**LAPORAN**

**PRAKTIKUM STRUKTUR DATA**

**Pertemuan ke – 8**

****

Disusun Oleh :

Rendra Eka Herlambang

175410028

**LABORATORIUM TERPADU**

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

**2018**

**SENARI LINKEDLIST**

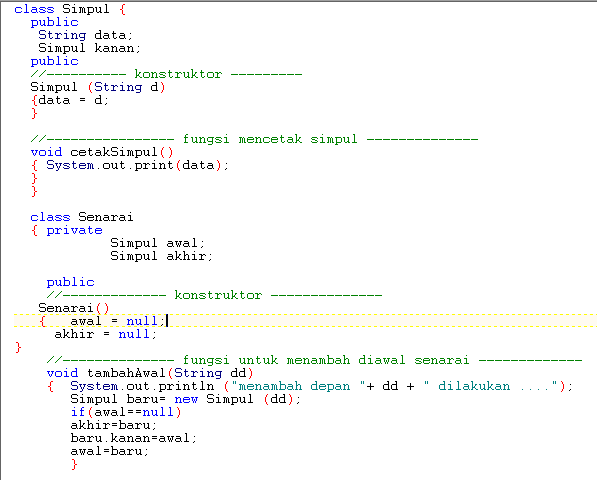
1. **TUJUAN**

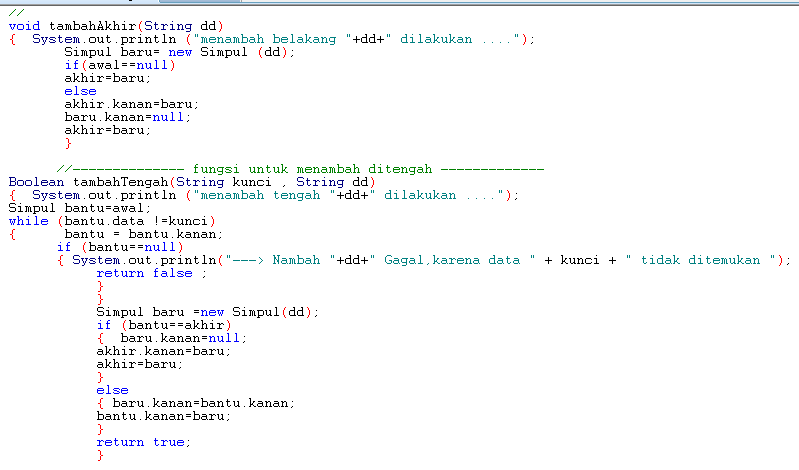
-Mahasiswa dapat membuat program dengan single linked list

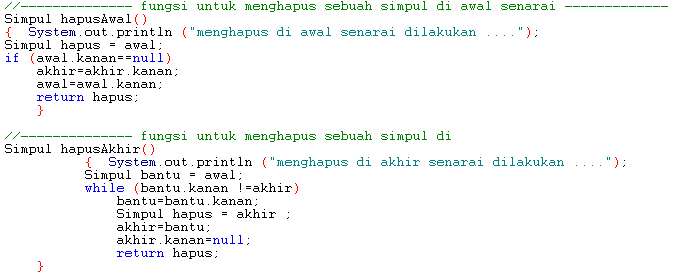
- Mahasiswa mampu mengimplementasikan senari beratai.

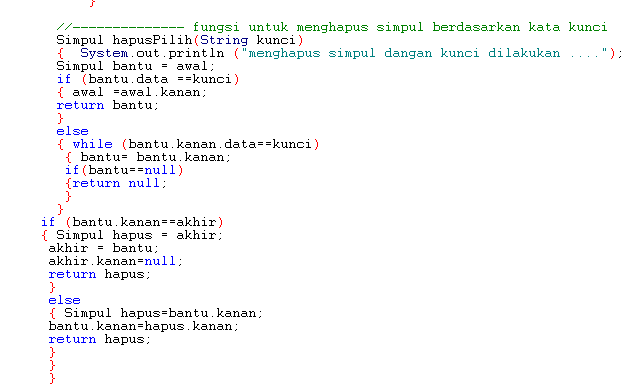
- Mahasiswa mampu mengimplemetasikan bentuk data dengan senari berantai.

