

# **TUGAS 4**

## **STRUKTUR DATA**



**Oleh:**

Martogi Jekson C. Siagian 2105541054

**PROGRAM STUDI TEKNIK ELEKTRO**

**FAKULTAS TEKNIK**

**UNIVERSITAS UDAYANA**

**2023**

## TUGAS 4

### Tugas

Pahamilah contoh program (source code) yang telah disediakan untuk konsep Circular Single Linked List dan Double Linked List. Selanjutnya tambahkanlah method/fungsi pada contoh program tersebut sebagai berikut.

- Pada Circular Single Linked List, buatlah sebuah method untuk melakukan penghapusan node sesuai posisi tertentu, posisi ini menjadi parameter input pada method/fungsi yang dibuat tersebut.
- Pada Double Linked List, buatlah sebuah method untuk melakukan penyisipan node sesuai posisi yang diinginkan, posisi ini menjadi parameter input pada method/fungsi yang dibuat tersebut.

Jawab:

#### 1. Circular Single Linked list

Berikut method yang saya tambahkan untuk melakukan penghapusan node

```
121 void deleteAtPosition(int posisi){
122     if(head == NULL){
123         cout<<"Linked List Kosong!";
124     }else{
125         if (posisi == 1) {
126             delFirst(); // Jika posisi yang dihapus adalah posisi pertama, panggil fungsi deleteFirst()
127             return;
128         }
129
130         cur = head;
131         Buku *prev = NULL;
132         int nomor = 1;
133
134         while(nomor < posisi && cur->next != head){
135             prev = cur;
136             cur = cur->next;
137             nomor++;
138         }
139
140         if(nomor != posisi){
141             cout << "Posisi tidak valid!" << endl;
142             return;
143         }
144
145         prev->next = cur->next;
146
147         if(cur == tail){
148             tail = prev; // Jika node yang dihapus adalah tail, perbarui tail
149         }
150
151         delete cur;
152     }
153 }
```

```
187 deleteAtPosition(2); // Menghapus node pada posisi ke-3
188 printCircular();
189 cout<<"\n\n"<<endl;
```

Berikut Penjelasannya:

- Fungsi deleteAtPosition menerima satu parameter, yaitu posisi, yang menunjukkan posisi node yang akan dihapus dari linked list.
- Fungsi ini pertama-tama memeriksa apakah linked list kosong. Jika kosong, maka akan mencetak pesan bahwa linked list kosong.

- Jika posisi yang dimasukkan adalah 1 (posisi pertama), maka akan memanggil fungsi delFirst() untuk menghapus node pertama dari linked list.
- Jika posisi bukan 1, fungsi akan mencari node yang akan dihapus. Untuk itu, variabel cur dan prev digunakan untuk menelusuri linked list.
- Fungsi ini akan berjalan hingga menemukan node pada posisi yang diinginkan atau mencapai akhir linked list.
- Ketika node pada posisi yang diinginkan ditemukan, fungsi akan mengubah pointer next pada node sebelumnya agar menunjuk ke node setelah node yang akan dihapus, dan kemudian menghapus node yang ditentukan.
- Jika node yang dihapus adalah tail (node terakhir), fungsi ini juga akan memperbarui pointer tail ke node sebelumnya.
- Jika posisi yang dimasukkan tidak valid (misalnya, lebih besar dari jumlah node yang ada dalam linked list), fungsi akan mencetak pesan bahwa posisi tidak valid.

Dalam Methode ini saya mencoba untuk menghapus node ke 2 berikut hasilnya

```
Judul Buku : Sistem Informasi
Pengarang Buku : Jogiyanto
Tahun Terbit : 2010
Judul Buku : XXX
Pengarang Buku : abc
Tahun Terbit : 2010
Judul Buku : Struktur Data 1
Pengarang Buku : Abdul Kadir
Tahun Terbit : 2008
```

```
Judul Buku : Sistem Informasi
Pengarang Buku : Jogiyanto
Tahun Terbit : 2010
Judul Buku : Struktur Data 1
Pengarang Buku : Abdul Kadir
Tahun Terbit : 2008
```

## 2. Double linked list 1

Berikut Methode yang saya tambahkan untuk melakukan penyisipan node sesuai posisi yang diinginkan

```
91 void insertAtPosition(string judul, string pengarang, int thnTerbit, int position){
92     if(position <= 0){
93         cout << "Posisi harus lebih besar dari 0." << endl;
94         return;
95     }
96
97     newNode = new Buku();
98     newNode->judul = judul;
99     newNode->pengarang = pengarang;
100    newNode->tahunTerbit = thnTerbit;
101
102    if(head == NULL){
103        // Jika linked list kosong, node baru akan menjadi head dan tail
104        newNode->prev = NULL;
105        newNode->next = NULL;
106        head = newNode;
107        tail = newNode;
108        return;
109    }
110
111    if(position == 1){
112        // Jika posisi yang diinginkan adalah di awal linked list
113        newNode->prev = NULL;
114        newNode->next = head;
115        head->prev = newNode;
116        head = newNode;
117        return;
118    }
119
120    cur = head;
121    int count = 1;
122
123    while(count < position - 1 && cur != NULL){
124        cur = cur->next;
125        count++;
126    }
127
128    if(cur == NULL){
129        // Jika posisi melebihi jumlah node yang ada
130        cout << "Posisi melebihi panjang linked list. Node akan ditambahkan di akhir." << endl;
131        newNode->prev = tail;
132        newNode->next = NULL;
133        tail->next = newNode;
134        tail = newNode;
135        return;
136    }
137
138    newNode->prev = cur;
139    newNode->next = cur->next;
140    if(cur->next != NULL){
141        cur->next->prev = newNode;
142    }
143    cur->next = newNode;
144 }
145
161 insertAtPosition("Algoritma Pemrograman", "Martogi J.S", 2021, 2); // Contoh penyisipan di posisi kedua
162 printDoubleLS();
163 cout << "\n\n" << endl;
```

