**TUGAS PEKAN 2 : MEMBUAT ADT SINGLE LINKED LIST PART 1**

Buatlah ADT SINGLE LINKED LIST yang akan menyimpan data sesuai dengan data yang Tim Anda dapatkan

1. Lakukan MOD 5 terhadap digit terkahir pada NIM ketua Tim Anda.
2. Sisa nya menentukan jenis data apa yang Anda simpan di list

* Sisa 0 : Data yang berhubungan dengan data terkait sekolah/kampus
* Sisa 1 : Data yang berhubungan dengan data terkait penjualan barang
* Sisa 2 : Data yang berhubungan dengan data terkait kesehatan
* Sisa 3 : Data yang berhubungan dengan data terkait pertandingan
* Sisa 4: Data yang berhubungan dengan data terkait pekerjaan

1. Data tersebut minimal memiliki 3 sub data. Contoh : data terkait sekolah/ kampus, misal data mahasiswa (tidak boleh digunakan karena dijadikan contoh). Data mahasiswa terdiri dari NIM, NAMA, IPK

**LENGKAPILAH ADT SINGLE LINKED LIST BERIKUT INI BESERTA MAIN PROGRAMNYA**

1. Isi SLL.h
   1. Deklarasikan List Anda disini

Type karyawan : <

Sesuaikan dengan jenis data yang Anda dapatkan sesuai panduan di atas

nama, npwp, jabatan : string

gaji : integer >

Type infotype : karyawan

Type adr : pointer to elmList

Type elmList : <

info : infotype

next : adr >

Type List : < first : adr>

* 1. Tuliskan primitive single linked list based on fungsi procedure yang ada pada file sll.cpp pada poin B

**Procedure** Create\_List (**In/Out** L : List)

**Function** New\_Elemen (data : infotype)🡪 adr

**Procedure** Insert\_First (**In/Out** L : List, **In** p : adr)

**Procedure** Insert\_Last (**In/Out** L : List, **In** p : adr)

**Procedure** Insert\_After (**In/Out** L : List, **In** prec, p : adr)

**Procedure** Delete\_First (**In/Out** L : List, **Out** p : adr)

**Procedure** Delete\_Last (**In/Out** L : List, **Out** p : adr)

**Procedure** Delete\_After (**In/Out** L : List, **In** prec : adr,**Out** p : adr)

**Procedure** Show (**In** L : List)

1. Isi SLL.cpp
   1. **Procedure** Create\_List (**In/Out** L : List)

*{I.S. –*

*F.S. Pointer First dari List L bernilai NIL}*

**Kamus data**

………….

**Algoritma** L.first = NIL

**Endprocedure**

* 1. **Function** New\_Elemen (data : infotype)🡪 adr

*{Function akan membuat elemen baru berisi data dan mengembalikan pointer yang menyimpan alamat dari elemen tersebut}*

**Kamus data**

P : adr

**Algoritma** allocate (P)

P 🡪 next = NIL

(P 🡪 info) = data

return P

**Endfunction**

* 1. **Procedure** Insert\_First (**In/Out** L : List, **In** P : adr)

*{I.S. Terdefinisi List L yang mungkin kosong dan sebuah elemen baru yang alamatnya disimpan oleh pointer P*

*F.S. Elemen baru telah masuk menjadi elemen paling pertama di list L}*

**Kamus data**

………….

**Algoritma**

If L.first == NIL then

L.first = P

else  P 🡪 next = L.first

L.first = P

endif

**Endprocedure**

* 1. **Procedure** Insert\_Last (**In/Out** L : List, **In** P : adr)

*{I.S. Terdefinisi List L yang mungkin kosong dan sebuah elemen baru yang alamatnya disimpan oleh pointer P*

*F.S. Elemen baru telah masuk menjadi elemen paling akhir di list L}*

**Kamus data**

Q : adr

**Algoritma**

if L.first != NIL then  Q = L.first

while Q 🡪 next != NIL do

Q = Q 🡪 next

endwhile

Q 🡪 next = P

else

L.first = P

Endif

**Endprocedure**

* 1. **Procedure** Insert\_After (**In/Out** L : List, **In** prec, p : adr)

*{I.S. Terdefinisi List L yang mungkin kosong, sebuah elemen baru yang alamatnya disimpan oleh pointer P dan pointer prec*

*F.S. Elemen baru telah masuk menjadi elemen setelah elemen yang alamatnya disimpan oleh pointer prec }*

**Kamus data**

………….