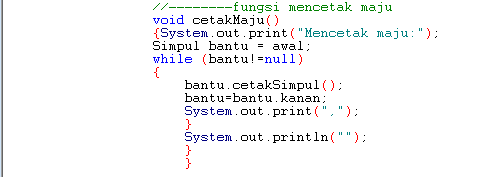
1. **PEMBAHASAN PRAKTIK**

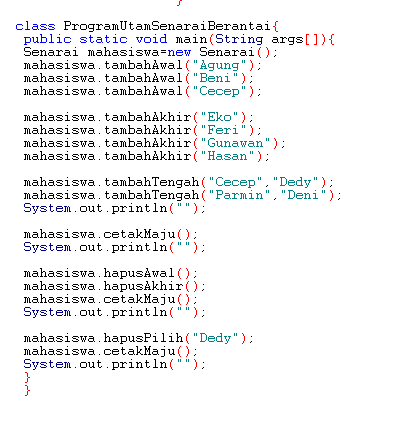


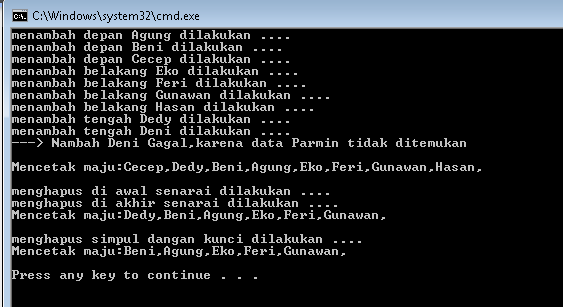












Pembahasan

class Simpul {

public

String data;

Simpul kanan;

//yaitu nama class simpul dengan data string dan simpul kanan.

public

//---------- konstruktor ---------

Simpul (String d)

{data = d;

}

// yaitu untuk konstruktor untuk mendekrasikan, simpul (String) data d

//---------------- fungsi mencetak simpul --------------

void cetakSimpul()

{ System.out.print(data);

}

}

//yaitu untuk keluaran atau outputan data.

class Senarai

{ private

Simpul awal;

Simpul akhir;

//yaitu untuk mendekrasikan class senari dengan simpul awal dan simpul akhir.

public

//------------- konstruktor --------------

Senarai()

{ awal = null;

akhir = null;

}

// yaitu menambah senari data awal dan akhir di beri null

//-------------- fungsi untuk menambah diawal senarai -------------

void tambahAwal(String dd)

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

Simpul baru= new Simpul (dd);

if(awal==null)

akhir=baru;

baru.kanan=awal;

awal=baru;

}

//yaitu untuk menambah senari, maka simpul baru,jika awal==null) maka akhir sama baru dan jika awal dengan baru.

void tambahAkhir(String dd)

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

Simpul baru= new Simpul (dd);

if(awal==null)

akhir=baru;

else

akhir.kanan=baru;

baru.kanan=null;

akhir=baru;

}

//untuk menambah belakang maka jika simpul awal==null maka akhir=baru, jika tidak maka akhir kanan barum dan akhir baru.

//-------------- fungsi untuk menambah ditengah -------------

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 ");nambah gagal karena data di temukan.

return false ;

}

//yaitu untuk menambah nilai tengah yaitu perulangan maka bantu kanan sebagai kunci, jika bantu==null maka hasil keluran

}

Simpul baru =new Simpul(dd);

if (bantu==akhir)

{ baru.kanan=null;

akhir.kanan=baru;

akhir=baru;

}

else

{ baru.kanan=bantu.kanan;

bantu.kanan=baru;

}

return true;akhir

}

//yaitu simpul baru, jka bantu==akhir maka baru kanan null maka akhir=baru jika tidak maka baru kanan= bantu kanan, maka bantu kanan =baru dan returun akhir.

