

MODUL 10

DOUBLE LINKEDLIST (SENARAI BERANTAI GANDA)



CAPAIAN PEMBELAJARAN

Mahasiswa dapat mengimplementasikan penggunaan Double Linked List



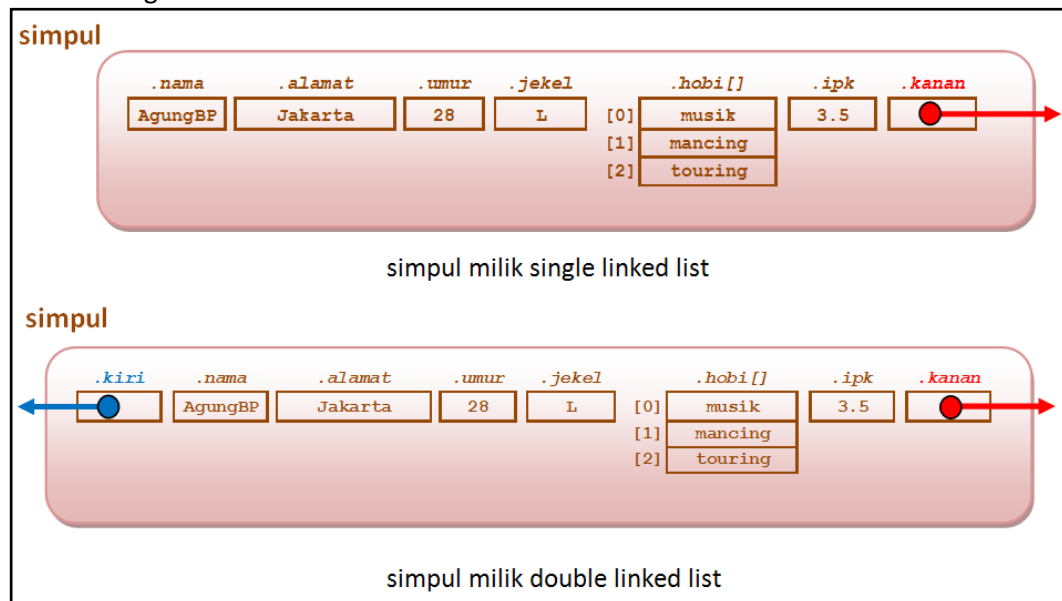
KEBUTUHAN ALAT/BAHAN/SOFTWARE

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)

Simpul milik Single Linkedlist :

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

Simpul milik Double Linkedlist :

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

Oleh karena strukturnya kelasnya berbeda, maka metode dalam penanganan datanya pun juga ikut berubah seperti tambah depan, tengah belakang, hapus depan, tengah, belakang, dan lain-lain. Berikut ini akan disajikan programnya.

```
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 kiri;
    simpul kanan;
}

class senaraiGanda
{
    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.kiri = null;
    baru.kanan = null;
}
else // jika senarai tidak kosong
{   baru.kanan = awal;
    awal.kiri = baru;
    awal = baru;
    awal.kiri = null;
}
}

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.kiri = null;
baru.kanan = null;
}
else // jika senarai tidak kosong
{ baru.kiri = akhir;
akhir.kanan = baru;
akhir = baru;
akhir.kanan = null;
}
}

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

public static void tambahTengah()
{
//-----bagian menentukan lokasi target-----
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 i = 1;
    while ((i<LOKASI) && (bantu!=akhir))
    { bantu = bantu.kanan;
      i++;
    }
    //-----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-----
    baru.kiri    = bantu.kiri;
    baru.kiri.kanan = baru;
    baru.kanan    = bantu;
    bantu.kiri    = baru;
}

}

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;
awal.kiri = null;
}
else
{ simpul bantu;
bantu = awal.kanan;
while (bantu.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
{
akhir = bantu.kiri;
akhir.kanan = null;
}
else
{ System.out.println("menghapus "+NAMACARI+" dilakukan..");
bantu.kanan.kiri = bantu.kiri;
bantu.kiri.kanan = bantu.kanan;
}
}
}
}

public static void cetakSenaraiMaju()
{
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("-----");
}
}

```

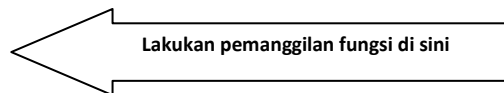
```

    }
}

public static void cetakSenaraiMundur()
{
    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 = akhir;
        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.kiri;
        }
        System.out.println("-----");
    }
}

public static void main(String[] args)
{
    inisialisasiSenaraiKosong();
    tambahDepan()
    tambahDepan()
    tambahDepan()
    tambahBelakang()
    tambahBelakang()
    cetakSenaraiMaju()
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
    .....
}
}

```



Program 10.1



PRAKTIK

1. Praktik 1

Tuliskan program 10.1 menggunakan TextPad kemudian lakukan penambahan data di depan, dibelakang dan di tengah. Lakukan juga menghapus data. Apakah berhasil? tunjukkan masing-masing hasil running outnya.

2. Praktik 2

Salah satu kelebihan double linkedlist adalah model linkedlist ini dapat mencetak baik secara maju maupun secara mundur. Tugas anda lakukan kedua kemampuan tersebut.



LATIHAN

1.
2.



TUGAS

1. Dengan menggunakan data yang ada isikan, ilustasikanlah proses menambah data di depan, di tengah dan di belakang. Ilustasikanlah 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).



REFERENSI

Disadur dari Buku Ajar Struktur Data Menggunakan Java, Agung Budi Prasetyo, 2017, hal: 94-101, <http://agungbudiprasetyo.atspace.com/buku/index.html>, diakses pada 12:07 PM 8/06/2019.