**LAPORAN**

**PRAKTIKUM STRUKTUR DATA**

**Pertemuan ke – 9**

****

Disusun Oleh :

Rendra Eka Herlambang

175410028

**LABORATORIUM TERPADU**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER AKAKOM YOGYAKARTA**

**2018**

**DOUBLE LINKEDLIST**

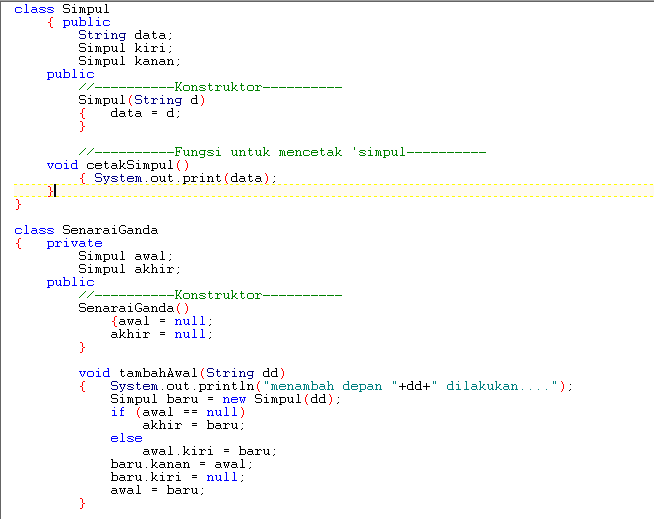
1. **TUJUAN**

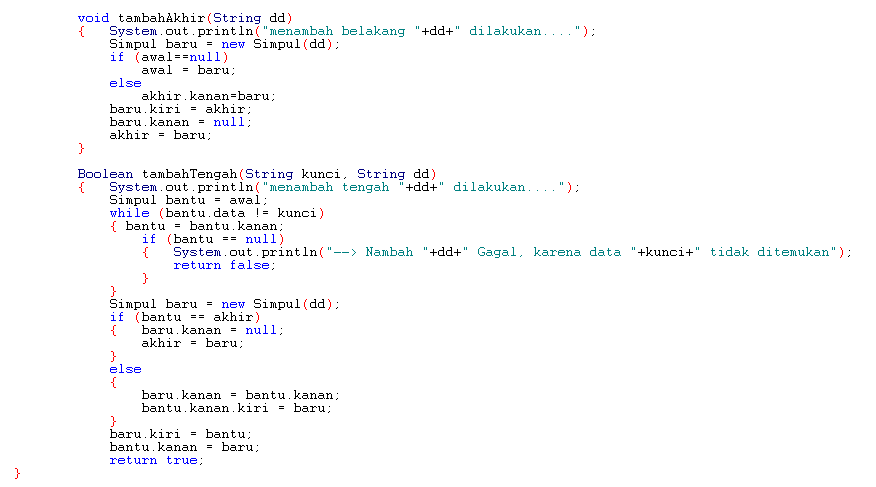
-Mahasiswa dapat membuat program dengan double linked list

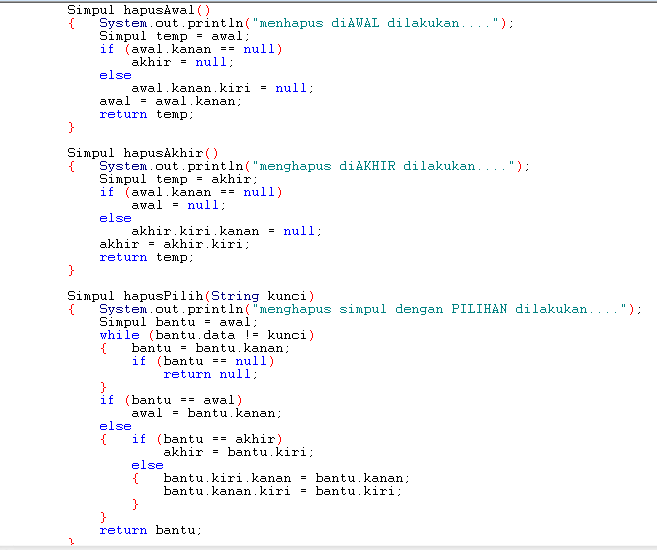
- Mahasiswa mampu mengimplementasikan double linked list.

