TUGAS STRUKTUR DATA DAN ALGORITMA

STUDI KASUS ADT LINKED LIST

Oleh :

ARSAL FADILAH

201524036

POLITEKNIK NEGERI BANDUNG

JURUSAN TEKNIK KOMPUTER

D-4 TEKNIK INFORMATIKA

2021

* Pseudocode

Struktur data :

typedef char infotype;

typedef struct tElmtlist \*address;

typedef struct tElmtlist {

infotype info;

address next;

}ElmtList;

typedef struct {

address First;

}List;

* + Create List

Deklarasi :

-

Algoritma :

Procedure CreateList(Input/Output List L)

Begin

L^.First = Nil

End

* + Insert Huruf Pertama, value ‘u’

Deklarasi :

Addres P

Algoritma :

Procedure InsertFirst(Input.Output List L, Infotype X)

Begin

P 🡸 alloc(1, address)

If P != Nil then

P^.Info 🡸 X

P^.Next 🡸 L.First

L.First 🡸 P

End If

End

* + Insert value ‘c’ setelah huruf pertama

Deklarasi :

Addres P, Last

Algoritma :

Procedure InsertLast(Input.Output List L, Infotype X)

Begin

P 🡸 alloc(1, address)

If P != Nil then

P^.Info 🡸 X

If L.First == Nil then

P^.Next 🡸 L.First

L.First 🡸 P

Else then

Last 🡸 L.First

While Last^.Next != Nil

Last 🡸 Last^.Next

End While

Last^.Next 🡸 P

End If

End If

End

* + Insert value ‘h’ setelah huruf kedua

Deklarasi :

Addres P, Last

Algoritma :

Procedure InsertLast(Input.Output List L, Infotype X)

Begin

P 🡸 alloc(1, address)

If P != Nil then

P^.Info 🡸 X

If L.First == Nil then

P^.Next 🡸 L.First

L.First 🡸 P

Else then

Last 🡸 L.First

While Last^.Next != Nil

Last 🡸 Last^.Next

End While

Last^.Next 🡸 P

End If

End If

End

* + Insert value ‘i’ setelah huruf ketiga

Deklarasi :

Addres P, Last

Algoritma :

Procedure InsertLast(Input.Output List L, Infotype X)

Begin

P 🡸 alloc(1, address)

If P != Nil then

P^.Info 🡸 X

If L.First == Nil then

P^.Next 🡸 L.First

L.First 🡸 P

Else then

Last 🡸 L.First

While Last^.Next != Nil

Last 🡸 Last^.Next

End While

Last^.Next 🡸 P

End If

End If

End

* + Insert value ‘n’ setelah huruf pertama

Deklarasi :

-

Algoritma :

Procedure InserAfter(Input.Output List L, address P, address Prec)

Begin

P^.Next 🡸 Prec^.Next

Prec^.Next 🡸 P

End

* + Delete value ‘i’ pada elemen list

Deklarasi :

Address Last, PrecLast

Algoritma :

Procedure DeleteLast(Input/Output List L, input/output address P)

Begin

Last 🡸 \*(L.First)

If Last^.Next == Nil then

Last 🡸 \*(L.First)

\*(L.First) 🡸 \*(L.First)^.Next

Last.Next 🡸 Nil

Else then

PrecLast 🡸 Nil

While Last^.Next != Nil then

PrecLast 🡸 Last

Last 🡸 Last^.Next

End While

\*P 🡸 Last

PrecLast^.Next 🡸 Nil

End If

End

* + Insert value ‘p’ di awal list Insert value ‘a’ setelah huruf ‘h’

Deklarasi :

-

Algoritma :

Procedure InserAfter(Input.Output List L, address P, address Prec)

Begin

P^.Next 🡸 Prec^.Next

Prec^.Next 🡸 P

End

* + Show semua elemen karakter

Deklarasi :

Addres P

Algoritma :

Procedure PrintInfo(Input List L)

Begin

P 🡸 L^.First

If P == Nil then

Write (layar) “List Kosong”

Else

Do then

Write (layar) P^.info

P = P^.Next

While P != Nil

End If

End

* + Cari huruf ‘h’ pada list, dan tampilkan posisi elemen tersebut pada list. Jika ada tampilkan posisinya, jika tidak ada tampilkan “tidak ada”

Deklarasi :

Int idx 🡸 1

Algoritma :

Function IdxNList(Input List L, infotype x) : int

Begin

While L.First != Nil And L.First^.Info != x then

Idx++

L.First 🡸 L.First^.Next

End while

If L.First^.info != x then

Return 0

Else

Return idx

End if

End

* + Delete semua elemen list karakter

Deklarasi :

Address X

Algoritma :

Procedure DelAll(Input/Output List L)

Begin

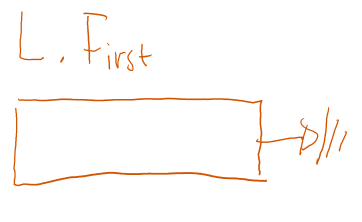
While L.First != Nil then

DeleteLast(L, X)