Berikut Penjelasannya:

- Input: Menerima informasi tentang buku baru yang ingin disisipkan (judul, pengarang, tahun terbit), serta posisi di mana buku tersebut akan dimasukkan ke dalam linked list.
- Memeriksa apakah linked list kosong. Jika ya, node baru akan menjadi head dan tail.
- Memeriksa apakah posisi yang diminta adalah di awal linked list.
- Melakukan iterasi melalui linked list hingga mencapai posisi yang diinginkan.

- Menyisipkan node baru ke dalam linked list pada posisi yang ditentukan, dengan mengatur pointer prev dan next agar terhubung dengan node sebelumnya dan setelahnya.

Pada methode ini saya mencoba menyisipkan node baru pada posisi kedua dan berikut hasilnya:

```
Judul Buku : Sistem Informasi
Pengarang Buku : Jogiyanto
Tahun Terbit : 2010
Judul Buku : Struktur Data 1
Pengarang Buku : Abdul Kadir
Tahun Terbit : 2008
Judul Buku : Basis Data
Pengarang Buku : Jogiyanto
Tahun Terbit : 2012

Judul Buku : Sistem Informasi
Pengarang Buku : Jogiyanto
Tahun Terbit : 2010
Judul Buku : Algoritma Pemrograman
Pengarang Buku : Martogi J.S
Tahun Terbit : 2021
Judul Buku : Struktur Data 1
Pengarang Buku : Abdul Kadir
Tahun Terbit : 2008
Judul Buku : Basis Data
Pengarang Buku : Jogiyanto
Tahun Terbit : 2012
```

### 3. Double linked list 2

Berikut Methode yang saya tambahkan untuk melakukan penyisipan node sesuai posisi yang diinginkan

```
11 void sisipNode(Buku* &head, string judul, string pengarang, int tahunTerbit, int posisi) {
12     Buku *temp = new Buku();
13     temp->judul = judul;
14     temp->pengarang = pengarang;
15     temp->tahunTerbit = tahunTerbit;
16     temp->prev = NULL;
17     temp->next = NULL;
18
19     if (posisi == 1) {
20         temp->next = head;
21         if (head != NULL) {
22             head->prev = temp;
23         }
24         head = temp;
25         return;
26     }
27
28     Buku *cur = head;
29     for (int i = 1; i < posisi - 1 && cur != NULL; ++i) {
30         cur = cur->next;
31     }
32
33     if (cur == NULL) {
34         cout << "Posisi melebihi panjang linked list" << endl;
35         return;
36     }
37
38     temp->next = cur->next;
39     temp->prev = cur;
40     if (cur->next != NULL) {
41         cur->next->prev = temp;
42     }
43     cur->next = temp;
44 }
```

```

63 // Menyesipkan node baru pada posisi tertentu
64 sisipNode(node1, "Algoritma Pemrograman", "Andi", 2017, 2);
65

```

Berikut Penjelasanbya:

- sisipNode: Fungsi ini bertujuan untuk menyisipkan node baru ke dalam linked list sesuai dengan posisi yang diinginkan.
- Buku\* &head: Merupakan referensi dari node pertama (head) dalam linked list.
- string judul, string pengarang, int tahunTerbit: Informasi yang ingin disimpan di node baru yang akan disisipkan.
- int posisi: Menunjukkan posisi di mana node baru akan disisipkan dalam linked list.
- Memeriksa jika posisi yang diminta adalah posisi pertama, maka node baru akan menjadi head baru.
- Jika bukan posisi pertama, maka node baru akan disisipkan di antara node yang sudah ada sesuai dengan posisi yang diminta.
- Mengubah pointer prev dan next dari node-node terkait untuk memasukkan node baru ke dalam linked list.

Pada methode ini saya mencoba menyisipkan node pada posisi kedua dan berikut hasilnya

```

Judul Buku : Basis Data
Pengarang Buku : Budi Raharjo
Tahun Terbit : 2015
Judul Buku : Algoritma Pemrograman
Pengarang Buku : Andi
Tahun Terbit : 2017
Judul Buku : Struktur Data
Pengarang Buku : Susanto
Tahun Terbit : 2016

```