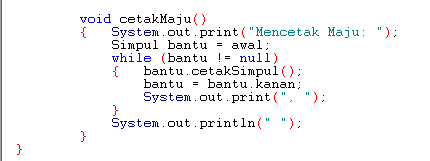
- Mahasiswa mampu mengimplemetasikan penggunaan linked list.

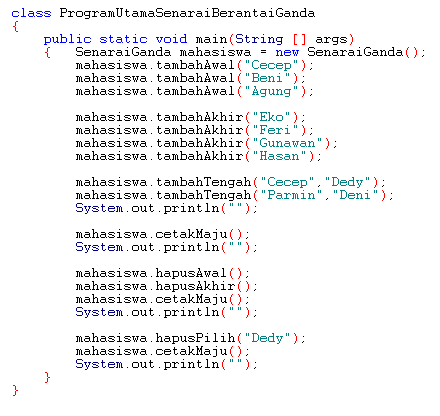
1. **PEMBAHASAN PRAKTIK**











Pembahasan

class Simpul

{ public

String data;

Simpul kiri;

Simpul kanan;  
//yaitu untuk mendekrasikan class simpul dan pada bagian, data,kiri dan kanan.

public

//----------Konstruktor----------

Simpul(String d)

{ data = d;

}

//yaitu untuk konstruktor yaitu mendekrasikan simpul data menjadi d.

//----------Fungsi untuk mencetak 'simpul----------

void cetakSimpul()

{ System.out.print(data);

}

}

//yaitu untuk mencetak simpul datanya tersebut.

class SenaraiGanda

{ private

Simpul awal;

Simpul akhir;

public

//----------Konstruktor----------

SenaraiGanda()

{awal = null;

akhir = null;

}  
//yaitu untuk senariGanda pada Simpul awal dan akhir maka nilai awal dengan akhir null atau kosong.

void tambahAwal(String dd)

{ System.out.println("menambah depan "+dd+" dilakukan....");

Simpul baru = new Simpul(dd);

if (awal == null)

akhir = baru;

else

awal.kiri = baru;

baru.kanan = awal;

baru.kiri = null;

awal = baru;

}  
// yaitu untuk method tambah awal ,maka jika data awal==null dan akhir =baru. Jika tidak maka data awal kiri=baru ,baru kanan=awal, baru.kiri=null ,awal =baru.

void tambahAkhir(String dd)

{ System.out.println("menambah belakang "+dd+" dilakukan....");

Simpul baru = new Simpul(dd);

if (awal==null)

awal = baru;

else

akhir.kanan=baru;

baru.kiri = akhir;

baru.kanan = null;

akhir = baru;

}

//yaitu bagian method akhir maka data menambah belakang, jika data awal==null maka awal= baru jika tidak maka maka akhir.kanan=baru, lalu baru.kiri = akhir maka data kanan null data akhir maka menjadi data baru.

Boolean tambahTengah(String kunci, String dd)

{ System.out.println("menambah tengah "+dd+" dilakukan....");

Simpul bantu = awal;

while (bantu.data != kunci)

{ bantu = bantu.kanan;

if (bantu == null)

{ System.out.println("--> Nambah "+dd+" Gagal, karena data "+kunci+" tidak ditemukan");

return false;

}

}

//yaitu method untuk menambah data tengah maka harus menggunakan data kunci jika ada data kunci maka maka bantu=bantu.kanan,jika bantu=null maka keluaran gagal karena data kunci tidak di temukan.

Simpul baru = new Simpul(dd);

if (bantu == akhir)

{ baru.kanan = null;

akhir = baru;

}

else

{

baru.kanan = bantu.kanan;

bantu.kanan.kiri = baru;

}

baru.kiri = bantu;

bantu.kanan = baru;

return true;

}

//yaitu pada method simpul baru maka jika bantu == akhir maka baru.kanan= null,akhir=baru, jika tidak maka baru.kanan= bantu.kanan,bantu.kanan.kiri=baru.