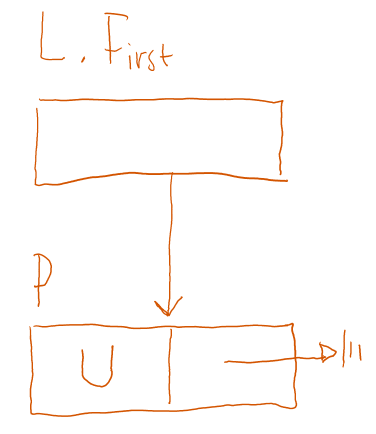
End while

End

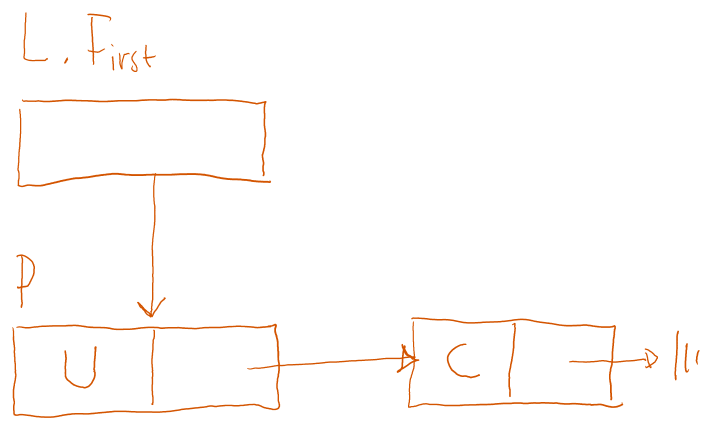
* Representasi/Ilustrasi Gambar Linked List
  + Create List



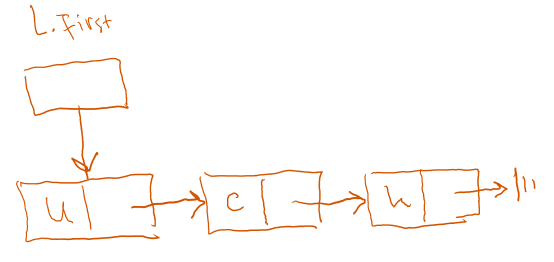
* + Insert Huruf Pertama, value ‘u’



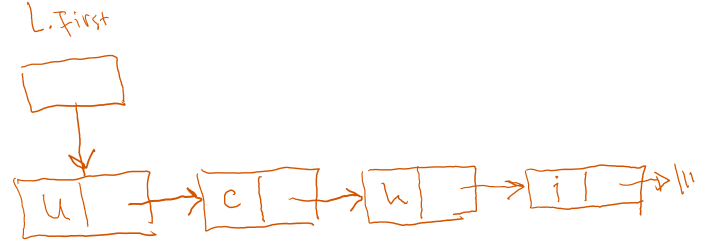
* + Insert value ‘c’ setelah huruf pertama



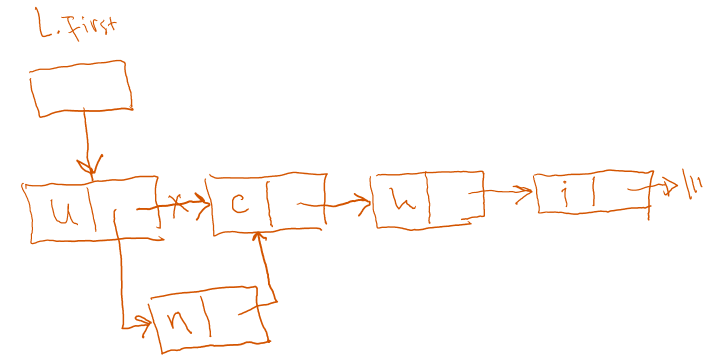
* + Insert value ‘h’ setelah huruf kedua



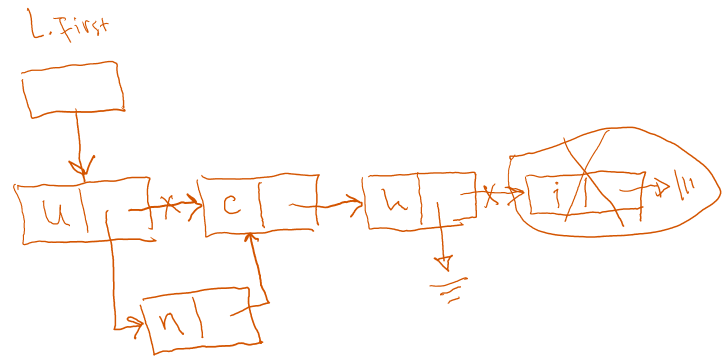
* + Insert value ‘i’ setelah huruf ketiga