**Algoritma** P 🡪 next = prec 🡪 next

prec 🡪 next = P

**Endprocedure**

* 1. **Procedure** Delete\_First (**In/Out** L : List, **Out** P : adr)

*{I.S. Terdefinisi List L yang mungkin kosong atau Cuma memiliki 1 elemen.*

*F.S. Jika list kosong, maka pointer P di assign dengan NIL, jika tidak kosong maka elemen paling awal dihapus dari List L, alamatnya disimpan oleh pointer p}*

**Kamus data**

………….

**Algoritma**

P = L.first

L.first = P 🡪 next

P 🡪 next = NIL

**Endprocedure**

* 1. **Procedure** Delete\_Last (**In/Out** L : List, **Out** p : adr)

*{I.S. Terdefinisi List L yang mungkin kosong atau Cuma memiliki 1 elemen.*

*F.S. Jika list kosong, maka pointer P di assign dengan NIL, jika tidak kosong maka elemen paling akhir dihapus dari List L, alamatnya disimpan oleh pointer p}*

**Kamus data**

Q, R : adr

**Algoritma** Q = L.first

while Q 🡪 next != NIL do

R = Q

Q = Q 🡪 next

endwhile

P = R 🡪 next

R🡪 next = NIL

**Endprocedure**

* 1. **Procedure** Delete\_After (**In/Out** L : List, **In** prec : adr,**Out** p : adr)

*{I.S. Terdefinisi List L yang mungkin kosong, sebuah elemen baru yang alamatnya disimpan oleh pointer P dan pointer prec*

*F.S. Elemen baru telah masuk menjadi elemen setelah elemen yang alamatnay disimpan oleh pointer prec }*

**Kamus data**

………….

**Algoritma** P = prec 🡪 next

prec 🡪 next = P 🡪 next

P 🡪 next = NIL

**Endprocedure**

* 1. **Procedure** Show (**In** L : List)

*{I.S. Terdefinisi List L yang mungkin kosong.*

*F.S. Jika list kosong maka tampilkan ke layar “list kosong”, jika tidak maka seluruh data pada list ditampilkan ke layar}*

**Kamus data**

P : adr

**Algoritma**

if L.first != NIL then

P = L.first

while P != NIL do

output(P 🡪 info.nama, P 🡪 info.npwp, P 🡪info.jabatan, P🡪info.gaji)

P = P 🡪 next

endwhile

else

output(“List Kosong”)

endif

endprocedure

1. Isi Main.cpp

*{Lengkapilah titik-titik berikut ini agar pada main program :*

1. *Deklarasi variable List*
2. *Mengcreate List*
3. *Memasukkan 4 data ke list dengan memanfaatkan dua jenis insert*
4. *Memanggil show data*
5. *Menghapus 2 data ke list dengan memanfaatkan dua jenis delete}*

*//deklarasi variable List.* ***Nama variable List adalah huruf pertama pada nama lengkap Anda dan huruf terakhir nama lengkap anggota tim Anda***

Kamus Lokal

IAI : List

data : infotype

P : adr

Algoritma

*//meng create list*

Create\_List(IAI)

*//meminta user* ***data pertama*** *yang akan dimasukkan ke list*

input(data.nama, data.nwp, data.jabatan, data.gaji)

*//melakukan create new elemen berdasarkan data yang diinputkan user*

P = New\_Elemen(data)

*//memanggil salah satu jenis insert (jangan insert after)*

Insert\_First(IAI, P)

*//meminta user* ***data kedua*** *yang akan dimasukkan ke list*

input(data.nama, data.nwp, data.jabatan, data.gaji)

*//melakukan create new elemen berdasarkan data yang diinputkan user*

P = New\_Elemen(data)

*//memanggil salah satu jenis insert (berbeda dengan insert yang data pertama, jangan Insert after)*

Insert\_Last(IAI, P)

*//meminta user* ***data ketiga*** *yang akan dimasukkan ke list*

input(data.nama, data.nwp, data.jabatan, data.gaji)

*//melakukan create new elemen berdasarkan data yang diinputkan user*

P = New\_Elemen(data)

*//memanggil salah satu jenis insert (insert First)*

Insert\_First(IAI, P)

*//meminta user* ***data keempat*** *yang akan dimasukkan ke list*

input(data.nama, data.nwp, data.jabatan, data.gaji)

*//melakukan create new elemen berdasarkan data yang diinputkan user*

P = New\_Elemen(data)

*//memanggil salah satu jenis insert (insert Last)*

Insert\_Last(IAI, P)

*//memanggil show*

Show(IAI)

*//melakukan penghapusan data dengan memanfaatkan salah satu jenis delete (bukan delete after)*

Delete\_First(IAI, P)

*//melakukan penghapusan data dengan memanfaatkan salah satu jenis delete (berbede dengan delete sebelumnya, dan bukan delete after)*

Delete\_Last(IAI, P)