

[DS-47-O2] Jurnal Modul 6
STRUKTUR DATA – Ganjil 2024/2025
" DLL "

A. Soal Jurnal PART 1

Deskripsi Soal:

- Buatlah program yang mengizinkan pengguna memasukkan beberapa elemen **integer** ke dalam **Doubly Linked List**.
- Pengguna dapat **menghapus elemen tertentu** dari DLL berdasarkan nilai yang dimasukkan pengguna.
- Buatlah ADT nya (DLL.h, DLL.cpp, dan Main.cpp)!

Instruksi:

1. Implementasikan prosedur `deleteByValue` yang menghapus elemen dengan nilai tertentu.
2. Jika nilai yang ingin dihapus tidak ada dalam list, tampilkan pesan bahwa elemen tidak ditemukan.
3. Tampilkan elemen dalam list setelah penghapusan dilakukan.

Contoh Input:

- Input: 5, 10, 15, 20
- Hapus elemen: 15

Contoh Output:

- Output: 5 – 10 – 20

Langkah-langkah penyelesaian:

1. Spesifikasi (Silahkan ditulis ulang dalam Bahasa C++)

```
Type infotype : int
Type address : pointer to elmList
Type elmList <
    info : infotype
    next : address
    prev : addrees
>

Type List <
    First : address
    Last : address
>

Procedure createList (input/output L : List)
Function createElemen (dataBaru: infotype) → address
Procedure insertLast (input/output L : List, input P : address)
Procedure InsertAfter (input/output Prec : address, P :
address);
Procedure deleteFirst (input/output L : List, output P : address)
Procedure deleteLast (input/output L : List, output P : address)
Procedure deleteAfter (input/output Prec: address, output P :
address)
Procedure printList(input L : List)
Procedure deleteByValue(input/output L : List, input value: int)
```

2. Implementasi (Silahkan ditulis ulang dalam Bahasa C++)

```
Procedure createList (input/output L : List)
{IS. –
    FS. Terbentuk sebuah list di mana, first dan last dari L bernilai NIL. }

Kamus
Algoritma
    First (L) ← NIL
    Last (L) ← NIL
```

Function createElemen (dataBaru: infotype) \rightarrow address

{Return alamat alokasi memori sebuah elmList yang berisi dataBaru. }

Kamus

P: address

Algoritma

alokasi (P)

info (P) \leftarrow dataBaru

next (P) \leftarrow NIL

prev (P) \leftarrow NIL

return P

Procedure insertLast (input/ouput L : List, input P : address)

{IS. Terdefinisi pointer P berisi alamat elmList, dan sebuah list L (L mungkin kosong).

FS. elmList yang ditunjuk oleh P ditambahkan ke dalam list sebagai elemen terakhir. }

Kamus

Algoritma

if First (L) = NIL then

 First (L) \leftarrow P

 Last (L) \leftarrow P

else

 (Last(L)) \leftarrow P

 prev (P) \leftarrow Last(L)

 Last (L) \leftarrow P

Procedure InsertAfter (input/output Prec : address, P : address);

{IS. Terdefinisi pointer Prec dan P berisi alamat elmList. Prec \neq Last(L).

FS. elmList yang ditunjuk oleh P ditambahkan ke dalam list setelah elmList yang ditunjuk oleh Prec. }

Kamus

Algoritma

next (P) \leftarrow next (Prec)

Procedure deleteFirst (input/ouput L : List, output P : address)

{IS. Terdefinisi sebuah list L (L tidak kosong dan mungkin berisi satu elemen).

FS. P berisi alamat elmList yang pertama, elmList yang ditunjuk oleh P dihapus dari list}

Kamus

Algoritma

P \leftarrow First (L)

if next (First (L)) = NIL then

 First (L) \leftarrow NIL

 Last (L) \leftarrow NIL

Else

 First (L) \leftarrow next (First (L))

 prev (First(L)) \leftarrow NIL

 next (P) \leftarrow NIL

//DEFINISIKAN PROCEDURE DELETELAST!

Procedure deleteAfter (**input/output** Prec: address, **output** P : address)

{IS. Terdefinisi pointer Prec berisi alamat elmList. Prec \neq Last(L). next(Prec) \neq Last(L).}

FS. P berisi alamat elmList setelah Prec, elmList yang ditunjuk oleh P dihapus dari list }

Kamus

Algoritma

P \leftarrow next (Prec)
next(Prec) \leftarrow next (P)
prev (next(P)) \leftarrow prec
next (P) \leftarrow NIL
prev (P) \leftarrow NIL

Procedure printList (**input** L: list)

{IS. Terdefinisi sebuah List L}

FS. Menampilkan semua info elmList di list. }

Kamus

Algoritma

P \leftarrow First (L)
while P \neq NIL do
 output (info (P))
 P \leftarrow next (P)

Procedure deleteByValue (**input/output** L: List, **input** value: int)

{IS. Terdefinisi sebuah list L}

FS. Elemen pada L yang memiliki nilai sama dengan value akan terhapus}

3. Buat main programnya sehingga memenuhi contoh input yang telah diberikan

B. SOAL JURNAL PART 2

Deskripsi Soal:

- Pada DLL yang telah didefinisikan, implementasikan fungsi **`reverseList`** yang membalik urutan elemen dalam list, sehingga elemen pertama menjadi terakhir dan elemen terakhir menjadi yang pertama.

Instruksi:

1. Implementasikan fungsi **`reverseList`** yang membalik urutan elemen-elemen dalam list.
2. Setelah dibalik, tampilkan list dari depan ke belakang untuk memverifikasi hasil.

Contoh Input:

- Input: 1, 2, 3, 4

Contoh Output:

- Output: 4 – 3 – 2 – 1

Langkah-langkah penyelesaian:

1. Buat implementasi procedure berikut ini (**Implementasikan juga pada header**):

Procedure reverseList (**input/ouput** L : List)

{IS. Terdefinisi List L

FS. Urutan elemen pada List L dibalik urutannya}

2. Buat main programnya sehingga memenuhi contoh input yang telah diberikan