//-------------- fungsi untuk menghapus sebuah simpul di awal senarai -------------

Simpul hapusAwal()

{ System.out.println ("menghapus di awal senarai dilakukan ....");

Simpul hapus = awal;

if (awal.kanan==null)

akhir=akhir.kanan;

awal=awal.kanan;

return hapus;

}

// yaitu untuk membuat simpul hapus maka jika.kanan==null maka akhir=akhir.kanan dan awal =awal.kanan.

//-------------- fungsi untuk menghapus sebuah simpul di

Simpul hapusAkhir()

{ System.out.println ("menghapus di akhir senarai dilakukan ....");

Simpul bantu = awal;

while (bantu.kanan !=akhir)

bantu=bantu.kanan;

Simpul hapus = akhir ;

akhir=bantu;

akhir.kanan=null;

return hapus;

}

//yaitu untuk bagian simpul akhir maka melakukan perulangan, bantu.kanan !=akhir maka, bantu=bantu kanan, simpul hapus=akhir maka akhir kanan, null maka hapus.

//-------------- fungsi untuk menghapus simpul berdasarkan kata kunci

Simpul hapusPilih(String kunci)

{ System.out.println ("menghapus simpul dangan kunci dilakukan ....");

Simpul bantu = awal;

if (bantu.data ==kunci)

{ awal =awal.kanan;

return bantu;

}

//yaitu menghapus berdasarkan kunci maka jika bantu.data==kunci maka awal=awal.kanan return bantu.

else

{ while (bantu.kanan.data==kunci)

{ bantu= bantu.kanan;

if(bantu==null)

{return null;

}

//maka jika tidak maka perulangan bantu.kanan.data==kunci, jika bantu=null.

}

if (bantu.kanan==akhir)

{ Simpul hapus = akhir;

akhir = bantu;

akhir.kanan=null;

return hapus;

}

// yaitu jika bantu.kanan==akhir maka simpul hapus=akhir maka akhir.kanan=null.

else

{ Simpul hapus=bantu.kanan;

bantu.kanan=hapus.kanan;

return hapus;

}

}

}

//--------fungsi mencetak maju

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 mencetak fungsi maju simpul awal maka perulangan bantu!==null maka akan centak simpul dan bantu=bantu.kanan.

class ProgramUtamSenaraiBerantai{

public static void main(String args[]){

Senarai mahasiswa=new Senarai();

mahasiswa.tambahAwal("Agung");

mahasiswa.tambahAwal("Beni");

mahasiswa.tambahAwal("Cecep");

//yaitu data yang di masukan pada bagian awal.

mahasiswa.tambahAkhir("Eko");

mahasiswa.tambahAkhir("Feri");

mahasiswa.tambahAkhir("Gunawan");

mahasiswa.tambahAkhir("Hasan");

//yaitu data yang di masukan pada bagian akhir.

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

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

//yaitu data yang di masukan bagian tengah.

System.out.println("");

mahasiswa.cetakMaju();

System.out.println("");

//yaitu untuk mencetak maju

mahasiswa.hapusAwal();

mahasiswa.hapusAkhir();

mahasiswa.cetakMaju();

System.out.println("");

//yaitu untuk menghapus awal,akhir

mahasiswa.hapusPilih("Dedy");

mahasiswa.cetakMaju();

System.out.println("");

//yaitu maka dendy kata kuncinya maka akan mencetak maju.

}

}

KESIMPULAN

Kita dapat menyimpulkan bahwa linked list adalah pengalokasia memori secara dinamis yang di gunakan untuk menyimpan data. Maka jika pada array kita dapat menyimpan data hanya sebanyak yang telah di defisisikan maka linked list menyimpan data seberapa pun kita mau.