

TUGAS PEKAN 2 APTSL

kelompok

- ①. Ihab Hasanain Akmal (103032330054)
- ②. Geni Dama (103032330095)
- ③. Salsabila Rahayu Putri (103032330153)

A. Isi SLL.h

a. Deklarasikan List Anda disini

Type karyawan : <

nama, npwp, jabatan : string

gaji : integer >

Type Infotype : karyawan

Type adr : pointer to elmList

Type elmList : <

Info : Infotype

next : adr >

Type List : < first : adr >

b. 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-Element (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)

c. Isi SLL.cpp

a. Procedure Create_List (in/out L : List)

{ 1.s. -

F.S. Pointer first dari List L bernilai NIL }

kamus data

Algoritma

L.First = NIL

Endprocedure



Ihab Hasanain Akmal (103032330054)
Geni Dama (103032330095)
Salsabila Rahayu Putri (103032330153)

b. function New-Element (data : Info type) \rightarrow 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 \rightarrow next = NIL

P \rightarrow info = data

return P

Endfunction

c. Procedure Insert-First (In/out L : List, In P : adr)

{ 1.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 \rightarrow next = L.first

L.first = P

endif

Endprocedure

d. Procedure Insert-Last (In/out L : List, In P : adr)

{ 1.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 \neq NIL then

Q = L.first

while Q \rightarrow next \neq NIL do

Q = Q \rightarrow next

endwhile




```

Q → next = P
else
  L.first = P
endif
Endprocedure

```

e. Procedure Insert-After (in/out L: List, in prec: adr, 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

f. 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 dari list L alamatnya disimpan oleh pointer P }

Kamus Data

Algoritma

if L.first ≠ NIL then

P = L.first

L.first = P → next

P → next = NIL

else

P = NIL

endif

Endprocedure



g. Procedure Delete-Last (in/out L: List, out P: adr)

{ 1-s Terdefinisi List L yang mungkin kosong atau cuma memiliki 1 elemen.

F-s Jika list kosong maka pointer P diassign dengan NIL, jika tidak kosong maka elemen paling akhir dihapus dari List L, alamatnya disimpan oleh pointer P

Kamus Data

Q, R : adr

Algoritma

```
if L.first != NIL then
  if L.first → next == NIL then
    P = L.first
    L.first = NIL
  else
    Q = L.first
    while Q → next != NIL do
      R = Q
      Q = Q → next
    endwhile
    P = R → next
    R → next = NIL
  endif
else
  P = NIL
endif
```

Endprocedure

h. Procedure Delete-After (in/out L: List, in prec: adr, out P: adr)

{ 1-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 = prec → next
prec → next = P → next
P → next = NIL
```

Endprocedure



ii. Procedure show (in L: list)

{ 1.s. Terdefinisi list L yang mungkin kosong

F.s. Jika list kosong maka tampilkan layar "list kosong".

Jika tidak maka seluruh data pada list ditampilkan layar

g

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

c. Isi Main.cpp

// deklarasi variabel list. Nama variabel 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.npwp, data.jabatan, data.gaji)

// melakukan create new elemen berdasarkan data yang diinput user

P = New-Element(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.npwp, data.jabatan, data.gaji)

// melakukan create new elemen berdasarkan data yang diinputkan user

P = New-Element(data)

// memanggil salah satu insert (berbeda dengan insert data pertama, jangan insert after)

Insert-Last(IAI, P)

// meminta user data ketiga yang akan dimasukkan ke list

Input (data.nama, data.npwp, data.jabatan, data.gaji)

// melakukan create new elemen berdasarkan data yang diinputkan user

P = New-Element(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.npwp, data.jabatan, data.gaji)

// melakukan create new elemen berdasarkan data yang diinputkan user

P = New-Element(data)

// memanggil salah satu 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 (berbeda dengan delete sebelumnya, dan bukan delete after)

Delete-last (IAI, P)