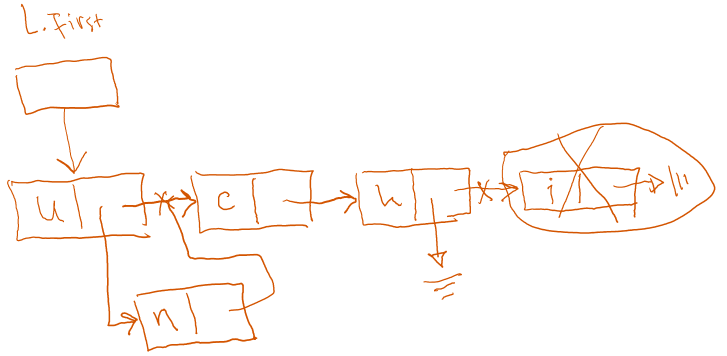
* + Insert value ‘n’ setelah huruf pertama



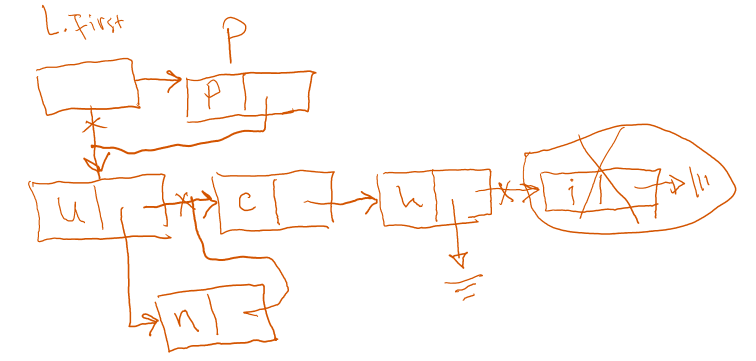
* + Delete value ‘i’ pada elemen list



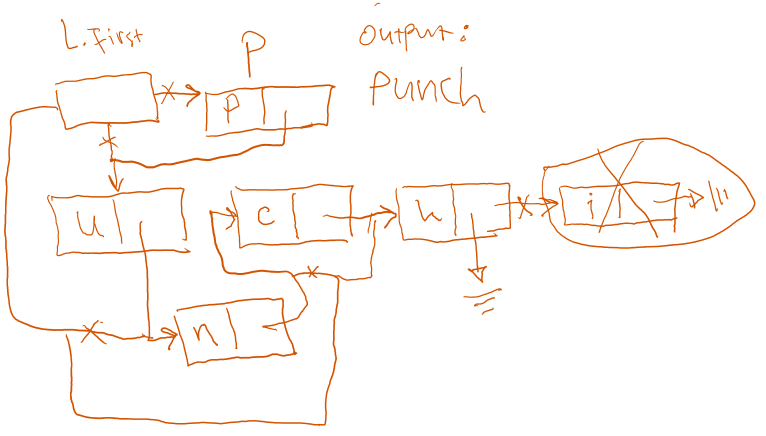
* + Insert value ‘p’ di awal list



* + Insert value ‘a’ setelah huruf ‘h’

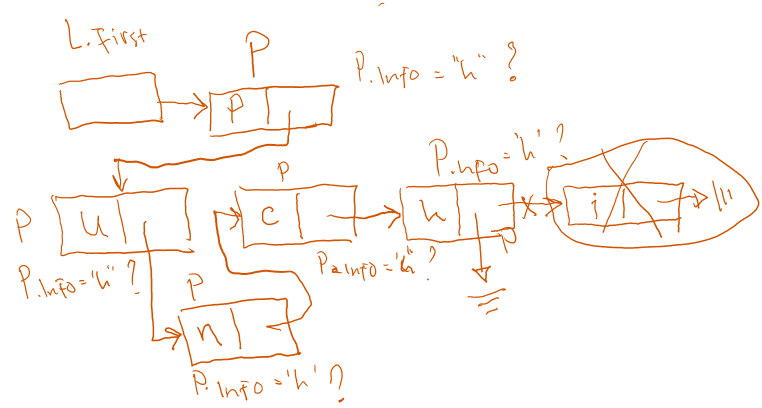


* + Show semua elemen karakter

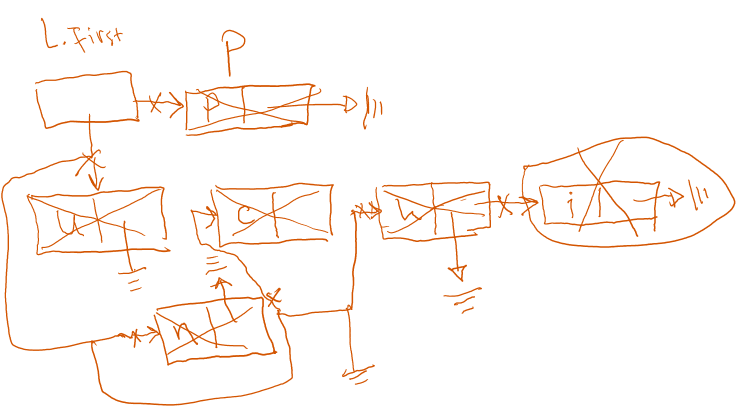


Cat : L.First sebagai parameter input saja

* + Cari huruf ‘h’ pada list, dan tampilkan posisi elemen tersebut pada list. Jika ada tampilkan posisinya, jika tidak ada tampilkan “tidak ada”



* + Delete semua elemen list karakter



//Hasil Running