Simpul hapusAwal()

{ System.out.println("menhapus diAWAL dilakukan....");

Simpul temp = awal;

if (awal.kanan == null)

akhir = null;

else

awal.kanan.kiri = null;

awal = awal.kanan;

return temp;

}

//yaitu method pada simpul hapus awal maka jika awal.kanan==null maka akhir null jika tidak maka awal.kanan.kiri=null dan awal=awal.kanan.

Simpul hapusAkhir()

{ System.out.println("menghapus diAKHIR dilakukan....");

Simpul temp = akhir;

if (awal.kanan == null)

awal = null;

else

akhir.kiri.kanan = null;

akhir = akhir.kiri;

return temp;

}

//yaitu simpul hapusAkhir maka jika simpul awal .kanan==null,maka awal null jika tidak maka akhir.kiri.kanan=null maka akhir=akhir.kiri

Simpul hapusPilih(String kunci)

{ System.out.println("menghapus simpul dengan PILIHAN dilakukan....");

Simpul bantu = awal;

while (bantu.data != kunci)

{ bantu = bantu.kanan;

if (bantu == null)

return null;

}

if (bantu == awal)

awal = bantu.kanan;

//yaitu untuk menghapus dengan pilihan yang di lakukan maka maka perulangan jika dat tidak=kunci maka bantu =.bantu.kanan jika bantu==null ,maka jika bantu== awal, maka awal=bantu.kanan

else

{ if (bantu == akhir)

akhir = bantu.kiri;

else

{ bantu.kiri.kanan = bantu.kanan;

bantu.kanan.kiri = bantu.kiri;

}

}

return bantu;

}

//yaitu maka jika tidak,jika bantu ==akhir ,akhir =bantu.kiri jika tidak maka bantu.kiri.kanan= bantu.kanan,bantu kanan.kiri maka=bantu.kiri.

void cetakMaju()

{ System.out.print("Mencetak Maju: ");

Simpul bantu = awal;

while (bantu != null)

{ bantu.cetakSimpul();

bantu = bantu.kanan;

System.out.print(", ");

}

System.out.println(" ");

}

}

//yaitu jika Meencetak Maju, perulangan maka pada tidak nilai awal=null maka bantu.cetakSimpul bantu=bantu.kanan.

class ProgramUtamaSenaraiBerantaiGanda

{

public static void main(String [] args)

{ SenaraiGanda mahasiswa = new SenaraiGanda();

mahasiswa.tambahAwal("Cecep");

mahasiswa.tambahAwal("Beni");

mahasiswa.tambahAwal("Agung");

//yaitu untuk data awal maka cecep,beni dan agung.

mahasiswa.tambahAkhir("Eko");

mahasiswa.tambahAkhir("Feri");

mahasiswa.tambahAkhir("Gunawan");

mahasiswa.tambahAkhir("Hasan");

//yaitu pada tambah pada akhir maka data eko,feri,gunawan,hasan.

mahasiswa.tambahTengah("Cecep","Dedy");

mahasiswa.tambahTengah("Parmin","Deni");

System.out.println("");

//yaitu untuk pada tambah tengah maka cecep,dedy dan parmin,deni

mahasiswa.cetakMaju();

System.out.println("");

mahasiswa.hapusAwal();

mahasiswa.hapusAkhir();

mahasiswa.cetakMaju();

System.out.println("");

mahasiswa.hapusPilih("Dedy");

mahasiswa.cetakMaju();

System.out.println("");

}

}

//yaitu untuk pada class ProgramUtamaSenaraiBerantaiGanda maka bersifat public maka pada dedy menghapus datanya. Maka melakukan cetakMaju.

KESIMPULAN

Kita dapat menyimpulkan bahwa double linked yaitu pengalokasian memori secara dinamis di gunakan menyimpan data,maka double linked list maka lebih flexibel di banding linked list karena pada sata double linked list semua simpul-simpulnya yang di dalamnya memiliki petunjuk yang di gunakan untuk mengkaitkan diri dengan simpul lain yang ada pada sebelah kanan dan sebelah kirinya.