        Scanner masukan = new Scanner(System.in);
        System.out.println("Tentukan Lokasi Penambahan Data");
        int LOKASI = masukan.nextInt();

        int jumlahSimpulYangAda = hitungJumlahSimpul();
        if (LOKASI==1)
            System.out.println("Lakukan penambahan di depan");

        else if (LOKASI > jumlahSimpulYangAda)
            System.out.println("Lakukan penambahan di belakang");

        else
        {
            //-----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 TENGAH : ");
            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 menemukan posisi yang dikehendaki-----
simpul bantu;
bantu = awal;
int N = 1;
while ((N<LOKASI-1) && (bantu!=akhir))
{ bantu = bantu.kanan;
  N++;
}

//-----bagian menciptakan & mengisi simpul baru-----
simpul 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 linkedlist lama-----
bantu.kanan = baru;
baru.kanan = bantu.kanan;
}

}

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

```

Lakukan penambahan ditengah disini.

Program 9.1

Setelah anda tulis program di atas, tambahkanlah pada program utama perintah untuk menambahkan didepan (4x) dan untuk menambahkan di belakang (4x) seperti berikut ini :

```

        tambahDepan();
        tambahDepan();
        tambahDepan();
        tambahDepan();
        tambahBelakang();
        tambahBelakang();
        tambahBelakang();
        tambahBelakang();
        tambahTengah();
        cetakSenarai();

```

Jalankan program dan amati apa yang terjadi pada hasil runningnya? Apakah data yang baru saja anda tambahkan di tengah linkedlist berhasil? Catatlah dan simpulkan dalam laporan anda.

2. Praktek 2

Sekarang kembangkan program anda seperti berikut.

```

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()
    { .....
    }

    public static void tambahDepan()
    { .....
    }

    public static void tambahBelakang()
    { .....
    }

    public static void cetakSenarai()
    { .....
    }

    public static int hitungJumlahSimpul()
    { .....
    }

    public static void tambahTengah()
    { .....
    }

    public static void hapus()

```

```

{ if (awal == null) // jika senarai masih kosong
{ System.out.println("senarai kosong, menghapus tidak dapat dilakukan");
}
else // jika senarai tidak kosong
{

Scanner masukan = new Scanner(System.in);
System.out.print("Silakan masukkan nama yang ingin dihapus : ");
String NAMACARI = masukan.nextLine();

if (awal == akhir) //jika hanya ada sebuah simpul
{ if (awal.nama.equals(NAMACARI))
{ System.out.println("menghapus "+NAMACARI+" dilakukan..");
inisialisasiSenaraiKosong();
}
else
System.out.println("data " +NAMACARI+" tidak ditemukan");
}

else if (awal.nama.equals(NAMACARI))//jika nama ditemukan di awal
{ System.out.println("menghapus "+NAMACARI+" dilakukan..");
awal = awal.kanan;
}

else
{ simpul bantu;
bantu = awal;
while (bantu.kanan.nama.equals(NAMACARI)==false)
{ bantu = bantu.kanan;
if (bantu.kanan == null) break;
}
if ((bantu== akhir) && (akhir.nama.equals(NAMACARI)==false))
{ System.out.println("data " +NAMACARI+" tidak ditemukan");
}
else if (akhir.nama.equals(NAMACARI))//jika nama ditemukan di akhir
{ bantu.kanan = null;
akhir = bantu;
}
else
{ System.out.println("menghapus "+NAMACARI+" dilakukan..");
bantu.kanan = bantu.kanan.kanan;
}
}
}
}

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

```

Lakukan penghapusan disini.

Program 9.2

Setelah anda tulis program di atas, tambahkanlah pada program utama perintah untuk menambahkan didepan (4x), untuk menambahkan di belakang (4x), dan menambahkan ditengah (1x) seperti berikut ini :

```
tambahDepan() ;  
tambahDepan() ;  
tambahDepan() ;  
tambahDepan() ;  
tambahBelakang() ;  
tambahBelakang() ;  
tambahBelakang() ;  
tambahBelakang() ;  
tambahTengah() ;  
hapus() ;  
cetakSenarai() ;
```

Jalankan program dan amati apa yang terjadi pada hasil runningnya? Apakah data yang baru saja anda hapus dari linkedlist berhasil? Catatlah dan simpulkan dalam laporan anda.



LATIHAN

1.
2.



TUGAS

1. Dengan menggunakan data yang ada isikan, ilustrasikanlah proses menambah data di tengah. Ilustrasikanlah juga proses menghapus data (hapus depan/ tengah/ belakang sangat tergantung oleh keadaan di manakah data anda ditemukan). 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: 88-93, <http://agungbudiprasetyo.atspace.com/buku/index.html>, diakses pada 12:07 PM 8/06/2019.

MODUL 10

DOUBLE LINKEDLIST (SENARAI BERANTAI GANDA)



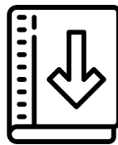
CAPAIAN PEMBELAJARAN

Mahasiswa dapat mengimplementasikan penggunaan Double Linked List



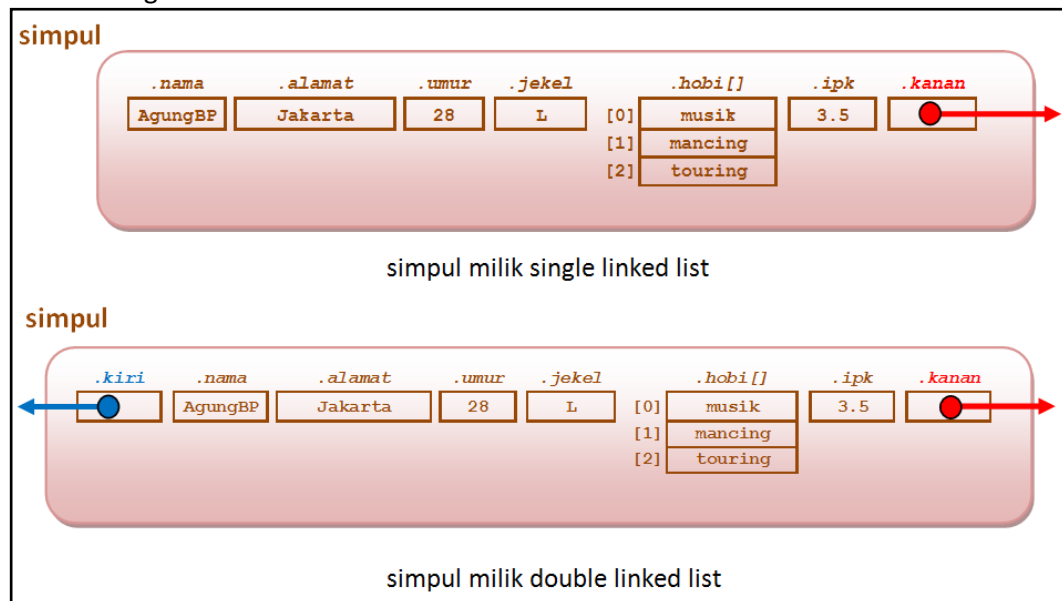
KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. TextPad



DASAR TEORI

Double Linked list atau disebut juga Senarai Berantai Ganda, hampir sama dengan *Single Linked List*. yaitu adalah pengalokasian memori secara dinamis. Bedanya adalah pada Double Linked List setiap simpul yang ada memiliki 2 buah penunjuk yang digunakan untuk mengkaitkan diri dengan simpul-simpul lain di kiri dan kanan. Perhatikan gambar berikut ini.



Gambar 10.1

Perbedaan lainnya adalah classnya (struktur)